SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of **Department of Economics, Management and Statistics**

Ph. D. program in **Economics, Statistics and Data Science**, **37°** cycle
Curriculum of **Big Data & Analytics for Business**

# Three shades of alignment: from distributional semantics to trustworthy and interpretable large language models

Filippo Pallucchini
Registration number: 883868

Supervisor: Fabio Mercorio

Coordinator: Matteo Manera

Academic Year 2024/2025

**Abstract**

This thesis advances a unified framework for alignment in Natural Language Processing (NLP), explored across three interdependent levels: distributional, behavioural, and epistemic. Together, these levels trace a coherent path from geometric correspondence in embedding spaces to trustworthy and interpretable behaviour in large language models (LLMs).

At the distributional level, alignment concerns the mapping of semantic structures between embedding spaces. The thesis introduces SeNSe (Embedding Alignment via Semantic Anchors Selection), an unsupervised method that identifies robust semantic anchors to enhance the stability and interpretability of cross-space mappings. Building on this foundation, MEAL (Multilingual Embeddings Alignment) applies distributional alignment to real-world data, estimating job similarity across multilingual labour market datasets. Complementing these contributions, Lost in Alignment offers the first systematic survey and taxonomy of cross-lingual contextual alignment methods, establishing a unified theoretical and methodological framework for contextual representation alignment.

At the behavioural level, alignment shifts from geometric correspondence to model behaviour: that is, how LLMs integrate, retrieve, and generate knowledge in ways consistent with factual and domain-specific constraints. Two novel Retrieval-Augmented Generation (RAG) systems are introduced: RE-FIN (Retrieval-based Enrichment for Financial Data), which enhances financial sentiment analysis through retrieval-based enrichment, and FLEX (Financial Language Enhancement with Guided LLM Execution), which achieves self-alignment via internal paraphrasing and perplexity-based selection. Together, these architectures embody complementary paradigms of external (knowledge-grounded) and internal (self-consistent) behavioural alignment, improving factual reliability and domain adaptation in financial NLP applications.

Finally, at the epistemic level, alignment addresses the correspondence between internal model representations and human-interpretable semantics. The thesis introduces SAFE (Sparse Autoencoder-based Framework for robust query Enrichment), a sparse autoencoder-based framework that detects and corrects factual inconsistencies in LLM outputs through interpretable feature representations. Deepening the sparse autoencoders' field, the thesis proposes SFAL (Semantic-Functional Alignment score), a novel metric for quantifying the degree of correspondence between semantic and functional feature spaces, enabling principled evaluation of auto-interpretability in sparse autoencoders. Together, SAFE and SFAL formalise epistemic alignment as the structural coherence between what models know, represent, and explain.

Overall, this thesis redefines alignment as a multidimensional principle connecting meaning, behaviour, and reasoning, charting a conceptual trajectory from distributional semantics to trustworthy, interpretable, and epistemically grounded large language models.

# List of publications

• *Anchors Selection for Cross-lingual Embedding Alignment through Time.* Proceedings of the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI), 2022. [267]

• *SeNSe: embedding alignment via semantic anchors selection.* Published in the International Journal of Data Science and Analytics (IJDSA), 2024. [221]

• *Explainable AI for Text Classification: Lessons from a Comprehensive Evaluation of Post Hoc Methods.* Published in the Cognitive Computation journal, 2024.[43]

• *Alignment of Multilingual Embeddings to Estimate Job Similarities in Online Labour Market.* Proceedings of the IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA): Industry Track, 2024. [67]

• *RE-FIN: Retrieval-based Enrichment for Financial data.* Proceedings of the 31st International Conference on Computational Linguistics (COLING): Industry Track, 2025. [220]

• *Self-explanatory and Retrieval-augmented LLMs for Financial Sentiment Analysis.* Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC), 2025. [269]

• *Lost in Alignment: A Survey on Cross-Lingual Alignment Methods for Contextualized Representation.* Published in the ACM Computing Surveys journal (CSUR), 2025. [268]

• *SAFE: A Sparse Autoencoder-Based Framework for Robust Query Enrichment and Hallucination Mitigation in LLMs.* Proceedings of the findings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2025. [2]

• *SFAL: Semantic-Functional Alignment Scores for Distributional Evaluation of Auto-Interpretability in Sparse Autoencoders.* Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP): Industry Track, 2025. [242]

# Acknowledgements

Alla mia famiglia, che accompagna me e la mia passione per la ricerca conferendole una dignità che la mia sola logica non sarebbe in grado di cogliere.

*Whoever does not know it and can no longer wonder, no longer marvel, is as good as dead, and his eyes are dimmed.*

*L'uomo al quale non è più familiare il senso del mistero, che ha perso la capacità di umiliarsi e meravigliarsi davanti alla creazione è come un uomo morto.*
*A. Einstein*

# Contents

# List of Acronyms

**AI** Artificial Intelligence
**API** Application Programming Interface
**BERT** Bidirectional Encoder Representations from Transformers
**BLI** Bilingual Lexicon Induction
**BiLSTM** Bidirectional LSTM
**CCA** Canonical Correlation Analysis
**CLM** Causal Language Modeling
**CSLS** Cross-Lingual Similarity Scaling
**COS** Cosine Similarity
**CoT** Chain-of-Thought
**ELMo** Em-beddings from Language Models
**ESCO** European Skills, Competences, Qualifications and Occupations
**FFN** Feed-Forward neural Network
**FLEX** Financial Language Enhancement with Guided LLM Execution
**FSA** Financial Sentiment Analysis
**GAN** Generative Adversarial Networks
**GPT** Generative Pre-trained Transformer
**IR** Information Retrieval
**LLM** Large Language Model
**LMI** Labour Market Intelligence
**LSC** Lexical Semantic Change
**LSTM** Long Short-Term Memory
**mBERT** Multilingual Bidirectional Encoder Representations from Transformers
**MEAL** Multilingual Embeddings Alignment
**MLM** Masked language modelling
**ML** Machine Learning
**NER** Named Entity Recognition
**NDCG** Normalised Discounted Cumulative Gain
**NLI** Natural Language Inference
**NLP** Natural Language Processing
**OJA** Online Job Advertisement
**OT** Optimal Transport
**PALM** Pathways Language Mode
**PCA** Principal Component Analysis

**POS** Part-Of-Speech
**PPMI** Positive Pointwise Mutual Information
**QA** Question-Answering
**RAG** Retrieval-Augmented Generation
**RE-FIN** Retrieval-based Enrichment for Financial Data
**RLHF** Reinforcement Learning from Human Feedback
**RNN** Recurrent Neural Network
**RoBERTa** Robustly Optimised BERT Pretraining Approach
**SAFE** Sparse Autoencoder-based Framework for robust query Enrichment
**SAE** Sparse Autoencoder
**SeNSe** Embedding Alignment via Semantic Anchors Selection
**SFAL** Semantic-Functional Alignment score
**SOTA** State Of The Art
**TF-IDF** Term Frequency–Inverse Document Frequency
**TLM** Translation Language Modeling
**XLM** Cross-Lingual Language Model
**XLM-R** XLM-RoBERTa

# List of Figures

# List of Tables

# Introduction

In 2013, the breakthrough idea by Mikolov et al. [246] opened a new era in language representation, following the distributional hypothesis. A substantial portion of the literature has since been dedicated to methods for *aligning* different embedding spaces. When embeddings are created through a dedicated training phase, with a specific corpus and algorithm, they need to be projected into a shared space to become comparable. This problem has evolved alongside the progress of embedding models, which have moved from static representations to contextualised ones. In these models, an embedding vector is no longer uniquely associated with a word; instead, its representation varies according to the surrounding context.

While the research community has continuously proposed approaches to this field, the term *alignment* has gradually assumed a broader meaning. It is no longer tied exclusively to embedding spaces, but also to one of the greatest challenges brought by the evolution of Natural Language Processing (NLP): the advent of Large Language Model (LLM). This challenge concerns the necessity to *align* LLMs' text generation with factual knowledge, specific domains, or human preferences (e.g., safety, ethics).

In both contexts, the goal of *alignment* is to bring into correspondence contents of similar nature. The ultimate aim of this thesis is to extend the concept of *alignment* even further, toward the ambitious objective of aligning natural language itself with the activation spaces of LLMs. This is first addressed through an innovative method for hallucination mitigation that exploits the internal representations of LLMs, and then through a new metric designed to measure the quality of those interpretations.

## 0.1 Embedding Alignment

The study of embedding alignment arises from the foundational distributional hypothesis that "you shall know a word by the company it keeps" [133]. Early models such as Word2Vec [246] and GloVe [276] operationalised this principle, generating dense vector representations of words based on their co-occurrence patterns. However, embeddings trained on different corpora or languages were not directly comparable, necessitating transformation-based alignment techniques.

A seminal contribution was the linear mapping model proposed by Mikolov et al. [245], which employed an orthogonal Procrustes transformation to map monolingual embedding spaces into a shared bilingual space. Later extensions incorporated Canonical Correlation Analysis (CCA) [92], Optimal Transport (OT) and Groov-Wasserstein alignments [280], and Wasserstein–Procrustes for-

mulations [123], which preserved geometric relationships among embeddings. These models sought to learn transformations that approximated isometries between vector spaces [177, 320].

The concept of anchor points, seed lexica or pivot words with known correspondences, emerged as central to the success of alignment [8, 268]. Our own contribution, **SeNSe** [221], formalised anchor selection as a semantic neighbourhood consistency problem. By selecting words whose local topological structure remains stable across embedding spaces, SeNSe improved the stability and interpretability of the learned mapping, especially in diachronic and low-resource contexts. For demonstrating the advantages brought by the method,we also propose a domain specific application of the method in the labour market domain, named **Multilingual Embeddings Alignment (MEAL)**[67].

The evolution from static to contextual embeddings [71, 277] transformed the alignment problem into one involving dynamic, context-dependent spaces. Contextual models such as ELMo [278], BERT [71], and Multilingual Bidirectional Encoder Representations from Transformers (mBERT) [281] enabled zero-shot cross-lingual transfer, but also revealed non-isomorphic geometries and anisotropy issues across languages [307, 344, 365]. Research addressed these challenges through adversarial alignment [307], teacher–student distillation [295], and hybrid static–contextual schemes [131].

Due to the importance of the topic and the absence of a comprehensive study of those methods in the literature we proposed a survey. In **Lost in Alignment** [268](Chapter 3), we provided the first systematic taxonomy of cross-lingual contextual alignment methods, categorising them into off-line, on-line, and joint training strategies.

## 0.2 Large Language Model alignment

The transition from embedding alignment to LLM alignment represents a conceptual leap, from geometric alignment between vector spaces to behavioural and epistemic alignment between model outputs and human intent. With the emergence of models such as Generative Pre-trained Transformer (GPT)-3 [35], Pathways Language Mode (PALM) [55] and LLaMA [339], alignment became synonymous with ensuring factual accuracy, value consistency, and controllability [392].

Early approaches centred on *instruction tuning* and *Reinforcement Learning from Human Feedback (RLHF)*, which fine-tuned LLMs to produce responses that align with human preferences [392, 163]. However, these methods struggled with factual drift and domain-specific adaptation. Retrieval-Augmented Generation (RAG) [183] emerged as a robust paradigm to anchor LLM outputs in verifiable, external knowledge bases. In this framework, a retriever component dynamically fetches relevant passages, which the generator then integrates during response synthesis. This architecture has been applied to open-domain Question-Answering (QA), conversational agents, and specialised knowledge retrieval tasks [395].

Building on this foundation, we introduced two domain-specific RAG systems designed for Financial Sentiment Analysis (FSA).
The first, **Retrieval-based Enrichment for Financial Data (RE-FIN)** [220], retrieves qualitative information from curated financial knowledge bases, enriching model inputs while minimising hallucinations. It demonstrated substantial improvements in FSA tasks compared with fine-tuned models [345, 192, 364]. The second, **Financial Language Enhancement with Guided LLM Execution (FLEX)** [269], adopts a self-rewriting strategy in which the LLM generates

paraphrases and applies perplexity-based selection to achieve internal self-alignment without explicit retrieval. Together, RE-FIN and FLEX embody two complementary paradigms: external knowledge-grounded alignment and internal semantic alignment.

Beyond factual grounding, LLM alignment also involves aligning model reasoning processes. Techniques such as Chain-of-Thought (CoT) prompting [356] and instruction distillation [388] refine LLMs' internal representations to favour truthful and consistent reasoning paths. Nevertheless, the risk of hallucination [161, 224] and epistemic opacity persists, prompting the need for interpretability-driven alignment mechanisms, explored in Part III of this thesis.

## 0.3 Alignment of spaces of different nature

The third and most abstract notion of alignment addressed in this thesis concerns the correspondence between heterogeneous representational spaces: linguistic, semantic, and functional. Unlike embedding alignment, which operates at the lexical level, and LLM alignment, which regulates output behaviour, this level focuses on aligning the *internal activation manifolds* of neural models with interpretable linguistic concepts.

Recent work in *mechanistic interpretability* [18, 110, 149] has shown that complex behaviours of LLMs can often be decomposed into interpretable subspaces within their hidden layers. Sparse Autoencoder (SAE)s [201] have emerged as key tools for isolating meaningful latent features corresponding to human-understandable phenomena, such as negation, sentiment polarity, and entity consistency. These methods aim to create a bridge between the latent representations of a model and the symbolic or semantic features of language.

However, the alignment of such heterogeneous spaces is not trivial. LLMs frequently exhibit *hallucinations*, outputs that deviate from factual or logical consistency [160, 338]. In response, we introduced **Sparse Autoencoder-based Framework for robust query Enrichment (SAFE)** [2], which detects and mitigates hallucinations by enriching model queries using interpretable features extracted from SAEs. SAFE integrates with multiple hallucination detection systems such as SINdex [3], HaloCheck [85], and SelfCheckGPT [226], and demonstrates performance improvements across most important State Of The Art (SOTA) benchmarks.

Reflecting on the fundamental components of SAFE, we proposed **Semantic-Functional Alignment score (SFAL)** [242], a novel metric-based framework for quantifying the alignment between linguistic semantics and internal features. SFAL measures the correlation between functional latent dimensions extracted by SAEs and their semantic counterparts in the embedding space. By analysing the mutual information between functional activations and semantic interpretations, SFAL formalises how internal mechanisms reflect external meaning.

Together, SAFE and SFAL define a new paradigm of *epistemic alignment*, where trustworthiness arises from structural correspondence between human-interpretable semantics and internal representations. This research moves beyond the distributional and behavioural levels of alignment to explore how meaning, truth, and interpretability can coexist within the same representational geometry.

## 0.4 Thesis Structure

The thesis is articulated in three main Parts, organised as follows:

**Part I** introduces the theoretical foundations of distributional alignment. It begins by outlining the evolution from static to contextual embeddings, setting the stage for a discussion on how semantic spaces can be projected and compared (Chapter 1). The Part then presents two original contributions: SeNSe (Chapter 2), a novel unsupervised approach to embedding alignment based on semantic anchor selection, and MEAL (Chapter 2), an application of multilingual alignment to the estimation of job similarity in labour market data. Finally, the Part concludes with Lost in Alignment (Chapter 3), the first systematic survey and taxonomy of cross-lingual alignment methods for contextualised representations.

**Part II** focuses on alignment in domain-specific LLMs. After reviewing the theoretical underpinnings of LLMs and RAG (Chapter 4), it presents two novel RAG-based systems, RE-FIN and FLEX, for financial sentiment analysis (Chapter 5). These systems illustrate two complementary paradigms of alignment: external, through dynamic retrieval of factual knowledge, and internal, through self-consistent model rewriting.

**Part III** explores trustworthy and interpretable alignment in LLMs, moving from behavioural control to mechanistic understanding (Chapter 6). It first analyses hallucination as a systemic issue in language models and introduces SAFE (Chapter 7), a sparse autoencoder-based framework for hallucination detection and mitigation. It then proposes SFAL (Chapter 8), a novel metric for measuring the correspondence between functional and semantic feature spaces, enabling quantitative assessment of interpretability.

Finally, the thesis concludes (Conclusions 8.7) with a reflection on the achieved results and outlines future research directions toward more robust, explainable, and ethically aligned NLP systems.

# Part I

# Distributional Alignment: from Static to Contextual Representations

In this first part of the thesis, we introduce the foundational level of alignment: the alignment of embedding spaces. Starting from the distributional hypothesis and the evolution from static to contextual embeddings, we explore how different models can be projected into shared semantic spaces. This level of alignment forms the basis for the more complex alignments discussed in later parts. Beyond its theoretical motivation, Part I also examines how alignment techniques can serve practical purposes, such as tracking semantic shifts or comparing languages and domains. The proposed methodologies, ranging from SeNSe to MEAL, demonstrate how embedding alignment can reveal meaningful relations between linguistic systems, while the survey Lost in Alignment provides a comprehensive framework to position these contextual embedding alignment methods within the wider research landscape.

# Chapter 1

# Background

## 1.1 Vector Semantics and Embeddings

Words that occur in similar contexts tend to have similar meanings. This link between similarity in how words are distributed and similarity in what they mean is called the **distributional hypothesis**. The hypothesis was first formulated in the 1950s by linguists like Joos [162], Harris [133], and Firth [97], who noticed that words which are synonyms (like oculist and eye-doctor) tended to occur in the same environment (e.g., near words like eye or examined) with the amount of meaning difference between two words "corresponding roughly to the amount of difference in their environments" [133]. In this chapter, we introduce **vector semantics**, which instantiates this linguistic hypothesis by learning representations of the meaning of words, called **embeddings**, directly from their distributions in texts. These representations are used in every natural language processing application that makes use of meaning, and the static embeddings we introduce here underlie the more powerful dynamic or contextualised embeddings like BERT. These word representations are also an example of representation learning, automatically learning useful representations of the input text. Finding such self-supervised ways to learn representations of the input, instead of creating representations by hand via feature engineering, is an important focus of NLP research[24].

### 1.1.1 Lexical Semantics

Let's begin by introducing some basic principles of word meaning. How should we represent the meaning of a word? In the n-gram models and in classical NLP applications, our only representation of a word is as a string of letters, or an index in a vocabulary list. For completeness, here we report a brief discussion of n-gram models and classical NLP applications like Naive Bayes classifiers.

**N-gram language model** It is the simplest kind of language models [163]. An n-gram is a sequence of $n$ words: a 2-gram (bigram) is a two-word sequence of words like "The water", or "water of", and a 3-gram (trigram) is a three-word sequence of words. n-gram model is a probabilistic model that can estimate the probability of a word given the n-1 previous words, and thereby also assign probabilities to entire sequences [163]. The assumption that the probability of a word depends only on the previous word is called a **Markov** assumption.

**Multinomial naive Bayes classifier** It is so called because it is a Bayesian classifier that makes two simplifying (naive) assumptions about how the features interact. The first intuition of the classifier is shown in Fig. 1.1. For NLP applications a text document is represented as if it were a **bag of words**, that is, an unordered set of words with their position ignored, keeping only their frequency in the document. In the example in the Fig. 1.1, instead of representing the word order in all the phrases like "I love this movie" and "I would recommend it", it is simply noted that the word *I* occured 5 times in the entire excerp, the word *it* 6 times, the word *love, recommend*, and *movie* once, and so on.

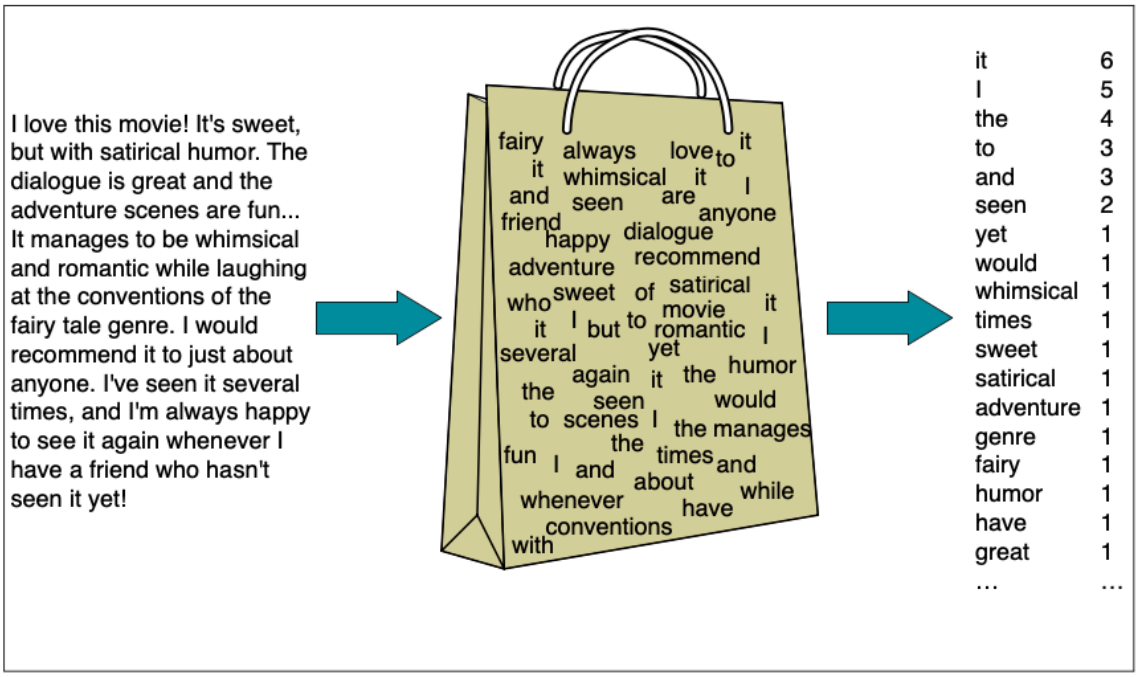



Figure 1.1: Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the bag-of-words assumption) and it is made use of the frequency of each word.

The second is commonly called the naive Bayes assumption: this is the conditional independence assumption that the probabilities of the features (for a text classfier they would be the words) are independent given the specific class and hence can be 'naively' multiplied.

But, these representations are very simplistic We'll want a model of word meaning to do all sorts of things for us. It should tell us that some words have similar meanings (*cat* is similar to *dog*), others are antonyms (*cold* is the opposite of *hot*), some have positive connotations (*happy*), while others have negative connotations (*sad*) It should represent the fact that the meaning of *buy*, *sell*, and *pay* offer differing perspectives on the same underlying purchasing event (if I buy something from you, you've probably sold it to me, and I likely paid you; this is the idea of **semantic frame**). More generally, a model of word meaning should allow us to draw inferences to address meaning-related tasks like QA or dialogue[163].

**Lemmas and Senses** Let's start by looking at how one word, like *mouse* (as done by Jurafsky and Martin [163]), might be defined in a dictionary (simplified from the online dictionary WordNet). Consider two phrases like:
1. *any of numerous small rodents...*
2. *a hand-operated device that controls a cursor...*
Here the form *mouse* is the **lemma**, also called the **citation form**. The form *mouse* would also be the lemma for the word *mice*; dictionaries don't have separate definitions for inflected forms like *mice*. Similarly *sing* is the lemma for *sing, sang, sung*. In many languages, the infinitive form is used as the lemma for the verb, so Spanish *dormir* "to sleep" is the lemma for *duermes* "you sleep". The specific forms *sung* or *carpets* or *sing* or *duermes* are called **wordforms**.

Each lemma can have multiple meanings; the lemma *mouse* can refer to the rodent or the cursor control device. These aspects of the meaning of *mouse* are called **word senses**. The fact that lemmas can be **polysemous** (have multiple senses) can make interpretation difficult.

**Synonymy** One important component of word meaning is the relationship between word senses. For example, when one word has a sense whose meaning is identical to a sense of another word, or nearly identical, the two senses of those two words are **synonyms**. Or, a more formal definition of synonymy (between words rather than senses) is that two words are synonymous if they are substitutable for one another in any sentence without changing the truth conditions of the sentence, the situations in which the sentence would be true. While substitutions between some pairs of words like *car/automobile* or *water/*$H_2O$ are truth-preserving, the words are still not identical in meaning. Indeed, probably no two words are absolutely identical in meaning. One of the fundamental tenets of semantics, called the **principle of contrast** [117, 244, 58], states that a difference in linguistic form is always associated with some difference in meaning. For example, the word $H_2O$ is used in scientific contexts and would be inappropriate in a hiking guide (water would be more appropriate), and this genre difference is part of the meaning of the word. In practice, the word synonym is therefore used to describe a relationship of approximate or rough synonymy [163].

**Word Similarity** While words don't have many synonyms, most words do have lots of similar words. *Cat* is not a synonym of *dog*, but *cats* and *dogs* are certainly similar words. In moving from synonymy to similarity, it will be useful to shift from talking about relations between word senses (like synonymy) to relations between words (like similarity). Dealing with words avoids having to commit to a particular representation of word senses, which will turn out to simplify our task. The notion of word similarity is very useful in larger semantic tasks. Knowing how similar two words are can help in computing how similar the meaning of two phrases or sentences are, a very important component of tasks like QA, paraphrasing, and summarisation. One way of getting values for word similarity is to ask humans to judge how similar one word is to another. A number of datasets have resulted from such experiments. For example the SimLex-999 dataset [138] gives values on a scale from 0 to 10, like the examples below, which range from near-synonyms (*vanish*, *disappear*) to pairs that scarcely seem to have anything in common (*hole*, *agreement*).

**Word Relatedness** The meaning of two words can be related in ways other than similarity. One such class of connections is called word relatedness [36], also traditionally called word association in psychology. Consider the meanings of the words *coffee* and *cup*. *Coffee* is not similar to *cup*; they share practically no features (coffee is a plant or a beverage, while a cup is a manufactured

object with a particular shape). But *coffee* and *cup* are clearly related; they are associated by co-participating in an everyday event (the event of drinking coffee out of a cup). Similarly, *scalpel* and *surgeon* are not similar but are related eventively (a surgeon tends to make use of a scalpel) [163]. One common kind of relatedness between words is if they belong to the same **semantic field**. A semantic field is a set of words which cover a particular semantic domain and bear structured relations with each other. For example, words might be related by being in the semantic field of hospitals (*surgeon*, *scalpel*, *nurse*, *anesthetic*, *hospital*), restaurants (*waiter*, *menu*, *plate*, *food*, *chef*), or houses (*door*, *roof*, *kitchen*, *family*, *bed*).

**Connotation** The word **connotation** has different meanings in different fields, but here we use it to mean the aspects of a word's meaning that are related to a writer or reader's emotions, sentiment, opinions, or evaluations. For example, some words have positive connotations (*wonderful*) while others have negative connotations (*dreary*). Even words whose meanings are similar in other ways can vary in connotation; consider the difference in connotations between *fake*, *knockoff*, *forgery*, on the one hand, and *copy*, *replica*, *reproduction* on the other, or *innocent* (positive connotation) and *naive* (negative connotation). Some words describe positive evaluation (*great*, *love*) and others negative evaluation (*terrible*, *hate*). Positive or negative evaluation language is called **sentiment** and word sentiment plays a role in important tasks like sentiment analysis, stance detection, and applications of NLP to the language of politics and consumer reviews. Early work on affective meaning [264] found that words varied along three important dimensions of affective meaning:

- **valence**: the pleasantness of the stimulus
- **arousal**: the intensity of emotion provoked by the stimulus
- **dominance**: the degree of control exerted by the stimulus

Thus, words like *happy* or *satisfied* are high on valence, while *unhappy* or *annoyed* are low on valence. *Excited* is high on arousal, while *calm* is low on arousal. *Controlling* is high on dominance, while *awed* or *influenced* are low on dominance. Each word is thus represented by three numbers, corresponding to its value on each of the three dimensions. Osgood et al. [264] noticed that in using these 3 numbers to represent the meaning of a word, the model was representing each word as a point in a three-dimensional space, a vector whose three dimensions corresponded to the word's rating on the three scales. This revolutionary idea that word meaning could be represented as a point in space (e.g., that part of the meaning of *heartbreak* can be represented as the point [2.45,5.65,3.58]) was the first expression of the vector semantics models that we introduce next [163].

### 1.1.2 Vector Semantics

**Vector semantics** is the strandard way to represent word meaning in NLP, helping to model many aspects of word meaning. The roots of the models lie in 1950s when two ideas converged: Osgood's [264] idea mentioned above to use a point in three-dimensional space to represent the connotation of a word, and the distributional hypothesis by Joos [162], Harris [133], and Firth [97] that proposed to define the meaning of a word by its distribution in language use, meaning its neighbouring words or grammatical environments. Their idea was that two words that occur in very similar distributions (whose neighbouring words are similar) have similar meanings. Let's consider the example by Jurafsky and Martin [163], suppose you didn't know the meaning of the word *ongchoi* (a recent borrowing from Cantonese) but you see it in the following contexts:

(1.1) *Ongchoi* is delicious sauteed with garlic.

(1.2) *Ongchoi* is superb over rice.

(1.3) ...*ongchoi* leaves with salty sauces ...

And suppose that you had seen many of these context words in other contexts:

(2.4) ...spinach sauteed with garlic over rice ...

(2.5) ...chard stems and leaves are delicious ...

(2.6) ...collard greens and other salty leafy greens

The fact that *ongchoi* occurs with words like *rice* and *garlic* and *delicious* and *salty*, as do words like *spinach*, *chard*, and *collard greens*, might suggest that *ongchoi* is a leafy green similar to these other leafy greens. We can do the same thing computationally by just counting words in the context of *ongchoi*. The idea of vector semantics is to represent a word as a point in a multidimensional semantic space that is derived from the distributions of word neighbours. Vectors for representing words are called **embeddings** (although the term is sometimes more strictly applied only to dense vectors rather than sparse vectors). The word "embedding" derives from its mathematical sense as a mapping from one space or structure to another, although the meaning has shifted [163].


Figure 1.2: A two-dimensional (t-SNE) projection of embeddings for some words and phrases, showing that words with similar meanings are nearby in space. The original 60-dimensional embeddings were trained for sentiment analysis. Simplified from Li et al. [187] with color added for explanation.

Fig. 1.2, taken from [163], shows visualisation of embeddings learned for sentiment analysis, showing the location of selected words projected down from 60-dimensional space into a 2-dimensional space. Notice the distinct regions containing positive words, negative words, and neutral function words. The fine-grained model of word similarity of vector semantics offers enormous power to NLP applications. Indeed, n-gram models or Naive Bayes classifiers cited before depend on the same words appearing in the training and test sets, with an absent or scarce ability to handle unseen terms. But, by representing words as embeddings, a classifier can assign sentiment as long as it sees some words with similar meanings. And, most importantly, vector semantic models can be learned automatically from text without supervision.

### 1.1.3 Words as vectors

Vector or distributional models of meaning are generally based on a **co-occurrence matrix**, a way of representing how often words co-occur. Two popular matrices are the term-document matrix and the term-term matrix; we'll look at the term-term matrix. A **vector** is, at heart, just a list or array of numbers and a **vector space** is a collection of vectors, and is characterised by its **dimension**. Vectors in a 3-dimensional vector space have an element for each dimension of the space. A **term-term matrix**, also called the **word-word matrix** or the **term-context matrix**, represents words as vectors of word counts. This matrix is thus of dimensionality of $|V|x|V|$, where $|V|$ is the vocabulary size, and each cell records the number of times the row (target) word and the column (context) word co-occur in some context in some training corpus. The context could be the document, in which case the cell represents the number of times the two words appear in the same document. It is most common, however, to use smaller contexts, generally a window around the word, for example of 4 words to the left and 4 words to the right, in which case the cell represents the number of times (in some training corpus) the column word occurs in such a $\pm 4$ word window around the row word [163]. Here are four examples of words in their windows:

(1.7) …is traditionally followed by **cherry** pie, a traditional dessert …

(1.8) …often mixed, such as **strawberry** rhubarb pie. Apple pie …

(1.9) …computer peripherals and personal **digital** assistants. These devices usually …

(1.10) …a computer. This includes **information** available on the internet …

If we then take every occurrence of each word (say **strawberry**) and count the context words around it, we get a word-word co-occurrence matrix. Fig. 1.3 shows a simplified subset of the word-word co-occurrence matrix for these four words computed from the Wikipedia corpus[1].

| | aardvark | ... | computer | data | result | pie | sugar | ... |
|---|---|---|---|---|---|---|---|---|
| **cherry** | 0 | ... | 2 | 8 | 9 | 442 | 25 | ... |
| **strawberry** | 0 | ... | 0 | 0 | 1 | 60 | 19 | ... |
| **digital** | 0 | ... | 1670 | 1683 | 85 | 5 | 4 | ... |
| **information** | 0 | ... | 3325 | 3982 | 378 | 5 | 13 | ... |

Figure 1.3: Co-occurrence vectors for four words in the Wikipedia corpus, showing six of the dimensions (hand-picked for pedagogical purposes). The vector for digital is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser [163].

Note in Fig. 1.3 that the two words *cherry* and *strawberry* are more similar to each other (both *pie* and *sugar* tend to occur in their window) than they are to other words like *digital*; conversely, *digital* and *information* are more similar to each other than, say, to strawberry. Fig. 1.4 shows a spatial visualisation.

[1] https://www.english-corpora.org/wiki/

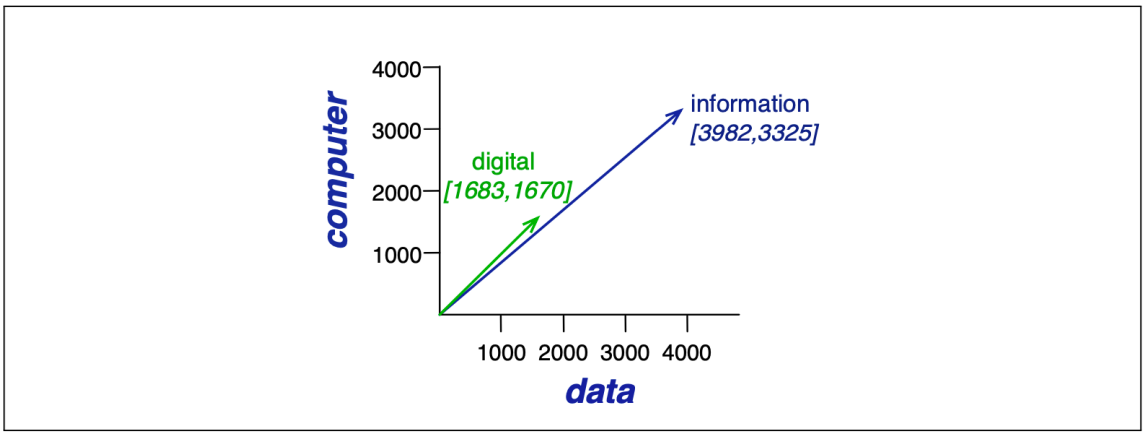



Figure 1.4: A spatial visualisation of word vectors for *digital* and *information*, showing just two of the dimensions, corresponding to the words *data* and *computer* [163].

### 1.1.4 Similarity measures

To measure similarity between two target words $v$ and $w$, we need a metric that takes two vectors (of the same dimensionality, for example, with words as dimensions, hence of length $|V|$) and gives a measure of their similarity. By far the most common similarity metric is the **cosine** of the angle between the vectors. The cosine, like most measures for vector similarity used in NLP, is based on the dot product operator from linear algebra, also called the inner product:

$$\text{dot product}(\mathbf{v},\mathbf{w}) = \mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N \tag{1.1}$$

The dot product acts as a similarity metric because it will tend to be high just when the two vectors have large values in the same dimensions. Alternatively, vectors that have zeros in different dimensions—orthogonal vectors—will have a dot product of 0, representing their strong dissimilarity. This raw dot product, however, has a problem as a similarity metric: it favours long vectors. The **vector length** is defined as

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2} \tag{1.2}$$

The dot product is higher if a vector is longer, with higher values in each dimension. Considering the term-term matrix, more frequent words have longer vectors, since they tend to co-occur with more words and have higher co-occurrence values with each of them. The raw dot product will thus be higher for frequent words. But this is a problem; is there a similarity metric that tells us how similar two words are regardless of their frequency? It is possible to modify the dot product to normalise for the vector length by dividing the dot product by the lengths of each of the two vectors. This **normalised dot product** turns out to be the same as the cosine of the angle between the two vectors. So, the Cosine Similarity (COS) metric between two vectors $\mathbf{v}$ and $\mathbf{w}$ can be computed as:

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \tag{1.3}$$

For some applications, it is usefull pre-normalise each vector by dividing it by its length, creating a unit vector of length 1. For **unit vectors**, the dot product is the same as the cosine. The cosine value ranges from 1 for vectors pointing in the same direction, through 0 for orthogonal vectors, to -1 for vectors pointing in opposite directions.

## 1.2 Static Word Embedding

In the previous sections we saw how to represent a word as a sparse, long vector with dimensions corresponding to words in the vocabulary in a collection. There are methods more sophisticated than using just the frequency, as showed in the term-term matrix, like: **Term Frequency–Inverse Document Frequency (TF-IDF)** weighting, usually used when the dimensions are documents, and **Positive Pointwise Mutual Information (PPMI)** algorithm, usually used when the dimensions are words. We now introduce a more powerful word representation: **embeddings**, short dense vectors. Its dimensions don't have a clear interpretation and the vectors are **dense**: instead of vector entries being sparse, mostly-zero counts or functions of counts, the values will be real-valued numbers that can be negative. It turns out that dense vectors work better in every NLP task than sparse vectors [163]. While we don't completely understand all the reasons for this, we have some intuitions. Representing words as 300-dimensional dense vectors requires our classifiers to learn far fewer weights than if we represented words as 50,000-dimensional vectors, and the smaller parameter space possibly helps with generalisation and avoiding overfitting. Dense vectors may also do a better job of capturing synonymy. For example, in a sparse vector representation, dimensions for synonyms like *car* and *automobile* are distinct and unrelated; sparse vectors may thus fail to capture the similarity between a word with *car* as a neighbour and a word with *automobile* as a neighbour. In this section we introduce one method for computing embeddings: **skip-gram with negative sampling**, sometimes called **SGNS**. The skip-gram algorithm is one of two algorithms in a software package called word2vec [247, 246]. The word2vec methods are fast, efficient to train, and easily available online with code and pretrained embeddings. Word2vec embeddings are **static embeddings**, meaning that the method learns one fixed embedding for each word in the vocabulary [163]. The intuition of word2vec is that instead of counting how often each word w occurs near, say, *apricot*, we'll instead train a classifier on a binary prediction task: "Is word $w$ likely to show up near *apricot*?" We don't actually care about this prediction task; instead, we'll take the learned classifier weights as the word embeddings. The revolutionary intuition here is that we can just use running text as implicitly supervised training data for such a classifier; a word $c$ that occurs near the target word *apricot* acts as gold 'correct answer' to the question "Is word $c$ likely to show up near *apricot*?" This method, often called **self-supervision**, avoids the need for any sort of hand-labelled supervision signal. This idea was first proposed in the task of neural language modeling, when Bengio et al. [24] and Collobert et al. [59] showed that a neural language model (a neural network that learned to predict the next word from prior words) could just use the next word in running text as its supervision signal, and could be used to learn an embedding representation for each word as part of doing this prediction task. The intuition of skip-gram is:

1. Treat the target word and a neighbouring context word as positive examples.

2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the learned weights as the embeddings.

### 1.2.1 The classifier

Let's start by thinking about the classification task, and then turn to how to train. Imagine a sentence like the following, with a target word *apricot*, and assume we're using a window of ±2 context words:

... lemon, a [tablespoon of apricot jam, a] pinch ...
c1 c2 w c3 c4

Our goal is to train a classifier such that, given a tuple $(w, c)$ of a target word $w$ paired with a candidate context word $c$ (for example (*apricot, jam*), or perhaps (*apricot, aardvark*)) it will return the probability that $c$ is a real context word (true for *jam*, false for *aardvark*):

$$P(+|w, c) \tag{1.4}$$

The probability that word $c$ is not a real context word for $w$ is just 1 minus Eq. 1.4:

$$P(-|w, c) = 1 - P(+|w, c) \tag{1.5}$$

How does the classifier compute the probability $P$? The intuition of the skip-gram model is to base this probability on embedding similarity: a word is likely to occur near the target if its embedding vector is similar to the target embedding. To compute similarity between these dense embeddings, we rely on the intuition that two vectors are similar if they have a high **dot product** (indeed, cosine is just a normalised dot product). In other words:

$$Similarity(w, c) \approx \mathbf{c} \cdot \mathbf{w} \tag{1.6}$$

The dot product $\mathbf{c} \cdot \mathbf{w}$ is not a probability, it's just a number ranging from $-\infty$ to $\infty$ (since the elements in word2vec embeddings can be negative, the dot product can be negative). To turn the dot product into a probability, we'll use the **logistic** or **sigmoid** function $\sigma(x)$, the fundamental core of logistic regression:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{1.7}$$

We model the probability that word $c$ is a real context word for target word $w$ as:

$$P(+|w, c) = \sigma(\mathbf{c} \cdot \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{c} \cdot \mathbf{w})} \tag{1.8}$$

The sigmoid function returns a number between 0 and 1, but to make it a probability we'll also need the total probability of the two possible events ( $c$ is a context word, and $c$ isn't a context word) to sum to 1. We thus estimate the probability that word $c$ is not a real context word for $w$ as:

$$\begin{aligned} P(-|w,c) &= 1 - P(+|w,c) \\ &= \sigma(-\mathbf{c}\cdot\mathbf{w}) = \frac{1}{1+\exp(\mathbf{c}\cdot\mathbf{w})} \end{aligned} \tag{1.9}$$

Eq. 1.8 gives us the probability for one word, but there are many context words in the window. Skip-gram makes the simplifying assumption that all context words are independent, allowing us just to multiply their probabilities:

$$P(+|w,c_{1:L}) = \prod_{i=1}^{L} \sigma(\mathbf{c}_i\cdot\mathbf{w}) \tag{1.10}$$

$$\log P(+|w,c_{1:L}) = \sum_{i=1}^{L} \log \sigma(\mathbf{c}_i\cdot\mathbf{w}) \tag{1.11}$$

In summary, skip-gram trains a probabilistic classifier that, given a test target word $w$ and its context window of $L$ words $c_{1:L}$, assigns a probability based on how similar this context window is to the target word. The probability is based on applying the logistic (sigmoid) function to the dot product of the embeddings of the target word with each context word. To compute this probability, we just need embeddings for each target word and each context word in the vocabulary.

Fig. 1.5 shows the intuition of the parameters we'll need. Skip-gram actually stores two embeddings for each word, one for the word as a target and one for the word considered as context. Thus, the parameters we need to learn are two matrices $\mathbf{W}$ and $\mathbf{C}$, each containing an embedding for every one of the $|V|$ words in the vocabulary $V$. Let's now turn to learning these embeddings (which is the real goal of training this classifier in the first place).

### 1.2.2 Learning skip-gram embeddings

The learning algorithm for skip-gram embeddings takes as input a corpus of text and a chosen vocabulary size $N$. It begins by assigning a random embedding vector for each of the $N$ vocabulary words, and then proceeds to iteratively shift the embedding of each word $w$ to be more like the embeddings of words that occur nearby in texts, and less like the embeddings of words that don't occur nearby. Let's start by considering a single piece of training data (we will use the same example of previous section, taken from Jurafsky and Martin [163]):

... lemon, a [tablespoon of apricot jam, a] pinch ...
c1 c2 w c3 c4

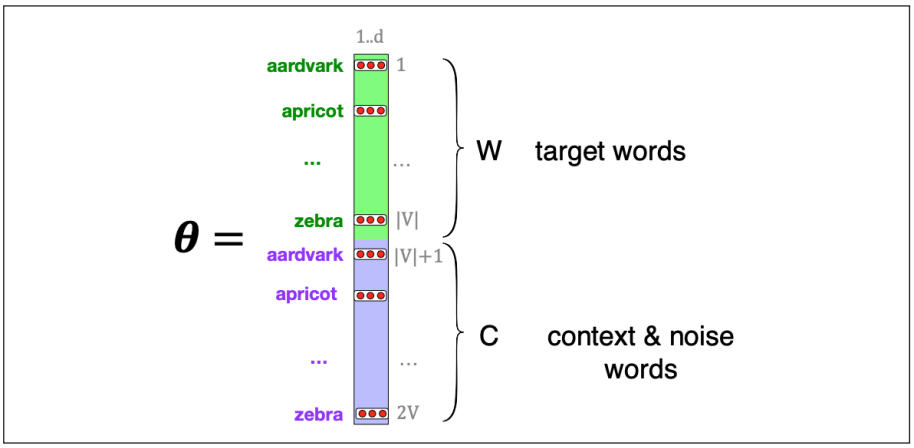



Figure 1.5: The embeddings learned by the skipgram model. The algorithm stores two embeddings for each word, the target embedding (sometimes called the input embedding) and the context embedding (sometimes called the output embedding). The parameter $\theta$ that the algorithm learns is thus a matrix of $2|V|$ vectors, each of dimension $d$, formed by concatenating two matrices, the target embeddings $\mathbf{W}$ and the context+noise embeddings $\mathbf{C}$ [163].

This example has a target word $w$ (apricot), and 4 context words in the $L = \pm 2$ window, resulting in 4 positive training instances: e.g. one positive instance $(w, c_{pos})$ would be *apricot, jam*). For training a classifier, we also need negative examples: e.g. one positive instance $(w, c_{neg})$ would be *apricot, aardvark*, where *aardvark* is randomly selected. In fact SGNS uses more negative examples than positive examples (with the ratio between them set by a parameter $k$). So for each of these $(w, c_{pos})$ training instances, we'll create $k$ negative samples, each consisting of the target $w$ plus a 'noise word' $c_{neg}$. A noise word is a random word from the lexicon, constrained not to be the target word $w$.

Given the set of positive and negative training instances, and an initial set of embeddings, the goal of the learning algorithm is to adjust those embeddings to

- Maximise the similarity of the target word, context word pairs $(w, c_{pos})$ drawn from the positive examples
- Minimise the similarity of the $(w, c_{neg})$ pairs from the negative examples.

If we consider one word/context pair $(w, c_{pos})$ with its $k$ noise words $c_{neg_1} \dots c_{neg_k}$, we can express these two goals as the following loss function $L$ to be minimized (hence the $-$); here the first term expresses that we want the classifier to assign the real context word $c_{pos}$ a high probability of being a neighbor, and the second term expresses that we want to assign each of the noise words $c_{neg_i}$ a high probability of being a non-neighbor, all multiplied because we assume independence:

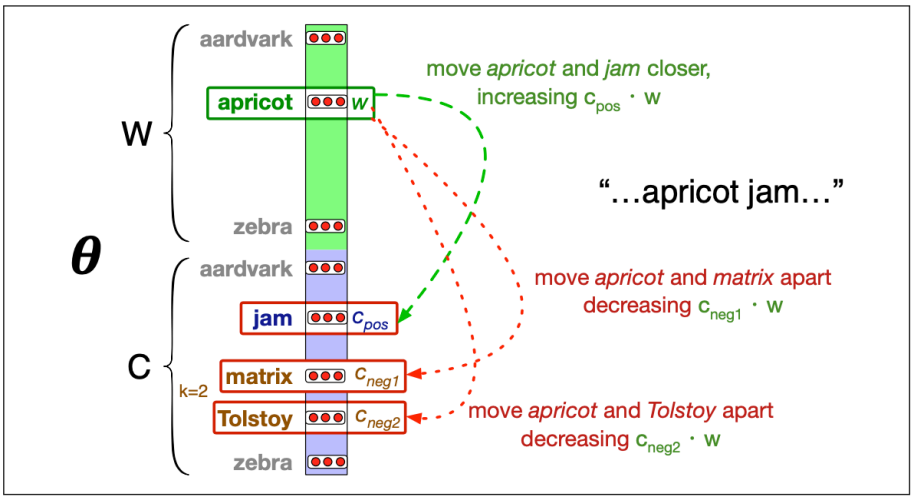



Figure 1.6: Intuition of one step of gradient descent. The skip-gram model tries to shift embeddings so the target embeddings (here for *apricot*) are closer to (have a higher dot product with) context embeddings for nearby words (here *jam*) and further from (lower dot product with) context embeddings for noise words that don't occur nearby (here *Tolstoy* and *matrix*) [163].

$$L = -\log\left[P(+|w, c_{pos})\prod_{i=1}^{k} P(-|w, c_{neg_i})\right] \tag{1.12}$$

$$= -\left[\log P(+|w, c_{pos}) + \sum_{i=1}^{k} \log P(-|w, c_{neg_i})\right] \tag{1.13}$$

$$= -\left[\log P(+|w, c_{pos}) + \sum_{i=1}^{k} \log\left(1 - P(+|w, c_{neg_i})\right)\right] \tag{1.14}$$

$$= -\left[\log \sigma(c_{pos} \cdot w) + \sum_{i=1}^{k} \log \sigma(-c_{neg_i} \cdot w)\right] \tag{1.15}$$

Like for language modelling, the calculations are done in log space to avoid underflow and increase speed (indeed, instead of multiplying probabilities we sum them, resulting in bigger numbers). So, we want to maximise the dot product of the word with the actual context words, and minimise the dot products of the word with the $k$ negative sampled non-neighbour words. We minimise this loss function using **stochastic gradient descent**. Fig. 1.6 shows the intuition of one step of learning.

To get the gradient, we need to take the derivative of Eq. 1.12 with respect to the different embeddings. It turns out the derivatives are the following:

$$\frac{\partial L}{\partial c_{pos}} = [\sigma(\mathbf{c}_{pos} \cdot \mathbf{w}) - 1]\mathbf{w} \tag{1.16}$$

$$\frac{\partial L}{\partial c_{neg}} = [\sigma(\mathbf{c}_{neg} \cdot \mathbf{w})]\mathbf{w} \tag{1.17}$$

$$\frac{\partial L}{\partial w} = [\sigma(\mathbf{c}_{pos} \cdot \mathbf{w}) - 1]\mathbf{c}_{pos} + \sum_{i=1}^{k}[\sigma(\mathbf{c}_{neg_i} \cdot \mathbf{w})]\mathbf{c}_{neg_i} \tag{1.18}$$

The update equations going from time step $t$ to $t+1$ in stochastic gradient descent are thus:

$$\mathbf{c}_{pos}^{t+1} = \mathbf{c}_{pos}^{t} - \eta[\sigma(\mathbf{c}_{pos}^{t} \cdot \mathbf{w}^{t}) - 1]\mathbf{w}^{t} \tag{1.19}$$

$$\mathbf{c}_{neg}^{t+1} = \mathbf{c}_{neg}^{t} - \eta[\sigma(\mathbf{c}_{neg}^{t} \cdot \mathbf{w}^{t})]\mathbf{w}^{t} \tag{1.20}$$

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta\left[[\sigma(\mathbf{c}_{pos}^{t} \cdot \mathbf{w}^{t}) - 1]\mathbf{c}_{pos}^{t} + \sum_{i=1}^{k}[\sigma(\mathbf{c}_{neg_i}^{t} \cdot \mathbf{w}^{t})]\mathbf{c}_{neg_i}^{t}\right] \tag{1.21}$$

Just as in logistic regression, then, the learning algorithm starts with randomly initialised **W** and **C** matrices, and then walks through the training corpus using gradient descent to move **W** and **C** so as to minimise the loss in Eq. 1.12 by making the updates in (Eq. 1.19)-(Eq. 1.21).

Recall that the skip-gram model learns two separate embeddings for each word $i$: the **target embedding** $\mathbf{w}_i$ and the **context embedding** $\mathbf{c}_i$, stored in two matrices, the **target matrix W** and the **context matrix C**. It's common to just add them together, representing word $i$ with the vector $\mathbf{w}_i + \mathbf{c}_i$. Alternatively we can throw away the **C** matrix and just represent each word $i$ by the vector $\mathbf{w}_i$. The context window size $L$ affects the performance of skip-gram embeddings, and experiments often tune the parameter $L$ on a devset.

### 1.2.3 Fasttext model

**Fasttext** [28] is an extension of word2vec model (seen in the previous sections) and adresses a problem with word2vec as we have presented it so far: it has no good way to deal with **unknown words**, that are those words that appear in a test corpus but were unseen in the training corpus. A related problem is **word sparsity**, such as in languages with rich morphology, where some of the many forms for each noun and verb may only occur rarely. Fasttext deals with these problems by using **subword models**, representing each word as itself plus a bag of constituent n-grams, with special boundary symbols < and > added to each word. For example, with $n = 3$ the word *where* would be represented by the sequence <where> plus the character n-grams:

<wh, whe, her, ere, re>

Then a skipgram embedding is learned for each constituent n-gram, and the word *where* is represented by the sum of all of the embeddings of its constituent n-grams. Unknown words can then be presented only by the sum of the constituent n-grams. A fasttext open-source library, including pretrained embeddings for 157 languages, is available at https://fasttext.cc.

## 1.3 Alignment of static word embedding

The representations introduced in the previous Sec. 1.2, capture distributional similarity and provide powerful dense features for NLP tasks. However, embeddings trained on different corpora, domains, or languages are learned independently and are therefore not directly comparable: their coordinate systems are arbitrary and may be rotated or reflected versions of one another. To make these embeddings comparable, we need to find a way to *align* them. The study of **embedding alignment** seeks to recover a transformation between such spaces, enabling meaningful comparisons across languages, time periods, or modalities [268, 221]. Aligning two clouds of embeddings, or high dimensional real vectors, is a fundamental problem in Machine Learning (ML) with applications in natural language processing such as unsupervised word and sentence translation [293, 100] or in computer vision such as point set registration [64] and structure-from-motion [337].

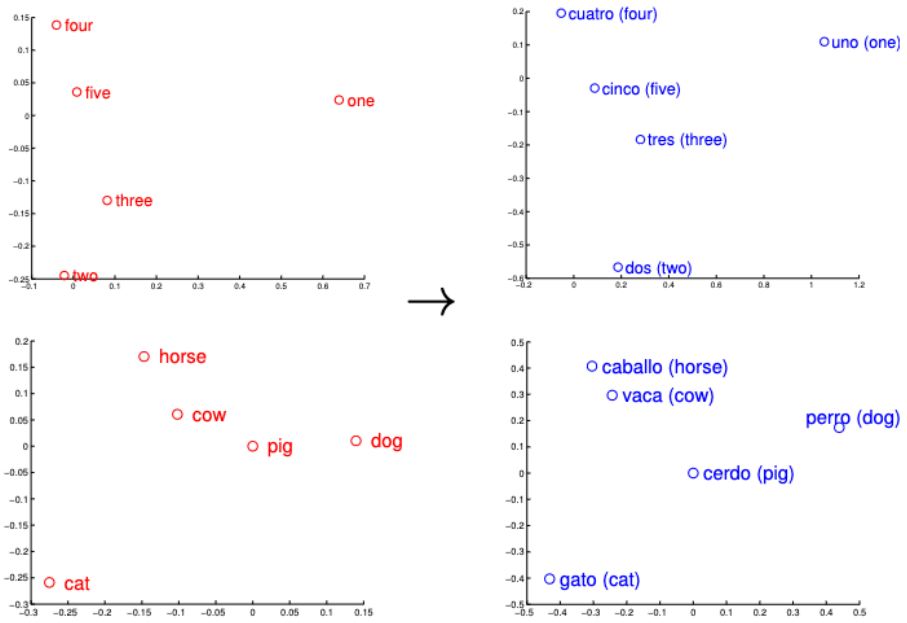



Figure 1.7: Distributed word vector representations of numbers and animals in English (left) and Spanish (right). The five vectors in each language were projected down to two dimensions using Principal Component Analysis, and then manually rotated to accentuate their similarity. It can be seen that these concepts have similar geometric arrangements in both spaces, suggesting that it is possible to learn an accurate linear mapping from one space to another [245].

Fig. 1.7 from Mikolov et al. [245] gives simple visualisation to illustrate why it's possible to align linearly two embedding spaces. In Fig. 1.7, we visualise the vectors for numbers and animals in English and Spanish, and it can be easily seen that these concepts have similar geometric arrangements. The reason is that as all common languages share concepts that are grounded in the real world (such as that *cat* is an animal smaller than a *dog*), there is often a strong similarity between the vector spaces. The similarity of geometric arrangements in vector spaces is the key reason why methods showed in next sections works well.

### 1.3.1 Linear Mapping and Orthogonal Procrustes Alignment

One of the earliest and most influential families of alignment methods is based on learning a **linear transformation** between source and target embeddings. Suppose we have two embedding matrices, $X, Y \in \mathbb{R}^{n \times d}$, representing $n$ words in $d$-dimensional spaces, and that some correspondences between them are known; these correspondences are known as **anchors** as we'll see in Sec. 1.3.5. The goal is to find a transformation matrix $Q$ that minimises the distance between paired embeddings:

$$\min_{Q} \|XQ - Y\|^2 \tag{1.22}$$

Imposing the orthogonality constraint $Q \in O(d)$ leads to the **Orthogonal Procrustes problem**:

$$\min_{Q \in O(d)} \|XQ - Y\|^2, \tag{1.23}$$

where $O(d)$ denotes the set of $d \times d$ orthogonal matrices, i.e., matrices satisfying $Q^\top Q = I_d$. Orthogonal transformations preserve distances and inner products, maintaining the semantic relationships within each embedding space. The problem admits a **closed-form solution**: let $U\Sigma V^\top$ be the singular value decomposition (SVD) of $X^\top Y$, then

$$Q^* = UV^\top. \tag{1.24}$$

This elegant solution, originally derived by Schönemann [305], can be understood intuitively by expanding the optimisation objective. Specifically, minimising $\|XQ - Y\|^2$ is equivalent to maximising the trace term $\mathrm{Tr}(Q^\top X^\top Y)$, since the other components are constant with respect to $Q$:

$$\min_{Q \in O(d)} \|XQ - Y\|_F^2 \iff \max_{Q \in O(d)} \mathrm{Tr}(Q^\top X^\top Y) \tag{1.25}$$

Substituting the SVD $X^\top Y = U\Sigma V^\top$, we obtain:

$$\mathrm{Tr}(Q^\top X^\top Y) = \mathrm{Tr}(Q^\top U\Sigma V^\top) = \mathrm{Tr}(\Sigma(U^\top QV)) \tag{1.26}$$

Letting $Z = U^\top QV$, and noting that $Z$ is also orthogonal since it is the product of orthogonal matrices, the optimisation simplifies to:

$$\max_{Z \in O(d)} \mathrm{Tr}(\Sigma Z) \tag{1.27}$$

Since $\Sigma$ is diagonal and its entries (the singular values) are non-negative, the trace is maximised when $Z = I_d$. Thus, the optimal solution is obtained when $Q = UV^\top$.

This result has an important geometric interpretation: $U$ and $V$ define rotations (and possibly reflections) in the source and target spaces, while $\Sigma$ contains the degree of correlation between their axes. The optimal mapping $Q^* = UV^\top$ therefore corresponds to the pure rotation (or reflection)

that best aligns $X$ to $Y$, without scaling or distortion, preserving semantic distances. In other words, $Q^*$ represents the most faithful rigid transformation between the two embedding manifolds.

Smith et al. [319] demonstrated that this orthogonality constraint is not only computationally convenient but also theoretically necessary for self-consistency when mapping between embedding spaces. Moreover, manifold alignment theory shows that this SVD-based solution minimises subspace misalignment and provides a geometric justification for linear embedding transformations [350].

### 1.3.2 Unsupervised Mapping: Wasserstein-Procrustes Formulation

In the absence of direct word correspondences, alignment can be formulated as a **joint optimisation** over both the mapping $W$ and a permutation matrix $P$ that encodes unknown word correspondences:

$$\min_{W \in O(d),\, P \in \mathcal{P}_n} \|XW - PY\|^2, \tag{1.28}$$

where $\mathcal{P}_n$ is the set of $n \times n$ permutation matrices. This *Wasserstein-Procrustes problem* couples the geometric transformation with an optimal matching, effectively combining ideas from orthogonal mapping and OT [123]. Although the joint problem is non-convex, alternating optimisation techniques are effective because each subproblem has a known solution: Procrustes alignment for $W$ and Hungarian or Sinkhorn matching for $P$. In practice, optimisation proceeds on the Stiefel manifold, the space of orthogonal matrices, with stochastic gradient updates.

### 1.3.3 Canonical Correlation Analysis Alignment

Another classical approach to alignment relies on **CCA**, a statistical method for finding correlated projections between two views of data. Given two sets of embeddings $X$ and $Y$ with known word correspondences (anchors), CCA finds projection vectors $u$ and $v$ that maximise the correlation $\text{corr}(Xu, Yv)$:

$$\max_{u,v} \frac{u^\top \Sigma_{XY} v}{\sqrt{u^\top \Sigma_{XX} u \, v^\top \Sigma_{YY} v}}, \tag{1.29}$$

where $\Sigma_{XX}$ and $\Sigma_{YY}$ are the covariance matrices of $X$ and $Y$, and $\Sigma_{XY}$ is their cross-covariance. The solution can be obtained through SVD on the *whitened* cross-covariance matrix $\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}$. The resulting CCA projections map embeddings into a shared latent space of maximal correlation [92, 211]. Extensions such as Deep CCA (DCCA) and Kernel CCA (KCCA) generalise the method to nonlinear mappings, yet retain the key property of a closed-form SVD solution.

### 1.3.4 Optimal Transport and Gromov-Wasserstein Alignment

A more recent and powerful family of approaches is grounded in **OT** theory, which models alignment as the problem of finding a cost-minimising coupling between two probability distributions. Each embedding set is viewed as a discrete measure over its vocabulary, and the OT formulation seeks a **transport plan** $T$ that minimises the overall cost of mapping the source embeddings $X$ onto the target embeddings $Y$:

$$\min_{T \in \mathcal{U}(p,q)} \langle T, C(X,Y) \rangle, \tag{1.30}$$

where $C(X,Y)$ is a cost matrix, whose entry $C_{ij}$ represents the distance (often the squared Euclidean distance) between a source vector $X_i$ and a target vector $Y_j$. The notation $\langle T, C(X,Y) \rangle = \sum_{i,j} T_{ij} C_{ij}$ denotes the total transport cost.

The transport plan $T$ defines how much *probability mass* should be transferred from each source embedding $X_i$ to each target embedding $Y_j$. However, $T$ must satisfy the so-called **prescribed marginals** $p$ and $q$, which specify how the total mass is distributed across the source and target domains. Formally, the set of admissible couplings $\mathcal{U}(p,q)$ is defined as:

$$\mathcal{U}(p,q) = \left\{ T \in \mathbb{R}_+^{n \times m} \,\middle|\, T\mathbf{1}_m = p,\, T^\top \mathbf{1}_n = q \right\}, \tag{1.31}$$

where $\mathbf{1}_m$ and $\mathbf{1}_n$ are vectors of ones of length $m$ and $n$. The first condition $T\mathbf{1}_m = p$ ensures that the total outgoing mass from each source word $X_i$ equals its probability $p_i$, while the second condition $T^\top \mathbf{1}_n = q$ ensures that the total incoming mass to each target word $Y_j$ equals its probability $q_j$. In other words, the marginals $p$ and $q$ act as constraints that preserve the overall probability distribution of both embedding spaces: the total mass in $X$ must exactly match the total mass received by $Y$. In most cross-lingual alignment applications, these marginals are chosen as uniform distributions (i.e., all words contribute equally), but they can also be weighted by word frequency to reflect corpus-specific priors.

This formulation, known as the *Kantorovich problem*, can be solved exactly or through **entropy-regularised** optimisation using Sinkhorn iterations [66], which introduce a smoothing term that drastically accelerates convergence.

However, classical OT is sensitive to arbitrary rotations of the embedding spaces because it directly compares coordinates in $\mathbb{R}^d$. To overcome this limitation, Alvarez-Melis and Jaakkola [8] proposed the *Gromov-Wasserstein (GW) alignment*, which instead compares the internal *relational geometry* of the spaces. In GW alignment, embeddings are treated as metric spaces characterised by their pairwise distances. Let $C_X(i,i') = \|X_i - X_{i'}\|^2$ and $C_Y(j,j') = \|Y_j - Y_{j'}\|^2$ denote the intra-space distance matrices for the source and target domains. GW then seeks a transport plan $T$ that minimises the discrepancy between these relational structures:

$$\min_{T \in \mathcal{U}(p,q)} \sum_{i,j,i',j'} \left| C_X(i,i') - C_Y(j,j') \right|^2 T_{ij}\, T_{i'j'}. \tag{1.32}$$

This quadratic objective encourages similar relational distances between pairs of words to be preserved across domains, aligning not only individual words but the entire geometric structure of the embedding manifolds. Although the problem is non-convex and NP-hard, it can be efficiently approximated through iterative projected-gradient methods [280], where each iteration alternates between solving a classic OT subproblem and updating the transport plan.

By design, GW alignment is invariant to rotations and translations of the embedding space, as it relies solely on pairwise distance relationships rather than absolute coordinates. Consequently, it provides a robust, geometry-preserving approach to unsupervised embedding alignment. In essence,

GW alignment seeks the transformation that best approximates an *isometry* between two metric spaces, thus offering a solid theoretical foundation for relational alignment in natural language embeddings.

### 1.3.5 Anchors in Alignment

The success of embedding alignment crucially depends on the choice of **anchor points**—a subset of words (or concepts) assumed to have equivalent semantics across the two embedding spaces [8]. These anchors, also called **seed lexicon**, guide the transformation toward a meaningful shared space. In the simplest supervised setting, anchors are derived from bilingual dictionaries or domain glossaries. However, this assumption is often unrealistic: lexica are incomplete, noisy, or biased toward frequent and unambiguous words [268].

**Lost in Alignment** [268], in Chapter 3, identifies anchor quality as one of the principal determinants of alignment stability, distinguishing between:

- **Supervised alignment**, which relies on gold-standard lexica and typically achieves high precision but poor scalability;
- **Semi-supervised alignment**, which begins with a small seed lexicon and iteratively expands it through self-learning or mutual nearest neighbours;
- **Unsupervised alignment**, which dispenses with anchors altogether, optimising both correspondences and mappings jointly.

While unsupervised methods (e.g., Wasserstein-Procrustes) are appealing, they often suffer from local minima and hubness phenomena. To mitigate these issues, recent research focuses on **data-driven anchor discovery**. Our SeNSe model [221], in Chapter 2 formalises this idea by constructing anchors based on *semantic neighbourhood consistency*. Instead of assuming a fixed bilingual lexicon, SeNSe identifies anchor candidates whose local semantic structure (i.e., nearest-neighbour distributions) remains stable across embedding spaces. These anchors are selected using a ranking function that measures the overlap between local semantic graphs, thereby filtering out words that exhibit high semantic drift or translation ambiguity. Empirical results show that SeNSe anchors yield more stable mappings and better bilingual lexicon induction performance, particularly in low-resource and diachronic scenarios.

In essence, anchors serve as the geometric and semantic scaffolding of alignment: they define the initial correspondence that guides optimisation. When selected adaptively—based on contextual or structural evidence—they can bridge the gap between purely unsupervised geometry-based methods and lexicon-dependent supervised approaches, achieving both accuracy and interpretability.

## 1.4 Contextual Word Embeddings

The **static embeddings** introduced in Sec 1.2 assign one fixed vector representation to each word type in the vocabulary, independent of its surrounding context. This approach, while effective for many tasks, suffers from an important limitation: words are often **polysemous**, as we introduced in previous Sec. 1.1.1, so they carry multiple meanings depending on usage. For instance, the word *bank* has very different senses in "the bank approved the loan" and "we sat by the river bank".

Static embeddings cannot distinguish these senses, as they are mapped to a single point in the embedding space.

To overcome this limitation, modern neural language models learn **contextual embeddings**, where each token is represented by a vector that depends on the specific linguistic context in which it occurs. The meaning of a word thus becomes a *function of its context*, not a static lookup.

### 1.4.1 From Static to Contextual Representations

Contextual embedding models assign each word token $x_i$ in a sentence $x_1, \ldots, x_n$ a context-sensitive embedding $\mathbf{h}_i$, computed by conditioning on all words in the sequence:

$$(x_1, x_2, \ldots, x_n) \longmapsto (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n) \tag{1.33}$$

where each $\mathbf{h}_i \in \mathbb{R}^d$ captures the semantics of token $x_i$ in its sentence-level context. Unlike word2vec [246] or GloVe [276], these models do not store a single embedding per word type, but rather *generate embeddings on the fly* through deep neural computation.

The fundamental architectural innovation that enables this is the **Transformer encoder** [348], which uses a mechanism called **self-attention** to integrate contextual information from all positions in the input sequence. Before discussing it, we briefly trace the historical evolution of contextual embeddings.

### 1.4.2 Historical Development

The first major step beyond static embeddings came with **ELMo** [278]), which produced word representations as functions of the internal states of a bidirectional Long Short-Term Memory (LSTM) trained on a large corpus. Each word token was represented by the concatenation of forward and backward LSTM hidden states, providing both left and right contextual information:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \tag{1.34}$$

ELMo demonstrated that context-dependent embeddings significantly improve performance across a wide range of NLP tasks, showing that meaning is better captured when both past and future words are integrated into the representation.

#### Feed-Forward Neural Networks and Activation Functions

Before introducing recurrent architectures, it is helpful to briefly review how a basic **Feed-Forward neural Network (FFN)** [297] operates. A neural network consists of a series of layers that apply learned linear transformations followed by nonlinear activation functions. Each layer computes a mapping from an input vector to an output vector, allowing the network to learn complex functions.

Given an input $\mathbf{x} \in \mathbb{R}^d$, a single-layer neural network computes:

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1.35}$$

where $\mathbf{W}$ is a matrix of learned weights, $\mathbf{b}$ is a bias vector, and $f(\cdot)$ is a nonlinear *activation function*. Common activation functions include:

$$\text{Sigmoid: } \sigma(x) = \frac{1}{1+e^{-x}} \tag{1.36}$$

$$\text{Tanh: } \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{1.37}$$

$$\text{ReLU: } \text{ReLU}(x) = \max(0, x) \tag{1.38}$$

Nonlinear activations are crucial: without them, a stack of linear layers would still represent only a linear transformation. The nonlinearity allows the network to learn hierarchical and compositional representations of input data.

For language tasks, feed-forward models such as the early neural language model by Bengio et al. [24] took as input a fixed-size window of previous words and predicted the next word. Formally, for a context window $(w_{t-n}, \ldots, w_{t-1})$ represented as embeddings $\mathbf{x}_{t-n}, \ldots, \mathbf{x}_{t-1}$, the model concatenates them and feeds them through a network:

$$\mathbf{h}_t = f(\mathbf{W}_1[\mathbf{x}_{t-n}; \ldots; \mathbf{x}_{t-1}] + \mathbf{b}_1) \tag{1.39}$$

followed by a softmax layer to predict the next word:

$$P(w_t \mid w_{t-n}, \ldots, w_{t-1}) = \text{softmax}(\mathbf{W}_2\mathbf{h}_t + \mathbf{b}_2) \tag{1.40}$$

While this architecture captures local patterns, it has a major limitation: the input window has a fixed size, so it cannot model dependencies longer than $n$ words. This limitation motivated the development of models capable of handling sequences of arbitrary length — the **Recurrent Neural Network (RNN)s** [359].

### From Feed-Forward to Recurrent Models

Recurrent Neural Networks extend the idea of neural computation to sequential data. Instead of processing each input independently, RNNs maintain a hidden *state* that evolves over time, allowing the network to remember information about previous elements in a sequence.

Conceptually, the RNN can be viewed as a FFN network that "loops" over time: the output at one step becomes part of the input for the next. This makes it particularly suited for language modeling, where the meaning of a word depends on preceding words. For example, in the phrase "the concert was held in the *cathedral*," the model must remember the context "was held in" to predict the next word.

Formally, given an input sequence of word embeddings $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, the hidden state $\mathbf{h}_t$ and output $\mathbf{y}_t$ are defined as:

$$\mathbf{h}_t = \tanh(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_x\mathbf{x}_t + \mathbf{b}_h) \tag{1.41}$$

$$\mathbf{y}_t = \text{softmax}(\mathbf{W}_y\mathbf{h}_t + \mathbf{b}_y) \tag{1.42}$$

The hidden state $\mathbf{h}_t$ acts as a dynamic summary of all inputs up to time $t$, and the parameters $(\mathbf{W}_h, \mathbf{W}_x, \mathbf{W}_y)$ are shared across all time steps, making the model efficient and flexible for sequences of varying lengths.

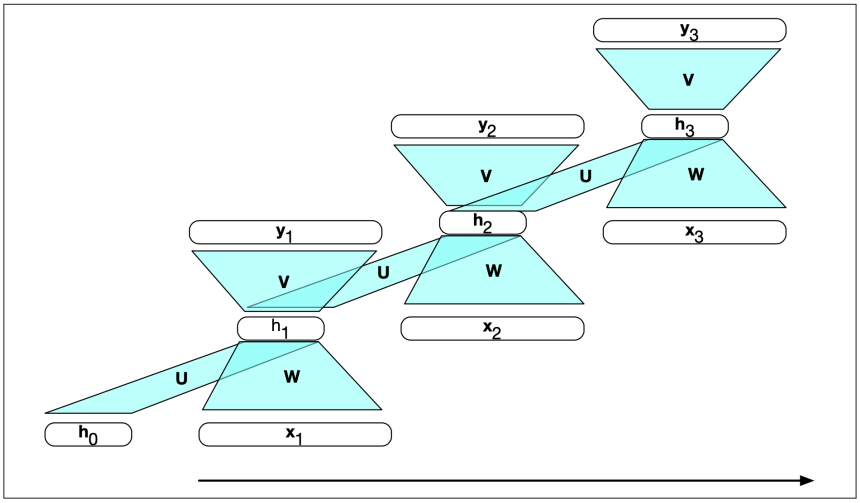



Figure 1.8: An RNN unrolled through time (taken from Jurafsky and Martin [163]). The same set of parameters is reused at each time step to update the hidden state $\mathbf{h}_t$ from the current input $\mathbf{x}_t$ and previous state $\mathbf{h}_{t-1}$. The hidden state acts as a dynamic memory summarising all prior context.

While RNNs can, in theory, capture dependencies over long spans, in practice they struggle due to the **vanishing and exploding gradient problem**. During backpropagation through time (BPTT) [359], the gradient of the loss with respect to earlier states becomes exponentially smaller or larger, making it difficult for the network to learn long-term relationships. If we consider the gradient of the loss $L$ with respect to an early hidden state $\mathbf{h}_{t-k}$, we have:

$$\frac{\partial L}{\partial \mathbf{h}_{t-k}} = \frac{\partial L}{\partial \mathbf{h}_t} \prod_{j=t-k+1}^{t} \mathbf{W}_h^\top \operatorname{diag}(f'(\mathbf{a}_j)) \tag{1.43}$$

where $\mathbf{a}_j$ are the pre-activation inputs and $f'$ the derivative of the activation. Because $\mathbf{W}_h$ and $f'$ often have eigenvalues smaller than 1, repeated multiplication drives the gradient toward zero, causing information from distant steps to fade.

### The LSTM Architecture

To mitigate these limitations, Hochreiter and Schmidhuber [139] proposed the **LSTM** network, which explicitly separates memory storage from control. The LSTM introduces an internal *cell state* $\mathbf{c}_t$ that can preserve information over long sequences, regulated by a set of *gates* that control how information is written, read, and forgotten.

At each time step $t$, given the input $\mathbf{x}_t$ and previous hidden state $\mathbf{h}_{t-1}$, the LSTM computes:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad \text{(input gate)} \tag{1.44}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad \text{(forget gate)} \tag{1.45}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad \text{(output gate)} \tag{1.46}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad \text{(candidate cell state)} \tag{1.47}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad \text{(cell state update)} \tag{1.48}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad \text{(hidden state output)} \tag{1.49}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\odot$ the elementwise product. The gates $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ take values between 0 and 1 and thus control, respectively, how much new information to add, how much old information to forget, and how much of the memory to expose as output.

This gating mechanism allows the LSTM to carry relevant information across many time steps while protecting it from interference, solving the vanishing gradient issue that limits standard RNNs.

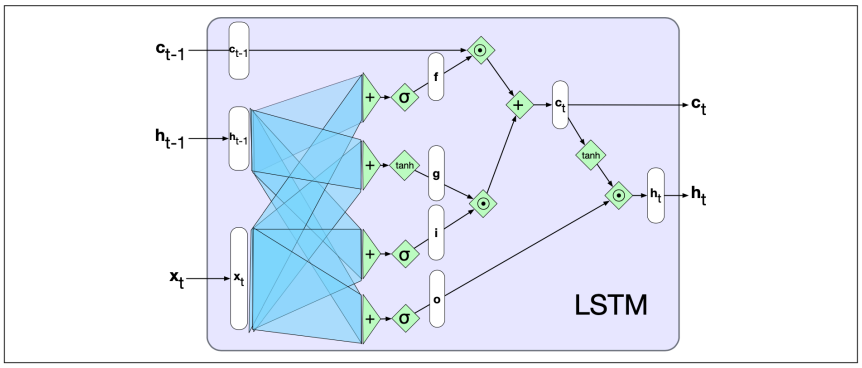



Figure 1.9: The LSTM cell structure (taken from Jurafsky and Martin [163]). The horizontal line at the top represents the cell state $\mathbf{c}_t$, which can carry information through time. Input, forget, and output gates (controlled by sigmoids) regulate the flow of information, while tanh non-linearities shape the candidate and output states.

### Bidirectional LSTMs and Contextualization

In the standard (unidirectional) LSTM, each hidden state $\mathbf{h}_t$ depends only on tokens preceding position $t$. However, many linguistic phenomena require access to both past and future context (e.g., disambiguating "*bank*" in "the bank of the river"). To model both directions, a **Bidirectional LSTM (BiLSTM)** [306] runs two separate LSTMs—one forward and one backward—and concatenates their hidden states:

$$\overrightarrow{\mathbf{h}}_t = \text{LSTM}_{\text{forward}}(x_t, \overrightarrow{\mathbf{h}}_{t-1}) \tag{1.50}$$

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\text{backward}}(x_t, \overleftarrow{\mathbf{h}}_{t+1}) \tag{1.51}$$

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \tag{1.52}$$

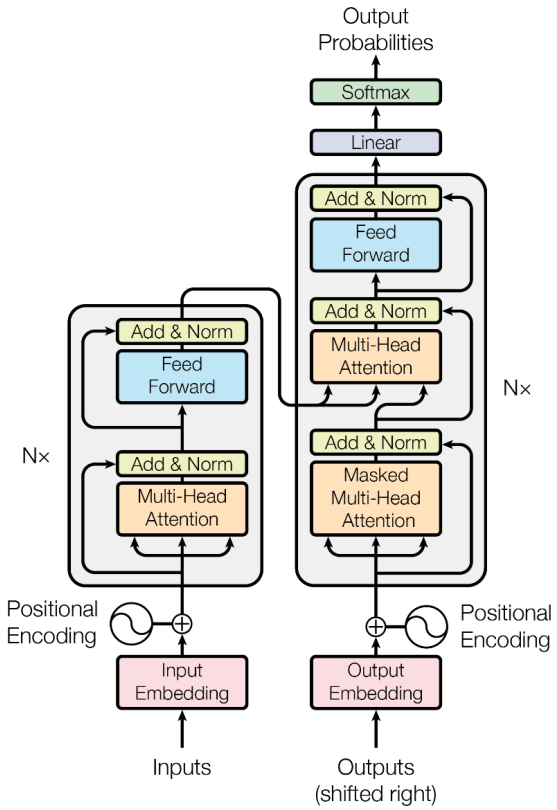



Figure 1.10: Transfomer Architecture taken from [348]

The resulting contextual representation $\mathbf{h}_t$ thus incorporates information from both sides of the current token. This bidirectional contextualization was the foundation of ELMo's architecture, which used pretrained BiLSTM layers to generate embeddings that reflect a word's meaning in its specific sentence context.

### Transition to Self-Attention Models

Although LSTMs successfully capture long-distance dependencies, they are inherently sequential—each time step must be processed in order—making training and inference slow for long sequences. The **Transformer architecture** [348] (shown in Fig. 1.10) replaced recurrence with **self-attention**, which allows all tokens in a sequence to interact simultaneously. This innovation enabled highly parallel computation and laid the foundation for **BERT** (Bidirectional Encoder Representations from Transformers) [71], which retains the idea of bidirectional contextualization while scaling efficiently to very large corpora and deeper networks.

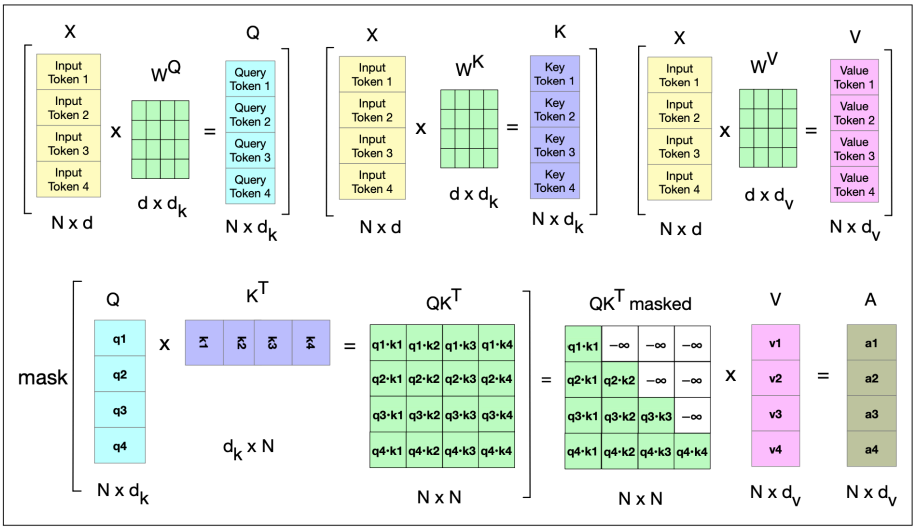



Figure 1.11: Schematic of the attention computation for a single attention head in parallel. The first row shows the computation of the **Q**, **K** and **V** matrices. The second row shows the computation of $\mathbf{QK}^\top$, the masking (the softmax computation and the normalising by dimensionality are not shown) and then the weighted sum of the value vectors to get the final attention vectors [163].

### 1.4.3 Bidirectional Transformer Encoders

The Transformer encoder maps a sequence of input token embeddings $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ to a sequence of contextual representations $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n]$. Each encoder layer consists of two main components:

1. A **multi-head self-attention** sublayer, which allows each token to attend to all others.
2. A **FFN** applied independently to each position.

Formally, for each attention head, we compute:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{1.53}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ are learned weight matrices. The attention mechanism computes the alignment scores between tokens as the scaled dot product:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{1.54}$$

Each attention weight reflects how much information a token should take from another token. Fig. 1.11 shows a schematic of all computations for a single attention head parallelised in matrix form. In bidirectional encoders like BERT, the attention matrix is **fully unmasked**, so every token can attend to all others on both sides—unlike autoregressive models (e.g., GPT) where attention is restricted to previous tokens.

Multiple attention heads operate in parallel:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O \tag{1.55}$$

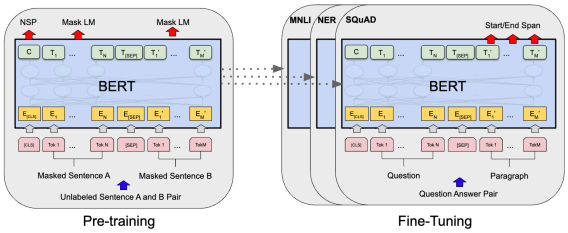



Figure 1.12: Overall pre-training and fine-tuning procedures for BERT taken from Devlin et al. [71]. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialise models for different downstream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers). In the figure, the authors denote the input embedding as $E$ and the final hidden vector $T$.

producing a richer contextual mixture of information. The output passes through a layer normalisation and residual connection:

$$\mathbf{Z} = \text{LayerNorm}(\mathbf{X} + \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \tag{1.56}$$

$$\mathbf{H} = \text{LayerNorm}(\mathbf{Z} + \text{FFN}(\mathbf{Z})) \tag{1.57}$$

Stacking $L$ encoder blocks yields deep representations $\mathbf{h}_i^{(L)}$ for each token $i$, which constitute its final contextual embedding.

### 1.4.4 Training via Masked Language Modeling

Bidirectional transformers are trained using the **Masked language modelling (MLM)** objective, which enables learning from both left and right context. A proportion of input tokens (typically 15%) are replaced with a special token [MASK]; the model must predict the original tokens based on the rest of the sentence:

$$\mathcal{L}_{MLM} = -\sum_{i \in M} \log P(x_i \mid x_1, \ldots, x_{i-1}, [MASK], x_{i+1}, \ldots, x_n) \tag{1.58}$$

where $M$ is the set of masked positions. This training signal forces the network to encode syntactic and semantic dependencies between all words, yielding rich contextual embeddings. Additionally, some models (like BERT) include a **next sentence prediction (NSP)** loss, encouraging the model to understand inter-sentence relations. In Fig. 1.12 we show the BERT idea as presented by the authors [71].

### 1.4.5 Extracting Contextual Representations

After pretraining, contextual embeddings can be extracted from any encoder layer. Typically, the final layer output $\mathbf{h}_i^{(L)}$ is used as the embedding of token $x_i$:

$$\mathbf{e}_i = \mathbf{h}_i^{(L)} \tag{1.59}$$

However, earlier layers tend to capture syntactic information (e.g., Part-Of-Speech (POS)), while higher layers capture semantic information. For many applications, embeddings are obtained by averaging the last $k$ layers:

$$\mathbf{e}_i = \frac{1}{k} \sum_{j=0}^{k-1} \mathbf{h}_i^{(L-j)} \tag{1.60}$$

These contextual embeddings change depending on the surrounding tokens, yielding distinct vectors for polysemous words. For example:

Context 1: *He sat by the river bank.*
Context 2: *The bank raised interest rates.*

The embeddings $\mathbf{e}_{bank}^{(1)}$ and $\mathbf{e}_{bank}^{(2)}$ will be distant in vector space, reflecting different senses.

### 1.4.6 Comparing Contextual and Static Embeddings

Static models like word2vec learn embeddings by predicting co-occurring words within a local window, optimising a binary classification loss (see Eq. 1.12). In contrast, contextual models implicitly optimise a prediction over all vocabulary tokens conditioned on the full sentence. Both approaches rely on the **distributional hypothesis** (see Sec. 1.1.2) but contextual models extend this hypothesis from types to **tokens**:

$$\text{Static: } f(w) = \mathbf{v}_w \quad \longrightarrow \quad \text{Contextual: } f(w, c) = \mathbf{v}_{w|c} \tag{1.61}$$

where $c$ denotes the surrounding context. The contextual mapping $f(w, c)$ allows a single word type to occupy multiple positions in semantic space depending on use, approximating a continuous model of word sense.

### 1.4.7 Fine-Tuning for Downstream Tasks

Pretrained transformer encoders can be **fine-tuned** for a wide range of downstream tasks by adding lightweight, task-specific layers and continuing gradient-based training. Given a labeled dataset $D = \{(x, y)\}$, we minimize a supervised loss:

$$\mathcal{L}_{task} = -\sum_{(x,y)\in D} \log P(y \mid x; \theta_{LM}, \theta_{task}) \tag{1.62}$$

Typical tasks include:

- **Sequence classification**: sentiment analysis, topic classification.
- **Sequence labeling**: Named Entity Recognition (NER), POS tagging.
- **Sentence-pair classification**: Natural Language Inference (NLI), paraphrase detection.

Because the pretrained parameters $\theta_{LM}$ already encode extensive linguistic knowledge, fine-tuning often requires minimal labeled data to achieve SOTA performance.

### 1.4.8 Alignment of Contextualised word embedding

Contextualised word embeddings extend the idea of static representations by allowing the same lexical item to assume different vector encodings depending on its surrounding context. Unlike static embeddings, which assign a single vector per word, contextual models such as BERT or Robustly Optimised BERT Pretraining Approach (RoBERTa) [210] dynamically adjust representations to reflect syntactic and semantic variability. As a result, aligning such embeddings across languages, domains, or models becomes a significantly more complex task, since the correspondence must be established not only between geometric spaces but also between context-dependent distributions of meaning.

This is exemplified in Fig. 1.13, adapted from [354], which shows a toy example of contextual embedding alignment where a single word may occupy multiple positions in the embedding space to reflect its distinct senses. This illustrates how contextual alignment must capture subtle semantic distinctions that emerge dynamically during language modeling.

Given the complexity and importance of this topic, we devoted an entire chapter of this Thesis (Chapter 3) to an in-depth analysis of SOTA methods for contextual embedding alignment. This chapter constitutes a core contribution of the work, providing the first systematic taxonomy, comparative discussion, and critical evaluation of existing approaches. Readers are therefore encouraged to refer to Chapter 3 for a comprehensive understanding of the methodologies, challenges, and perspectives that define this rapidly evolving research field.

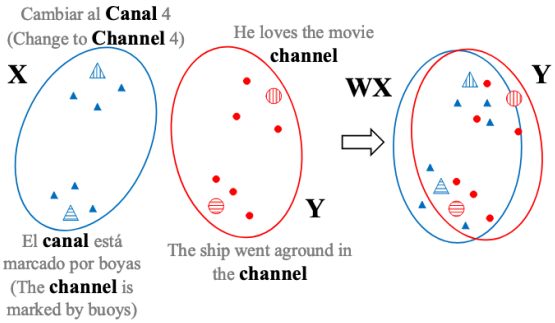



Figure 1.13: A toy illustration of the method, where contextualised embeddings of the word *canal* from Spanish is transformed to the semantic space of English. Taken from Wang et al. [354]

# Chapter 2

# Semantic Aware Static Embedding Alignment

In this chapter, we present SeNSe, a novel approach for finding qualitative anchors and improving embedding alignment, and MEAL, a framework that exploits the SeNSe logic for aligning embeddings trained with data from online job vacancies, for a deep understanding of semantics in the labour market.

## 2.1 SeNSe: Embedding Alignment via Semantic Anchors Selection

Word embeddings have proven extremely useful across many NLP applications in recent years. Several key linguistic tasks, such as machine translation and transfer learning, require comparing distributed representations of words belonging to different vector spaces within or among different domains and languages to be aligned, known as *embedding alignment*. To this end, several existing methods exploit words that are supposed to have the same meaning in the two corpora, called seed lexicon or anchors, as reference points to map one embedding into the other. All those methods consider only the word that is supposed to have the same meaning in the two spaces to choose anchors, whilst its neighbours or similar words are neglected. We propose SeNSe, an unsupervised method for aligning monolingual embeddings, generating a bilingual dictionary composed of words with the most similar meaning among word vector spaces. Our approach selects a seed lexicon of words used in the same context in both corpora without assuming a priori semantic similarities. We compare our method with well-established benchmarks showing SeNSe outperforms SOTA methods for embedding alignment on bilingual lexicon extraction in most cases.

[1]Note: The included results in this chapter are partially supported within the research activity of an ongoing H-2020 Project H2020-SC6-TRANSFORMATIONS-2018-2019-2020 named Pathways to Inclusive Labour Markets, grant agreement no. 101004703 named "Technological transformations, skills, and globalization - future challenges for shared prosperity" grant no 101004703 — PILLARS (https://www.h2020-pillars.eu/), aimed at using AI to predict future trends in the European Labour Market

### 2.1.1 Introduction

In the last decade, learning distributed representations of words in a vector space (also called word embeddings) has shown to be extremely helpful in several NLP applications [308]. For several years, there has been ongoing research in the field of learning universally applicable word representations; to this purpose, a wide range of methods has been used, including, but not limited to, (i) co-occurrence matrix factorisation [276], (ii) neural network training [246] and (iii) transformers [71]. Given the widespread interest in this subject, a critical question arises: *can we achieve consistency between two distinct word vector models and transfer lexical and semantic knowledge between them?* Indeed, several essential linguistic tasks require comparing distributed representations of words belonging to different vector spaces within or among different domains and languages. For this reason, word embeddings alignment has found fertile ground in research in the last years [298]. The potential to align word embedding spaces is intriguing for multiple reasons: (i) it enables us to compare the meanings of words across languages, a critical aspect for numerous NLP tasks, including, but not limited to, Bilingual Lexicon Induction (BLI) [383], machine translation [125], mining parallel corpora [177], cross-lingual Information Retrieval (IR) [159], etc; (ii) by establishing a shared representation space, cross-lingual word embeddings facilitate model transfer across languages, bridging the gap between resource-rich and low-resource languages [298]. Examples of applications are text classification [171], sentiment analysis [393], and dependency parsing [9]. Finally, (iii) even embeddings trained on the same corpus but in different phases produce misaligned vector spaces, making it impracticable to compare word vectors trained with different hyper-parameter settings or at different times [303]. An application in this field is Lexical Semantic Change (LSC) detection. Known as Semantic Shift, this procedure entails recognising and explaining fluctuations in language usage across various contexts, encompassing distinct periods and domains. By monitoring word sense evolution, Semantic Shift facilitates a more profound comprehension of the disparities in language among separate communities.

Most of the current approaches for embedding alignment primarily focus on constructing a "seed lexicon", which is essentially a collection of words that share the same meaning in both corpora. These words, commonly referred to as anchors, serve as reference points for learning a transformation that maps embeddings from one space to another by minimising their distances. As a result, numerous researchers have proposed various methods for identifying these anchors, as elaborated in Sec. 2.1.2. However, as far as we know, all the previous works build the seed lexicon, considering only the relation between pairs of words to determine if they have the same meaning in the two corpora. Conversely, in this research, we propose a method for anchor selection that considers the two candidate words and their neighbours in the respective vector spaces. The intuition behind this idea is that the semantic neighbourhood of a word better defines its meaning in its context than solely the representation of the word itself. For example, using the neighbourhood within the vector can assist in clarifying the meaning of a word in its usage context, particularly when the word has multiple interpretations. The name SeNSe (Semantic anchors Selection) reflects the use of the semantics of terms during the anchors' selection procedure. We show that, in most cases, our method outperforms SOTA in a complex evaluation task like BLI.

#### Motivation

Several NLP applications rely on the use of aligned vector spaces. Mapping cross-lingual word embeddings is a crucial application, particularly in BLI, where the resulting embeddings play a

key role in creating a bilingual dictionary. In particular, BLI refers to the process of automatically discovering and aligning translations of words or phrases in two different languages without relying on pre-existing bilingual dictionaries. This is typically accomplished by identifying each source word's translation through identifying its nearest counterpart in the target language [17]. This downstream task is widely employed as the standard evaluation metric for alignment models, having been utilised by numerous authors over the years. For instance, Shi et al. [314] proposed methods that combine unsupervised bitext mining and unsupervised word alignment, while Zhang et al. [382] integrated static word embeddings with contextual representations. Additionally, Li et al. [195] introduced a semi-supervised post-hoc reranking method. In addition to this use case, cross-lingual word embeddings can be utilised as features directly in NLP models. These models are designed for multiple languages and aim to enable cross-lingual transfer. In essence, the main idea is to train a model using data from one language and then employ it in another language by utilizing common cross-lingual features [298]. Täckström's work serves as an example of this application [327] for Dependency parsing while Zhang published one about POS tagging [101]. Cross-lingual word embeddings also provide valuable assistance in IR tasks [380, 253]. They act as helpful features that can establish a connection between the semantics of a query and the semantics of the target document collection (e.g. the query can talk about *cars*. In contrast, a relevant document may contain a near-synonym *automobile* [298]). As previously discussed, the ability to align word embeddings could significantly contribute to progress in Diachronic LSC detection, which involves automatically identifying changes in word senses over time, which is a flourishing new field within NLP [99, 129, 304]. Some interesting works in this field include the one proposed by Su et al. [324], who suggested a lexical-level masking strategy to post-train a converged language model, and the one proposed by Hofmann et al. [140]. Moreover, Tang et al. [330] present an unsupervised method to learn dynamic contextualised word embeddings by time-adapting a pretrained masked language model using prompts from manual and automatic templates. The prompts used the most similar term related to a specific word in two different periods. One of the logics used to select those terms is similar to our method, as it considers the most similar terms in the embedding space, but it is applied to contextual embedding. To clarify the matter, we propose an example in the context of Labour Market Intelligence (LMI) since it is a domain that could benefit from these techniques and show increasing use of embeddings, e.g., in detecting new emerging occupations or comparing them [114, 351, 217].

**An Inspiring Example in the Labour Market** Suppose two different embeddings are generated from job ads corpora: English (*EN*) and Italian (*IT*) language. The vectors of occupations and skills contain semantic and lexical information in their respective labour market when the job ads have been posted. Nevertheless, certain terms may undergo changes in their usage over time or possess distinct meanings in the two countries. For instance, the occupation that is called *Data Scientist* in the *EN*, might have required skills and mansions that are requested for a *Data Engineer* in *IT*. Or it could be that recruiters' use of the term *Digital Manager* is not the same as 5 years ago. How can we effectively compare the two labour markets or job ads from different years using their embeddings? Opting for common general words as anchors might be biased, as they could be used in entirely different contexts. For sure, *Data science* will appear in both vocabularies, but is it a good anchor candidate? We want to do it by checking if terms around *Data science* in the *IT* vector space are semantically similar to the ones in the *EN* one since it would mean that the term is used in similar contexts.

Another example could be found in the field of microblog posts, where it would be interesting to apply LSC detection as an additional feature to the study reported by Qu et al. [285] for tracking changes in topics over time.

#### Contribution

The contributions of this paper go through three directions:

**(1)** the strength of SeNSe is the flexibility of the application tied to the robust logic used; indeed, no other works select anchors for static embeddings alignment considering the semantic neighbour of the candidate words. We evaluate the effectiveness of the bilingual mappings for the BLI task, i.e., a word translation task that measures the accuracy of the induced dictionary compared to a gold standard. For this task, we compare the Precision@1, Precision@5, Precision@10 of SeNSe with previously reported SOTA results on benchmark datasets.
**(2)** An analysis of the benefits of SeNSe mapping compared to traditional unsupervised method alignment strategy (VecMap) is performed using a real work case scenario.
**(3)** SeNSe is applicable over each static word embedding; it is implemented and available to the whole community as an off-the-shelf Python tool on GitHub[2].

### 2.1.2 Related work

In this section, we review the main approaches for word embeddings and their respective alignment methods.

#### Traditional Word Embeddings

Traditional Word embeddings, like Word2Vec [246], FastText [28], and GloVe [276], are numerical representations derived from co-occurrence statistics, capturing word similarities in contexts. Word2Vec includes CBOW and skip-gram models, with CBOW predicting a target word from its context, and skip-gram predicting context from a target word. FastText extends skip-gram by incorporating sub-word information, benefiting rare words and typos. GloVe, a global log-bilinear regression model, stands out for unsupervised learning. Unlike training on entire sparse matrices, GloVe leverages statistical insights from nonzero elements in a word co-occurrence matrix. Its objective is to align word vectors' dot product with the logarithm of words' probability of co-occurrence, offering a nuanced understanding of word relationships. For this kind of models, the starting point for learning bilingual mapping is the choice of the seed lexicon to use; in previous literature, we can observe in brief three main methods for the selection of the seed lexicon:

**Off-the-shelf**: Most earlier approaches [245] utilized off-the-shelf or automatically generated bilingual lexicons, primarily consisting of common words. While the initial approach employed up to 5K pairs, later techniques have demonstrated the potential to obtain a cross-lingual embedding space with few seed pairs, as few as 25 [13].

**Weak supervision**: Alternative methods utilise weak supervision to generate seed lexicons, relying on cognates [319], shared numerals [13], or identical string spellings [320]. Obtaining such weak supervision is straightforward, and its effectiveness in producing competitive results with readily available lexicons has been demonstrated.

[2]https://github.com/filippopallucchini/SeNSe

**Learned**: Most unsupervised cross-lingual word embedding methods rely on the mapping approach and autonomously acquire an initial seed lexicon. As an example, the approach described in [177] adopts an adversarial learning strategy, incorporating the training of a discriminator to distinguish between projected target language embeddings and the original ones during the initial mapping process.

Since our work relies on this last category, we will provide a detailed description of these methods. The initialisation method employed in [11] is based on the heuristic assumption that translations exhibit similar similarity distributions across different languages. In Hoshen et al.'s [142] work, they reduce the dimensionality of vectors for the N most frequent words through Principal Component Analysis (PCA). Following this step, their objective is to identify the optimal transformation that minimises the sum of Euclidean distances while imposing cyclical consistency constraints, ensuring that vectors, once projected to the other language space and then back, remain unchanged. To establish alignment between monolingual word embedding spaces, Alvarez-Melis and Jaakkola address this challenge by employing the Gromov-Wasserstein method to solve an OT problem [8]. Each of the three anchor selection approaches outlined has specific drawbacks; for instance, some methods necessitate translated corpora, while others rely on assumptions that aren't universally valid, thereby constraining or compromising the potential for alignment in various applications (such as the assumption of identical meanings for specific terms across different corpora, as observed in off-the-shelf methods). Some approaches, like [245, 173], are built upon the idea that certain concepts in two languages share similar geometric arrangements, such as animals and numbers. The notion is connected to the intuitive understanding that as all common languages share concepts grounded in the real world (e.g., a cat is an animal smaller than a dog), the vector spaces often have a strong similarity. Rather than opting to select particular anchors for alignment, they depend exclusively on the most frequently used words in both corpora. However, this approach comes with certain limitations. Specifically, when applied to two corpora exhibiting substantial differences, like corpora from different domains or two languages with distinct cultural contexts, the method's performance may prove unsatisfactory due to inherent disparities. To eliminate the requirement for bilingual data in practical applications, Artetxe's method [13] leverages the structural similarity of embedding spaces. It creates a numeral dictionary comprising words that match the [0-9]+ regular expression in both vocabularies (e.g., 1-1, 1992-1992...) to achieve this. The underlying notion assumes that these terms possess identical meanings in both corpora, enabling them to serve as guides during the alignment process. However, even generic words like numerals might exhibit entirely different usages across distinct corpora (e.g., "1996" could represent a year in one context and the price of a product in another). While most of the early works assumed some, albeit minimal, amount of parallel data [245, 74, 101], some fully-unsupervised methods have been shown to perform on par with their supervised counterparts [177, 13, 8]. Although effective, the mappings result from several processing stages, necessitating either precise initial estimates or subsequent refinements after mapping, which may involve addressing the influence of common words on word neighbourhoods. The aforementioned methods have certain limitations. Some require translated corpora, while others depend on non-general assumptions that restrict or diminish the applicability of alignment, as discussed earlier. Moreover, to assess whether two words have the same meaning in two different embedding spaces, none of the presented methods considers their neighbours in their relative vector spaces. This limits the evaluation to their morphological similarity, translation, or the assumption that translations have analogous similarity distributions across spaces. In this research, we select anchors differently from them based on both their translations and the translations of their neighbours in their relative vector spaces. The fundamental concept is that words with comparable

meanings also tend to share their most similar words within their corresponding vector spaces. Finally, it is worth mentioning that some approaches in the literature are aimed at the alignment of contextual embeddings (see, e.g., [354, 15, 5, 207, 307, 42, 174]).

#### Contextual Word Embeddings

**Contextual embeddings**, in contrast to their static counterparts, have proven their capacity to offer more holistic representations of meaning. These models are a type of word representation in which the meaning of a word is influenced by its surrounding context in a sentence. Unlike traditional word embeddings which assign a fixed vector to each word regardless of context, contextual word embeddings capture the dynamic meaning of a word based on its context within a sentence. The most important models are ELMo [277], BERT [168], GPT [286], XL-Net [376], and RoBERTa [210]. Aligning these kinds of models poses a challenge owing to their dynamic nature [307]. By dynamically linking words to their various contexts, context embeddings provide a richer semantic and syntactic representation than traditional context-independent embeddings [277]. One simple approach to leverage this more comprehensive representation involves directly applying existing transfer algorithms to contextual embeddings instead of their static counterparts [344].In this context, each pair of tokens is represented by various vectors, each reflecting its specific contextual characteristics. Even when there's access to supervision in the form of a dictionary, it remains uncertain how to effectively leverage this information for multiple contextual embeddings corresponding to a word translation pair; for example, Xu et al. [371] decide to clusterise each word in its sense and then map them. Moreover, SOTA evaluation tasks are less applicable to such embeddings than classic mono-sense embeddings. For example, for evaluating BERT on a benchmark for semantic similarity, one should consider just one a-contextual meaning for the words in the dataset, not considering the advantage of contextual word embeddings. Hence, we prioritise static embedding alignment due to the extensive research history and benchmarks. This decision is driven by the significant impact of embeddings alignment, as emphasised in Sec. 2.1.1.

### 2.1.3 Proposed method

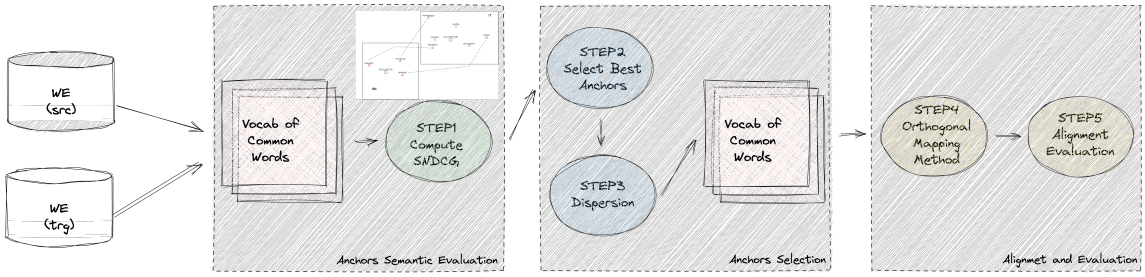



Figure 2.1: Diagram of SeNSe.

As shown in previous sections, several works try to improve the alignment's performance by using different seed lexicons since the transformation's quality depends on the initial dictionary's size and accuracy. To the best of our knowledge, papers proposing unsupervised methods for anchor selection in static embedding alignment have certain limitations, specifically, they do not address the study of semantic similarity among words across corpora. Accordingly, we have devised a method

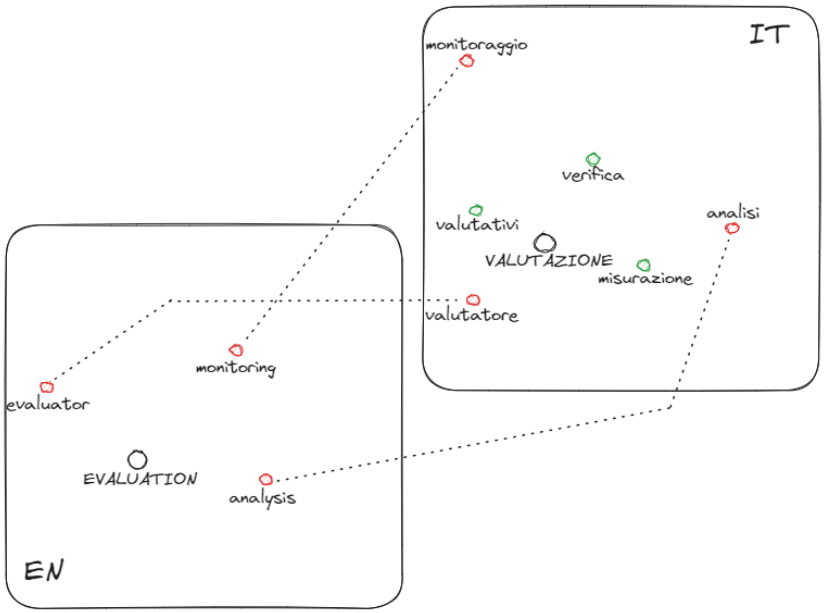



Figure 2.2: Example of $SNDCGscore$. The red points represent the three most similar terms of the word *EVALUATION* in an embedding space trained with English corpus, whereas the green points represent the three most similar terms of the word *VALUTAZIONE* (the translation of *EVALUATION*) in an embedding space trained with Italian corpus. The dotted line indicates the position of the translated term from EN to IT space.

that automatically identifies the optimal anchors for embedding alignment by selecting words that are less likely to exhibit a shift in meaning. To this aim, we consider a potential candidate word and its neighbouring words, which are the closest words in the vector space. Indeed, in the world of static word embeddings, words are typically represented based on their co-occurrence patterns within a large corpus of text. This means that words frequently appearing in similar contexts tend to have similar word embeddings or vector representations. So, when you have a specific word in an embedding space, words that appear near it in that space are likely to be related in meaning or context. This concept is often referred to as distributional semantics, and it's a fundamental idea behind many word embedding models like Word2Vec [245], GloVe [276], and others. Analysing the nearby words or vectors in the embedding space can provide insights into a specific word's meaning and usage. Below we describe the building blocks of SeNSe, depicted in Fig. 2.1, using the case of BLI. The method readily extends to other alignment scenarios, with the sole distinction that for cross-lingual alignment, a Translator, such as online Google Translate (GT), is required to generate anchor candidate pairs. This step is unnecessary when the embeddings share the same language. The process can be streamlined into five key steps:

**Vocab of common words:** In this section, we translate all terms from the source embedding into the target language using GT, followed by a comparison through string matching with the target embedding dictionary. The outcome yields a list of shared word pairs that exist in both vocabularies.

**STEP 1 - Compute SNDCG:** For each couple of the list previously created, we compute a score of semantic similarity using the Normalised Discounted Cumulative Gain (NDCG) score named $SNDCGscore$ = *Semantic Normal Discounted Cumulative Gain*. The computation of this score involves considering both the COS between the potential anchor in the target language and the $n$ most similar words of the potential anchor in the source language when translated into the target language (and vice versa: COS between the potential anchor in the source language and the most similar words of the potential anchor in the target language when translated into the source language), as well as the ranking position of these words. Fig. 2.2 helps to understand the process just explained.

**STEP 2 - Select Best Anchors:** We carefully choose the Best Anchors by retaining only those with a specific threshold value of $SNDCGscore$. Additionally, to account for cases where two source anchors might have the same translation, we deduplicate the list by selecting the pair of anchors with the highest $SNDCGscore$.

**STEP 3 - Dispersion:** To ensure a well-balanced and more evenly distributed in space list of anchors, we filter out the ones that are too close. Our goal is to prevent the alignment from being biased towards specific points in the vector space merely due to the prevalence of highly similar terms. To accomplish this, we meticulously pinpoint closely located anchors within the space and selectively retain the most qualitative ones, thereby maximizing their $SNDCGscores$. By the conclusion of **STEP 3**, we acquire the **seed lexicon**, constituting the definitive list of anchors for executing the alignment.

**STEP 4 - Orthogonal Mapping Methods:** The output of the previous step is the Seed Lexicon that will be used for the alignment through Orthogonal Procrustes (OP). The goal of OP is to learn an orthogonal transformation matrix $Q$ (i.e., $Q^TQ = I_d$, where $I_d$ is the d-dimensional identity matrix), that closest maps $A$ to $B$, namely, $Q^* = argmin_{Q:Q^TQ=I_d}||AQ - B||$. It has been shown that this problem accepts a closed-form solution via Singular Value Decomposition (SVD) [305]. The orthogonality of matrix $Q$ ensures that $AQ$ undergoes only unitary transformations, such as reflection and rotation, thereby preserving the inner product between its word vectors.

**STEP 5 - Alignment Evaluation:** This last step is necessary to evaluate the goodness of the alignment. As discussed in the next section, we evaluated the quality of the bilingual mappings generated with the BLI task.

Let's now describe the process more analytically. Considering two corpora (in this paper, we will consider cross-lingual alignment since it is the most widely used case in the literature despite cross-domain or cross-temporal ones) along with their respective vocabularies $I$ and $J$, we can designate $X$ and $Y$ as the embedding matrices representing each corpus. Here, $X_i$ corresponds to the word vector of the $i$th word in the source language, and $Y_j$ corresponds to the word vector of the $j$th word in the target language (where $i \in I$ and $j \in J$). For each $X_i$ and $Y_j$, where $I_i = J_j$, we compute the set of k-most similar words in the respective embedding spaces, $M_{X_i}$ and $N_{Y_j}$ so that $m_i = \{m \in M_{X_i}\}$ and $n_j = \{n \in N_{Y_j}\}$. In the context of handling corpora with diverse languages, we undertake the process of translating anchor words and the k-most similar words from the source language to the target language and vice versa; so we will have $i$th word translated in $i_T$ and the $j$th in $j_T$, and we will compute the k-most similar words where $I_i = J_{j_T}$ or $I_{i_T} = J_j$. As a final step, for each couple, we compute the $SNDCGscore_{i_j}$ (Eq. 2.1), to measure the quality of the possible

anchor.

$$R = \sum_{k=1}^{K} \frac{cos(Y_j, M_{X_{i_{T_k}}}) + cos(X_i, M_{Y_{j_{T_k}}})}{log_2(k+1)} \tag{2.1}$$

Then, we select the list of anchors with the best $SNDCGscores$. First, we delete all anchors that have $SNDCGscore <= 0$, then we look at the distribution of $SNDCGscore$ values, and we compute the value $\epsilon = f(\sigma)$, where $\sigma$ is the percentage of the left tail that we want to cut. So we will have $d$ pair of $i$ and $j$ that have a sufficient $(\epsilon)$ $SNDCGscore$ so that we can create a dictionary $D = \{\delta \in \mathcal{I}, \delta \in \mathcal{J} \ | SNDCGscore_\delta > \epsilon\}$. The spatial arrangement of chosen anchors in the embedding space is not predetermined, and there is a potential for most of anchors to aggregate in a specific region, resulting in limited terms available for alignment in other areas. In response to this challenge, we have introduced a control mechanism to avert the focal concentration of anchors within the embedding space. In particular, we develop a dispersion logic so that starting from the anchor with the highest $SNDCGscore$, for each $s \in D, s = 1, ..., d$ we compute the similarity score $sim_{s,s+1} = cos(s, s+1)$. We delete all elements when $sim_{s,s+1} >= \gamma$, where $\gamma$ is an arbitrary value. Moreover, since multiple terms in the source language could have the same translation, we deduplicate anchors selecting just the candidate term with the highest $SNDCGscore$. After this process, it is possible to proceed with the alignment searching for a transformation matrix $W$ such that $WX_\delta$ approximates $Y_\delta$ through a simple optimisation problem (Eq. 2.2):

$$\min_{\mathbf{W}} \sum_{\delta=1}^{d} ||WX_\delta - Y_\delta||^2 \tag{2.2}$$

Hence, we establish word similarity based on two criteria: the shared morphology after translation and the presence of identical neighbouring words in the embedding space. This dual criterion guarantees the selection of anchors with the highest semantic similarity across both corpora. Using only the morphology is insufficient because individual words can hold different meanings in the two domains or languages. Therefore, it is essential to devise a versatile method capable of handling diverse scenarios effectively. After defining the anchors, a linear transformation is employed as a mapping function for alignment. This transformation aims to minimise the sum of squared Euclidean distances for the entries in the anchor dictionary. Most modern works like [177], [307], try to align spaces in an iterative way using adversarial learning but this implies a huge amount of computational resources and less transparency. Finally, we present our algorithm termed *Semantic anchors Selection* (SeNSe) as follows:

As it is possible to notice from the description of the method, there are a lot of parameters to set for the optimisation of the algorithm:

- $K$ = the number of most similar terms related to the possible anchors used to compute the $SNDCGscore$.
- $\sigma$ = represents the proportion of top anchors to be considered for determining the $\epsilon$ limit value based on the $SNDCGscore$.
- $\gamma$ = dispersion coefficient used to decrease the number of terms concentrated in the same portion of the vector space.

The above mentioned method could be applied in any domain, requiring only two static embedding models (as the input can be a simple text file containing pairs of words and their corresponding vector representations) and a translation tool.

**Algorithm 1:** SeNSe($\mathcal{I}, \mathbf{Y}$)

**translate** $I \rightarrow I_t$
  **for** $\delta \in I_t \cap Y$ **do**
    $SNDCGscore_\delta$
  **end**
  $SeedLexicon \leftarrow \delta \in I_t \cap Y | SNDCGscore_\delta > \epsilon$;
  $OrderSeedLexiconbySNDCGscore$;
  **for** $s \in SeedLexicon$ **do**
    **for** $s+1 \in SeedLexicon$ **do**
      **if** $cos(X_s, X_{s+1}) \geq \gamma$ **then**
        $SeedLexicon' \leftarrow SeedLexicon \setminus \{X_{s+1}\}$
      **end**
    **end**
  **end**
**end**
**return** $SeedLexicon'$;

### 2.1.4 Evaluation

#### Experimental setting

Following common practice, we evaluate the quality of the bilingual mappings for the BLI task, i.e., a word translation task that measures the accuracy of the induced dictionary compared to a gold standard. For this task, we compare the Precision@1 of SeNSe with previously reported SOTA results on benchmark datasets [74, 14, 177]. We report results on the widely used, publicly available dataset VecMap. The dataset in question was initially provided by Dinu and Baroni[74], and it has since been expanded through subsequent extensions[14, 12]. It includes bilingual dictionaries that map words from English (en) to four languages: German (de), Finnish (fi), Italian (it), and Spanish (es). For a comprehensive description of the experimental settings concerning this BLI task, refer to Artetxe et al. [11]. Several previous works evaluate their method on favourable conditions. Specifically, prevailing unsupervised methods have primarily undergone testing on comparable corpora, such as Wikipedia, which provides a strong cross-lingual signal not readily available in strictly unsupervised scenarios. Furthermore, certain datasets contain unusually small embeddings, consisting of merely 50 dimensions and around 5,000 to 10,000 vocabulary items [383].

In a noteworthy exception, Lample et al.[177] demonstrate favourable outcomes on the English-Italian dataset VecMap. These results are reported alongside their primary experiments conducted on Wikipedia. Although this dataset exclusively employs monolingual corpora, it remains applicable to a pair of relatively similar Indo-European languages. To gain a comprehensive understanding of how our method compares to previous approaches under various conditions, including more demanding scenarios, we conduct our experiments on the well-established dataset introduced by Dinu et al. [74] and its subsequent extensions by Artetxe et al. [13]. Specifically, the dataset contains 300-dimensional CBOW embeddings trained on WacKy crawling corpora for English, Italian, and German, Common Crawl for Finnish, and WMT News Crawl for Spanish. The gold standards for evaluation were derived from dictionaries constructed using Europarl word alignments and accessible

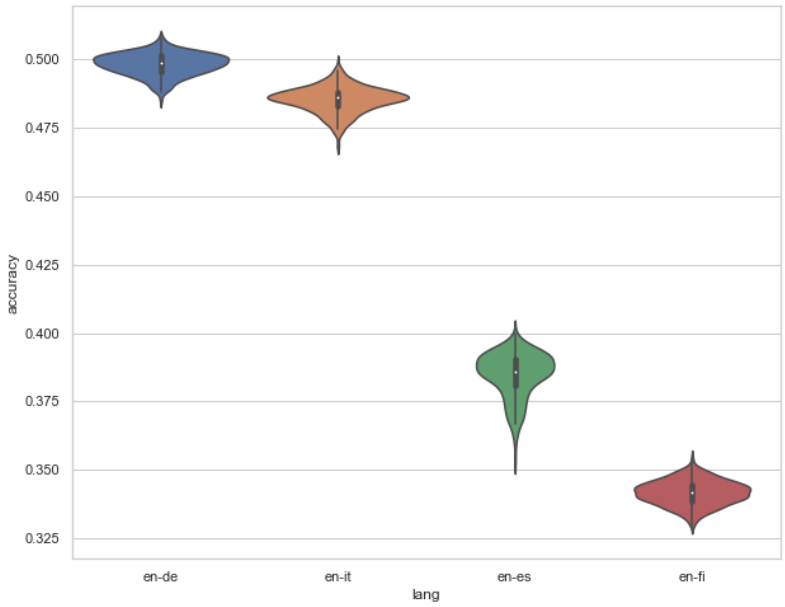



Figure 2.3: Violin Plot of the grid search analysis. We present the accuracy values of the BLI for each bilingual couple about various hyperparameter values.

at OPUS. Tiedemann et al. [335], partition the data into two sets, namely a test set containing 1,500 entries and a training set containing 5,000 entries. The training set remains unused due to the unsupervised nature of SeNSe. The focus of SeNSe on those four languages is to allow for reproducibility and comparability with SOTA benchmarks, as shown in the experimental evaluation. The SeNSe approach is language-independent as it can be applied to any language. To perform BLI we utilise the Cross-Lingual Similarity Scaling (CSLS) technique, which is a distance metric between embeddings needed to compute nearest neighbours. When calculating nearest neighbours in high-dimensional spaces, common distance metrics like Euclidean distance or COS can give rise to the hubness problem [289, 74]. Specifically, certain words are highly probable to serve as the closest neighbours for numerous other words (hubs), while some words may not be the nearest neighbour to any word. Researchers have tackled this issue in the literature by creating alternative distance metrics, for instance, the inverted softmax [319] or the CSLS [177]. As previously stated, we employ the CSLS similarity as a seamless substitute for COS whenever a distance metric is required, and its calculation is performed as follows (Eq. 2.3):

$$CSLS(x, y) = 2cos(x, y) - \frac{1}{n} \sum_{y' \in N_{Y(x)}} cos(x, y') - \frac{1}{n} \sum_{x' \in N_{X(y)}} cos(x', y) \tag{2.3}$$

where $N_{Y(x)}$ is the set of n nearest neighbours of $x$ in the embedding space that $y$ comes from: $Y = \{y_1, ..., y_{|Y|}\}$, and vice versa for $N_{X(y)}$. In practice, we use n = 10. Numerous studies have demonstrated that this method outperforms other approaches in alleviating the hubness problem [289, 74] when conducting searches in high-dimensional spaces.

| lang | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|
| en-de | 49.81% | 00.40% | 48.47% | 49.53% | 49.87% | 50.13% | 50.80% |
| en-es | 38.48% | 00.82% | 35.33% | 38.07% | 38.60% | 39.07% | 40.00% |
| en-fi | 34.18% | 00.44% | 32.93% | 33.85% | 34.20% | 34.48% | 35.46% |
| en-it | 48.53% | 00.47% | 46.80% | 48.27% | 48.60% | 48.80% | 49.87% |

Table 2.1: mean, standard deviation (std), minimum value (min), first quartile (25%), median (50%), third quartile (75%) and maximum value (max) of accuracy for each couple of languages considering all hyperparameters cited above

**Hyperparameters**

We present the parameter values utilised for conducting the experiments, as described in Sec. 2.1.3. In Fig. 2.3 we report a Violin Plot of the grid search analysis performed over the 3 hyperparameters of the model, i.e. the number of most similar terms used $K$, the proportion of top anchors to consider $\sigma$, and the dispersion coefficient $\gamma$, used to filter-out terms that are too close in the vector space. For $K$ we search among the 9 values ($K \in \{5, 10, 15, 20, 25, 30, 35, 40, 45\}$), which means that to evaluate possible anchors, we compute the $SNDCGscore$, with the semantic neighbour of at least their 5 most similar terms and at most the 45 most similar. For $\sigma$ among the 13 ($\sigma \in \{0, 0.03, 0.05, 0.08, 0.1, 0.13, 0.15, 0.18, 0.2, 0.23, 0.25, 0.28, 0.3\}$), meaning that, after the best anchors are found, we discard the ones with lowest quality in terms of $SNDCGscore$, in a percentage ranging from 0% to 30%. The ranges for $K$ and $\sigma$ were determined empirically through preliminary experiments. It was observed that the performance of SeNSe began to decrease significantly after reaching the boundary values. Given computational constraints, a comprehensive grid search across the entire spectrum of values was deemed impractical. Then we selected 5 values for $\gamma$: ($\gamma \in \{\mu - 2std, \mu - std, \mu, \mu + std, \mu + 2std\}$) where $\mu$ and $std$ are respectively the mean and the standard deviation of the normalised distribution of COS among all the anchors chosen so far. In this case, we selected values according to the range rule of thumbs, representing respectively the top $95\%, 84\%, 50\%, 16\%$ and $5\%$ of the data. The number of resulting run is $9 \cdot 13 \cdot 5 = 585$ for each couple of languages. Descriptive statistics are reported in Tab. 2.1. The selected parameter values obtained through a grid search are as follows: $K = 35$, $\sigma = 0.13$, and $\gamma =$ the mean of the COS of the most similar term for each vocabulary. Our method is implemented in Python using NumPy and Gensim library. The experiments were conducted on an Intel Xeon 48CPU machine and required 10 GB RAM running for 171.4 sec on average for performing the anchor selection task. We compare our method with principal competitors and we report experiments performed by [11, 153, 19].

### 2.1.5 Results and discussion

In this section, we first compare SeNSe against SOTA alignment algorithms, then present a real-world application for a qualitative evaluation of the results.

**Experimental results against SOTA algorithms**

The comparison against the SOTA is twofold. The first leg is a typical benchmark task for word embedding alignment, namely bilingual lexicon induction. This task is meant to quantify vocabulary induction performance, that is, given a benchmark set of words in a source language, find the translation in the target language using the CSLS measure in the aligned space. The results of SeNSe and SOTA models on the four language pairs of the benchmark VecMap dataset are reported in Tab. 2.2 in terms of accuracy[3]

| Method | EN-IT | EN-DE | EN-FI | EN-ES |
|---|---|---|---|---|
| Supervised | | | | |
| Mikolov et al. (2013) [245] | 34.93% | 35.00% | 25.91% | 27.73% |
| Faruqui et al. (2014) [92] | 38.40% | 37.13% | 27.60% | 26.80% |
| Shigeto et al. (2015) [316] | 41.53% | 43.07% | 31.04% | 33.73% |
| Dinu et al. (2015) [74] | 37.7% | 38.93% | 29.14% | 30.40% |
| Lazaridou et al. (2015) [178] | 40.2% | - | - | - |
| Xing et al. (2015) [366] | 36.87% | 41.27% | 28.23% | 31.20% |
| Zhang et al. (2016) [101] | 36.73% | 40.80% | 28.16% | 31.07% |
| Artetxe et al. (2016) [14] | 39.27% | 41.87% | 30.62% | 31.40% |
| Artetxe et al. (2017), 25 dict. [13] | 37.27% | 39.60% | 28.16% | - |
| Artetxe et al. (2017), 5k dict. [13] | 39.67% | 40.87% | 28.72% | - |
| Smith et al. (2017) [319] | 43.1% | 43.33% | 29.42% | 35.13% |
| Conneau et al., Procrustes-CSLS(2018) [176] | 44.9% | 46.5% | 33.5% | 35.1% |
| Artetxe et al. (2018) [12] | 45.27% | 44.13% | 32.94% | 36.60% |
| Doval et al., $MSF$ (2018) [75] | 47.7% | 47.5% | 35.4% | 38.7% |
| Doval et al., $MSF_{\mu}$ (2018) [75] | 48.4% | 47.7% | 34.7% | 38.9% |
| Kementchedjhieva et al. (2018) [167] | 45.3% | 48.5% | 31.4% | - |
| Jawanpuria et al. $GeoMM$(2019) [153] | 48.3% | 49.3% | 36.1% | 39.3% |
| Jawanpuria et al. $GeoMM_{multi}$(2019) [153] | 48.7% | 49.1% | 36.0% | 39.0% |
| Unsupervised | | | | |
| Zhang et al., $\lambda = 1$ (2017) [383] | 0.00% | 0.00% | 0.00% | 0.00% |
| Zhang et al., $\lambda = 10$ (2017) [383] | 0.00% | 0.00% | 0.01% | 0.01% |
| Smith et al. (2017) [319], cognates | 39.9% | - | - | - |
| Artetxe et al. (2017) [13], num. | 39.40% | 40.27% | 26.47% | - |
| Conneau et al. Adv - Refine-CSLS(2018) [177] | 45.15% | 46.83% | 0.38% | 35.38% |
| Artetxe et al. (2018) [11] | 48.13% | 48.19% | 32.63% | 37.33% |
| Azpiazu et al. (2020) [19] | 49.02% | 48.18% | 34.82% | **42.15%** |
| SeNSe | **49.66%** | **50.60%** | **35.04%** | 39.47% |

Table 2.2: Accuracy (%) of the proposed method is compared to previous work. The results are obtained using the framework presented in [19, 11, 153]. The remaining results were reported in the original papers.

[3]SOTA results from [19]. For unsupervised methods, the average accuracy across 10 runs is reported. Unlike SOTA unsupervised methods, SeNSe does not possess random parameters. Therefore, the results of a single run are reported since they remain consistent upon re-running the method.

| Method | EN-IT | EN-DE | EN-FI | EN-ES |
|---|---|---|---|---|
| Precision@1 | | | | |
| Artetxe et al. (2018) [11] | 48.13% | 48.19% | 32.63% | 37.33% |
| SeNSe | **49.66%** | **50.60%** | **35.04%** | **39.33%** |
| Precision@5 | | | | |
| Artetxe et al. (2018) [11] | 63.07% | 65.33% | 51.62% | 52.13% |
| SeNSe | **63.13%** | **68.13%** | **54.92%** | **55.27%** |
| Precision@10 | | | | |
| Artetxe et al. (2018) [11] | 67.13% | 70.33% | 58.71% | 56.93% |
| SeNSe | **68.67%** | **73.03%** | **59.97%** | **61.00%** |

Table 2.3: The accuracy (%) achieved by our proposed method, as compared to Artetxe et al.'s approach [11] with Precision@1, @5, @10

In most cases, SeNSe achieves the best performances among all unsupervised strategies, even improving over SOTA-supervised models in some cases (3 over 4). Our model shows an improvement most noticeable for Italian and German, where SeNSe obtains an improvement of almost 1% and 3%, respectively. Despite being entirely unsupervised, our method outperforms previous supervised approaches and achieves the best results in three out of four language pairs. The only exception is English-Spanish, where Azpiazu et al. [19] improve. This last observation leads to a deep focus on this model, the best-unsupervised model that overcomes our performance just in one language [19]; indeed, it holds a big difference in its nature from our method. Our method needs just the embeddings of the two interested languages to be trained, while Azpiazu et al. [19] needs many word embeddings in different languages to enact its hierarchical mapping logic. This is a major weakness of the model, as it is not always easy to have data of the domain concerned and consistent in quality and numerosity. The vast majority of current benchmarks for word embedding alignment are evaluated on the accuracy of the BLI problem. The drawback of this method lies in its evaluation process, which solely determines if the first retrieved target vector matches the source vector. It overlooks the distance between the target term and its translation, resulting in an evaluation that assumes only binary values of 0 or 1. In other words, the distance of a word from its semantic neighbourhood is not properly considered. However, for several tasks, like similarity, categorisation, or LSC detection, having a source word mapped in the correct neighbour in the target language is more important than BLI accuracy. The intuition behind this idea is that the semantic neighbourhood of a word better defines its meaning in its context than the representation of the word itself. This, indeed, was one of the main motivations of SeNSe. We conduct a second evaluation to assess SeNSe's performance against the SOTA. It utilises **Precision@5** and **Precision@10**, which consider the 5 and 10 neighbouring elements of the word, respectively. The hyperparameters remain consistent with those mentioned earlier. We compare our method with Artetxe et al. [11] since it is the best-unsupervised model reproducible; indeed, we were not able to apply Azpiazu et al. [19] and also, as mentioned before, it requires embeddings in 107 languages to be reproduced. In Tab. 2.3 we reported precision@1, precision@5, and precision@10 for each dataset with both models. The result shows that SeNSe outperforms the SOTA on all the datasets and metrics considered. More specifically, In Tab. 2.3 we can see that, using the **Precision@5** and **Precision@10** as metrics,

the difference between SeNSe and the best SOTA method [11] increases considerably for all the benchmarks. This finding seems to confirm our intuition that SeNSe can align words of the source language with the correct semantic area in the target vector space.

### Application on a real scenario

In order to perform a qualitative evaluation of SeNSe, we applied it to a real example in the European Labour Market domain, as part of an ongoing H-2020 Project H2020-SC6-TRANSFORMATIONS-2018-2019-2020. In this task, we generate a variety of vector representations of the domain-specific text corpus through fastText with the following parameters: *model=CBOW*, $dim = 100$, $epochs = 200$ and $learning\ rate = 0.1$. Our experiments use a large corpus of job ads collected from different online sources. Within this dataset, we selected all the online job vacancies published in the source and target languages from the year 2019 to 2022. The following steps preprocessed titles: (1) tokenisation, (2) lower case reduction, (3) punctuation and stopwords removal (using a list of English stop-words taken from NLTK, (4) n-grams computation, (5) removal of short words (less than three characters). For task (4) we employed the Gensim phrase model for bi-grams, tri-grams, and four-grams, filtering out words' n-grams with a count lower than five and a threshold smaller than 0.5. This threshold represents the limit above which an n-gram should be accepted. The higher the threshold, the smaller the number of n-grams. We employed the bi-gram scoring function described in [246], and we repeated the process to obtain tri-grams and four-grams. Task (5) was performed after task (4) since some words of length lower than three may form n-grams relevant to our analysis (e.g. *it_manager*, *ui_developer*, etc.).

| Id Number | Occupation in EN | Id Number | Occupation in EN | Id Number | Occupation in EN |
|---|---|---|---|---|---|
| 1 | data analyst | 16 | store manager | 31 | welder |
| 2 | mechanical engineer | 17 | biomedical engineer | 32 | plumber |
| 3 | data science | 18 | radiologist | 33 | human resources officer |
| 4 | pharmacist | 19 | accountant | 34 | biologist |
| 5 | nurse | 20 | chef | 35 | nutritionist |
| 6 | surgeon | 21 | butcher | 36 | biochemical |
| 7 | lawyer | 22 | warehouse worker | 37 | physic |
| 8 | informatic | 23 | forklift driver | 38 | real estate agent |
| 9 | bricklayer | 24 | civil engineer | 39 | cnc turner |
| 10 | interiors designer | 25 | dressmaker | 40 | bartender |
| 11 | logistics supervisor | 26 | hairdresser | 41 | cyber security |
| 12 | electrician | 27 | cashier | 42 | veterinarian |
| 13 | graphic designer | 28 | customer care | 43 | physiotherapist |
| 14 | software developer | 29 | statistician | 44 | cad designer |
| 15 | architect | 30 | driver | 45 | dishwasher |

Table 2.4: Legend of Identification number in Fig. 2.4, 2.5, 2.6

We selected 45 occupations, as indicated in Tab. 2.4, for which we possess information about their translation from the target language to the source language. Subsequently, we confirmed

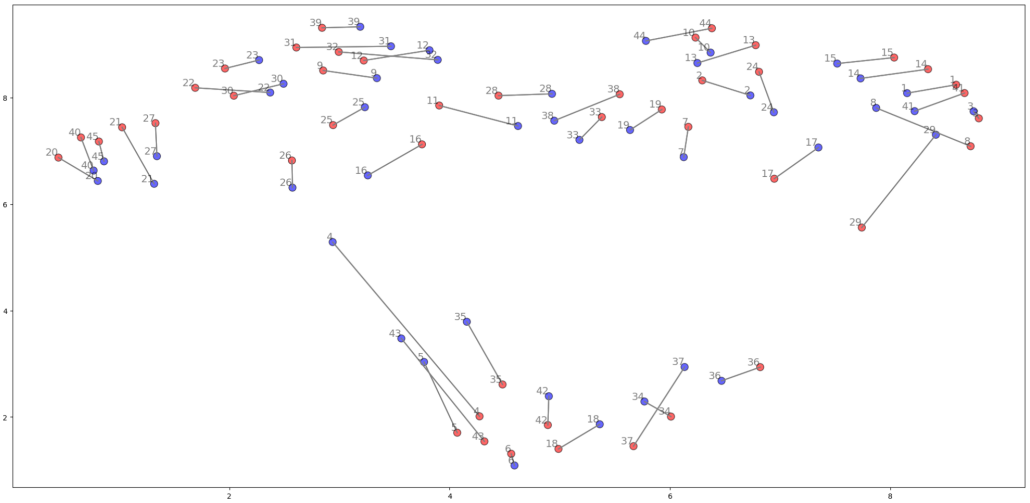


Figure 2.4: UMAP plot of LM Embeddings aligned using SeNSe. As reported in Table, an individual Identification Number is allocated to each icon, which corresponds to a specific occupation. 2.4

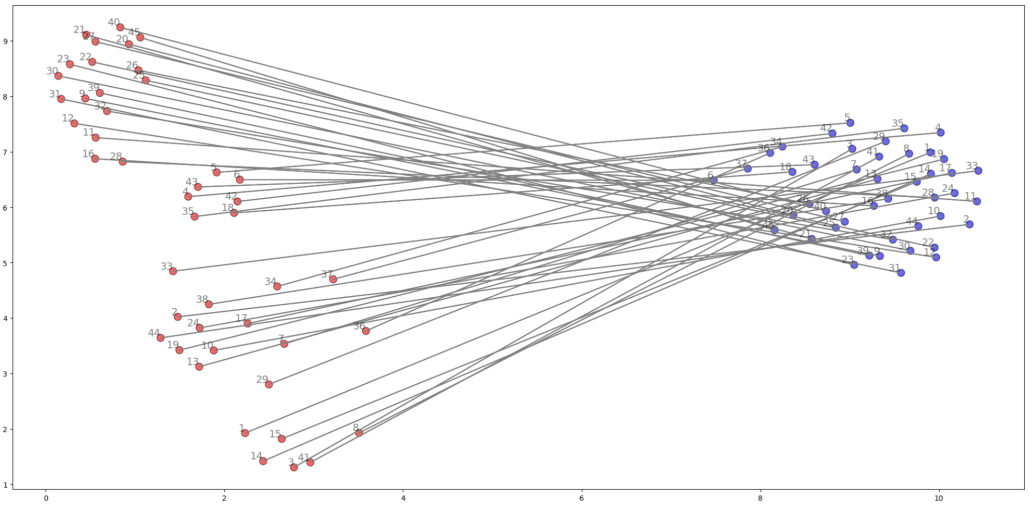


Figure 2.5: UMAP plot of LM Embeddings aligned using Artetxe et al. [11]. As reported in Table, an individual Identification Number is allocated to each icon, which corresponds to a specific occupation. 2.4

whether the two aligned embeddings indeed displayed the same translated term nearby. To better visualise the results of the method, in Fig. 2.4 we provide a scatter plot produced over the resulting embedding aligned from the source language to the target one, generated using UMAP (Uniform Manifold Approximation and Projection) that is a dimension reduction technique that not only serves as a visualization tool similar to t-SNE but also enables general non-linear dimensionality

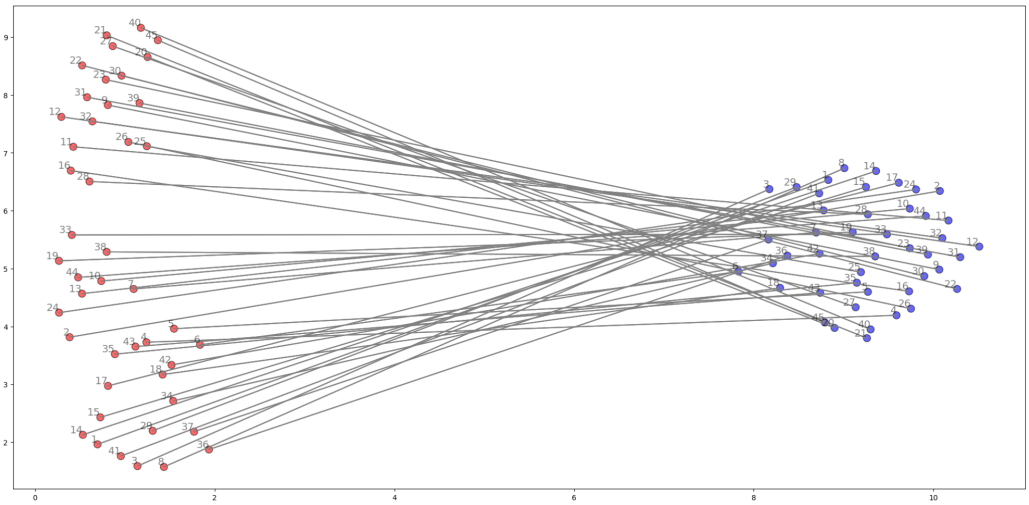


Figure 2.6: UMAP plot of LM Embeddings not aligned. As reported in Table, an individual Identification Number is allocated to each icon, which corresponds to a specific occupation. 2.4

reduction [238]. For comparison, we present two additional scatter plots of the vectors for the selected professions: one (Fig. 2.5) obtained after aligning the two models using Artetxe et al. [11], and the other without any alignment method applied (Fig. 2.6). Upon examining the results, it becomes evident that SeNSe exhibits a higher level of precision in aligning the two vector spaces compared to Artetxe et al. [11], thus providing a genuine basis for comparison in the analysis of the source and target labour markets. It is interesting to notice that the alignment method proposed by Artetxe et al. [11] maintains a clear separation between the source and target instances, which are respectively on the left and the right end of Fig. 2.5. The aligned source terms are shifted to be closer to their counterpart in the target space but remain close to all the other source instances. This makes the separation between semantic areas in the two spaces less clear. To the contrary, in Fig. 2.4, we can see that words belonging to the same semantic area are close in the aligned embedding space, even tough they belong to different origins (i.e. source and target vector spaces). For instance, in the bottom right corner, we can see, close to each other, the terms *biochemical*, *biologists*, and *physic* in both the source and target embeddings. This validates the effectiveness of SeNSe in accurately capturing the semantic nuances of words during the alignment process and making the final alignment more robust.

### 2.1.6 Conclusions and Future Works

In this work, we proposed SeNSe, a novel approach for selecting anchors for embedding alignment. The main advantage of our method resides in the flexibility of the application tied to the robust logic used; indeed, none of the existing works choose anchors for aligning static embeddings while taking into account the semantic neighbours of the candidate words. We showed that the alignment produced in this fashion outperforms the benchmark in most of the cases, in all of them if we consider only resource-poor methods like SeNSe. Moreover, testing with **Precision@5** and **Precision@10** the fit of the source word in its target neighbours, we showed that SeNSe outperforms the SOTA

on every benchmark dataset. This model has some limitations, such as its potential ineffectiveness in aligning embeddings across multiple languages without a pivot language. Another limitation pertains to its applicability solely within the static word embedding domain. Consequently, we are actively working on applying the model to achieve contextual embedding alignment. The methodology presented draws inspiration from a real-world project within the LMI and caters to the specific tasks at hand. A real example in this field has been utilised to qualitatively evaluate the proposed method. Finally, the model is implemented and available to the whole community as an off-the-shelf Python tool on GitHub.

## 2.2 Alignment of Multilingual Embeddings to Estimate Job Similarities in Online Labour Market

In recent years, word embeddings have proven relevant for studying differences and similarities among job professions and skills required by the labour market across countries, providing valuable insights about the labour market dynamics to support policy and decision-making. In such a scenario, aligning word embeddings constructed across different countries and languages becomes key to allowing experts to reason on the labour market, catching technological and cultural shifts across borders. This paper proposes MEAL, an unsupervised method for aligning domain-specific monolingual embeddings trained with Online Job Advertisement (OJA)s data. Our approach is based on SeNSe[221] and selects a seed lexicon of anchors, i.e. words with the same meaning in both corpora that will be used as pivots in the alignment, without assuming a priori semantic similarities. Indeed, exactly as SeNSe, to asses this relationship MEAL takes into account the semantic similarity between the neighbour of the two words in the word embedding space. Particularly, it chooses optimal anchors that are less susceptible to meaning shift. We deploy MEAL within the research framework of a European H-2020 Project that aims to use Artificial Intelligence (AI) technologies to predict the future of the European labour market. Specifically, we apply it to the embeddings we train on 7+ millions of OJAs collected in 2022. As a main outcome, MEAL allows stakeholders and policymakers (i) to estimate job similarities in Online Labour Markets across Europe, facilitating the assessment of how well these markets align with the taxonomy outlined by the official European Skills and Competences taxonomy, and (ii) to obtain indicators to support a data-driven policy design at a very fine-grained territorial level.

### 2.2.1 Introduction

Over the past few years, learning distributed representations of words in a vector space has shown to be extremely helpful in a wide range of applications in LMI, such as extracting insights from job postings [95, 60], analysing skills and occupations demand [384, 113], classifying OJAs and facilitating more precise matching between job seekers and opportunities [127, 115]. However, several essential linguistic tasks require comparing distributed representations of words belonging to different vector spaces within or among domains and languages. For instance, the diverse nature of labour markets across various nations highlights the importance of tailoring solutions to specific regional and occupation-based challenges. By acknowledging and addressing the unique characteristics of each country's labour market, it is possible to improve the efficiency of job matches and reduce the likelihood of mismatches [343, 61]. Malandri et al. [217] deepen this theme by saying that with the rapid pace of technological advancements and globalisation, the labour markets

across various nations have become increasingly diverse, posing the problem of comparing different labour markets. Therefore, a critical question arises: *can we achieve consistency between two distinct word vector models and transfer lexical and semantic knowledge across diverse corpora?* WEs alignment has found fertile ground in research in the last years [298]. The potential to align word embedding spaces is intriguing for several reasons: (i) it allows for comparing word meanings across languages, essential for many NLP tasks such as BLI [383], machine translation [125], mining parallel corpora [177], and cross-lingual IR [159]; (ii) by creating a shared representation space, cross-lingual word embeddings enable model transfer between languages, helping bridge the gap between resource-rich and low-resource languages [298], with applications in text classification [171], sentiment analysis [393], and dependency parsing [9]; (iii) even embeddings trained on the same corpus at different times or with different hyper-parameters result in misaligned vector spaces, complicating comparisons [303] and finally, (iv) enables a more nuanced comparison and interpretation of various models, bridging the gap between the model and the user, potentially enhancing adoption by making the results more accessible and understandable [43]. As mentioned in previous sections, most current approaches focus on constructing a *seed lexicon*, i.e. a collection of words that share the same meaning in both corpora. These words, called anchors, serve as reference points for the process of learning a transformation that maps word embeddings from one space to another by minimising their distances. Numerous researchers have proposed methods for identifying anchors, as elaborated in Sec. 2.1.2. However, as far as we know, previous works consider only the relation between pairs of words to determine if they have the same meaning in the two corpora. Conversely, we propose a method for anchor selection, namely MEAL (Multilingual Embeddings ALignment), that considers the two candidate words and their neighbours in the respective vector spaces. The intuition behind this idea, taken from SeNSe[221], is that the semantic neighbourhood of a word better defines its meaning in its context than solely the representation of the word itself. For instance, using the neighbourhood within the vector can assist in clarifying the meaning of a word in its context, particularly when the word has multiple interpretations. We validated our method against the most performing method applicable in SOTA in a complex evaluation task such as BLI. Subsequently, we applied our method to an LMI real-life project aimed at assigning the most similar professions in different countries' markets to their English counterparts. By applying these techniques to corpora comprising OJAs from diverse countries, we comprehensively understand how professions and skills are conceptualised, enabling a nuanced comparison of differences and similarities.

### Contribution

The contributions of this paper are threefold:

**(1)** The design and implementation of an alignment methodology called MEAL that, as a novelty, performs multilingual word embeddings alignment considering the semantic neighbour of the anchors' selection, then evaluating the approach on the bilingual mappings for the SOTA BLI task;
**(2)** The application and deployment of MEAL in an H2020 project$^{\diamond}$ in the labour market field. The outcomes reveal the approach was useful for estimating the coherence of cross-country online labour market (EN, IT, FR, DE, NL), by processing 7M+ Online Job Ads;
**(3)** MEAL is available to the whole community for being applied in other domains as an off-the-shelf Python tool on GitHub[4].

[4]https://github.com/filippopallucchini/MEAL

### 2.2.2 Related work

In this section, we delve into the primary efforts dedicated to analysing changes in the labour market using AI, exploring key approaches for word embeddings along with their respective alignment methods.

**Labour Market through AI.** As already mentioned, many recent studies delve deep into labour markets using AI techniques, employing advanced methods to analyse and predict trends, offering valuable insights for strategic decision-making in this dynamic field [272]. Boselli et al. [29] classify OJAs through ML and leverage AI to identify and analyse the unique set of skills required for each profession in the labour market. Colombo et al. [60] develop a set of innovative tools for LMI by applying ML techniques to web vacancies in the IT labour market, allowing them to uncover the nuanced variations in skill demands across various occupations, which may indicate a fragmented job market where specific skills are prioritised for particular roles. They leverage an AI-powered platform to comprehensively analyse the unique blend of skills and qualifications required across various occupations within the dynamic labour market, taking into consideration regional variations, industry trends, educational attainment, and levels of professional experience. Fettach et al. [95] developed a system that reveals the dynamic changes in skill sets and job roles across industries amidst ongoing labour market disruptions from technological advancements and changing societal demands. Malandri et al. [217] develop a Web-based tool for automatically enriching the standard occupation and skill taxonomy (European Skills, Competences, Qualifications and Occupations (ESCO)) with new occupation terms extracted from OJAs. The ability of a profession to be interpreted differently across various markets highlights the importance of understanding the unique characteristics of each market. For instance, Malandri et al. [217] were able to understand that while the ICT labour market in Northern Europe has reached a higher level of maturity compared to the South, it influences how the term *ICT specialist* is perceived and used within those regions. This discrepancy underscores the need for a nuanced approach when interpreting professional titles across markets. For the works related to word embedding alignment, you could refer to Sec. 2.1.2.

### 2.2.3 Deploying MEAL

**OJAs corpus:** As part of the H-2020 Project, 7M+ of OJAs published in 2022 from various online portals across 28 European countries were collected[5]. We focused on the OJAs published in 2022 in five specific countries: 838,902 from Germany (DE), 1,045,346 from France (FR), 520,613 from Italy (IT), 603,862 from Netherlands (NL), 971,935 from the United Kingdom (EN).
**ESCO taxonomy:** The ESCO taxonomy provides a multilingual European classification of Skills, Competencies, and Occupations relevant to the EU labour market, education, and training. Taxonomies like ESCO help to identify the required skills of workers in a particular job, and they are crucial for accurately tracking changes in labour market demands. The taxonomy provides descriptions of 3008 occupations and 13,890 skills linked to these occupations, translated into 28 languages (all official EU languages plus Icelandic, Norwegian, Ukrainian, and Arabic)[6]. ESCO organises occupations hierarchically for a total of 5 levels, at the highest level there are nine occupational groups, and as you go down the hierarchy the occupations become more specific. We focused on

[5]Source of OJAs: The EUROSTAT Web Intelligence Hub, see https://www.cedefop.europa.eu/en/tools/skills-online-vacancies
[6]https://esco.ec.europa.eu/en/about-esco/what-esco

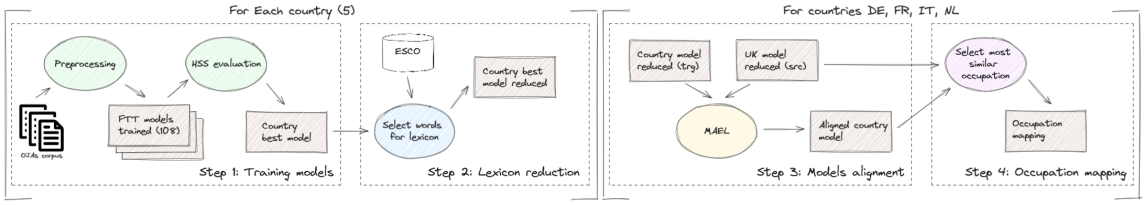



Figure 2.7: MEAL application workflow

the occupations at the fourth level and we used the translations of five specific countries: IT, DE, NL, FR, and EN, which cover the 80% of the European OJAs in the dataset.

### 2.2.4 Applying MEAL

Fig. 2.7 shows the system workflow consisting of three steps:

**STEP 1 - Training models:** Starting from the vacancies corpus, we obtained a vector representation of each of the five countries considered using a word embedding algorithm. Applying a classic preprocessing pipeline to these texts: (1) tokenisation, (2) lower case reduction, (3) punctuation and stopwords removal, and (4) n-grams computation; we obtained the corpus for training the model of word embeddings. We used as models architectures FastText [28] generating 108 models for each country corpus. Hyperparameter selection for each architecture was performed with a grid search over the following parameter sets: algorithm $\in$ {SG, CBOW}, loss function $\in$ {ns, hs} vector size $\in$ {50, 100, 300}, number of epochs $\in$ {10, 50, 100} and learning rate $\in$ {0.01, 0.05, 0.1}. A model evaluation has been performed to select the word embedding that better preserves ESCO taxonomic relations. Specifically, we employed two metrics: COS and Hierarchical Semantic Similarity (HSS) presented in [112], a measure for the semantic similarity between pairs of concepts in a taxonomy. We applied these two metrics to all combinations of occupations in ESCO. For each generated model, we computed the Spearman rank correlation coefficient between the distribution of HSS values and the distribution of COS values. After calculating the correlation for each of the 108 models trained in every country, we aggregated these values across all countries. Then, we selected the model with the highest average correlation as the optimal choice for all countries. The highest average Spearman's rank correlation coefficient value achieved was 0.16, the best model has the following parameters: algorithm = SG, loss function = ns, size = 300, epochs = 50, and learning rate = 0.01. The five models with these best parameters are the ones that were then aligned with MEAL, the labour market application of SeNSe.

**STEP 2 - Lexicon Reduction:** To align the models generated in the previous step, we ensure comparability by constraining the vocabulary size. Specifically, we select only the top 200,000 most frequent terms from various corpora. Additionally, we identified ESCO occupations common across all five corpora and incorporated only the ESCO skills associated with these shared occupations. These new vocabularies were then used as the lexicon for the alignment process, as described in Sec. 2.1.3.

**STEP 3 - Alignment:** We use the EN as a pivot and align the other four country models to it using MEAL. The parameters used for the alignment were identified as the best in Sec. 2.1.4. After aligning the vectors with MEAL and making them comparable, we get a set of vectors for each of the five countries. These vectors will be used in the next step to compare the EN occupation with

that of the other four countries.

**STEP 4 - Occupation Matching:** We utilised the aligned vectors to establish a mapping mechanism between EN occupations and the occupations of other countries. We used COS to measure the degree of similarity. For each of the 376 EN occupations, we selected the most similar occupation in the other countries. An example is given in Tab. 2.5, where for the occupation *2512 - Software developers* we identify the occupations that are most similar to it. Additionally, we have provided the degree of similarity and its English translation. A graphical representation is also provided in Fig. 2.8.

Table 2.5: An example of EN occupation mapping.

| EN occupation | Country | Most similar occupation | Degree of similarity | English translation |
|---|---|---|---|---|
| Software developers | DE | Softwareentwickler | 0.60 | Software developers |
| | FR | Concepteurs et analystes de logiciels, et concepteurs de multimédia non classés ailleurs | 0.43 | Software and applications developers and analysts not elsewhere classified |
| | IT | Sviluppatori di software | 0.55 | Software developers |
| | NL | Softwareontwikkelaars | 0.67 | Software developers |

### 2.2.5 Outcomes to the EU and Result Comments

We conducted a study to explore the correlation between the current state of the labour market and the ESCO taxonomy by using the mapping mechanism described in Sec. 2.2.4. The hierarchical structure of the taxonomy allows performing analyses at different levels of aggregation. Specifically, we focused on analysing both occupational groups, at the highest level, and the individual occupations, at the lowest level. Our analysis was based on EN data and focused on two key questions: ***(Q1) How closely the current labour markets fit with the reference taxonomy?*** and ***(Q2) What are the most representative skills of occupations across labour markets?*** To address answer (Q1), we used ESCO as a reference and determined the number of times the mapping of EN occupation using MEAL matches the ESCO translation (defined as *correct* in this phase). For instance, in Tab. 2.5, DE, IT, and NL provide the *correct* match for *Software developers*, while FR does not. In this case, the ratio of *correct* match for this occupation is 75%. By using this metric, we can compare occupations and countries. The values of *correct* translations are expressed as percentages and are aggregated to the highest level of the ESCO hierarchy, which are the 9 occupational groups[7], as displayed in Fig. 2.9a. In an ideal scenario, where the labour market fits perfectly with the ESCO taxonomy, one would expect that for each occupation in a country, its most similar occupation would be the same occupation translated into the language of that country [116, 222].

Except for group 4, FR has the highest percentage of *correct* occupation matches with respect to the EN, especially in groups 1, 2, and 3. We can assume that the professions in these groups have similar meanings in both countries. On the other hand, DE and IT do not appear to be well-suited for the EN job market, except for group 4. The distribution of COS values for *corrected* matches is displayed in Fig. 2.9b, allowing us to analyse different countries' degrees of fit. For instance, for group 1 in NL, even though the number of *correct* matches is lower than in other countries, the degree of similarity is higher, it is around 0.7, meaning that the few DE occupations that are matched are

[7]Group *6 - Skilled agricultural, forestry, and fishery workers* were excluded as they are not advertised through online job portals.

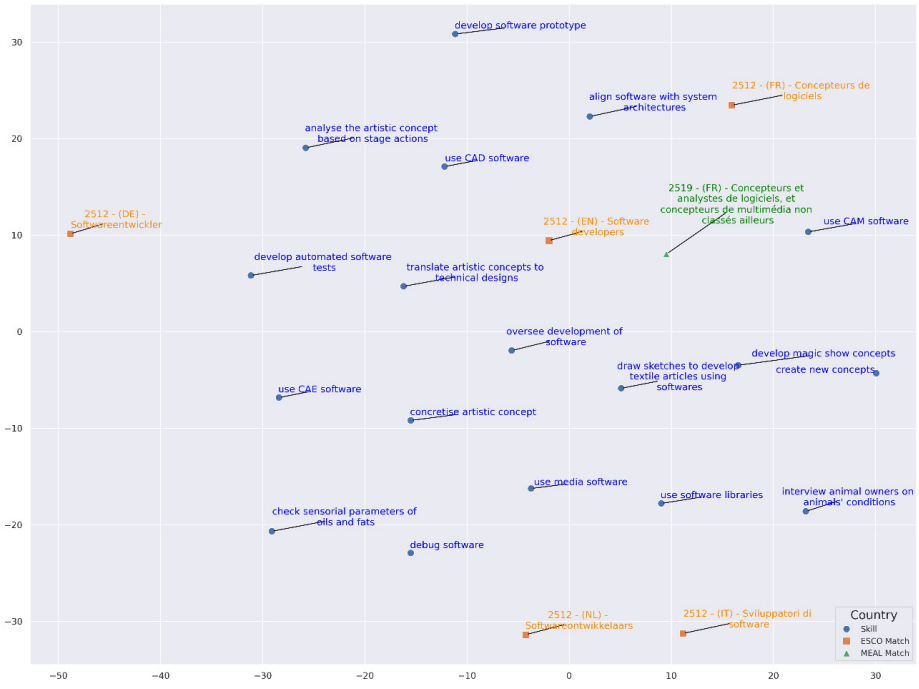
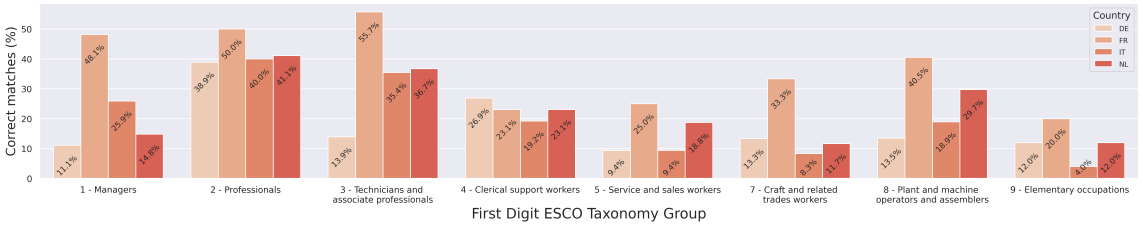
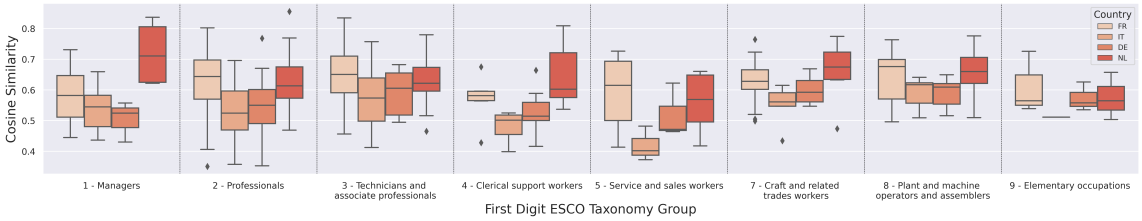



Figure 2.8: Example of cross-country occupation mapping based on embedding alignment. Starting from a set of skills frequently required for *Software Developers* in the EN, we identify the most similar occupations and associated skills across countries. Orange squares represent occupations, blue dots represent skills, and green triangles highlight the closest matching occupation in France—*Concepteurs et analystes de logiciels*.

done so with a high degree of confidence. To address (Q2), we focused on individual occupations and analysed their relationship with skills. We specifically calculated the top 5 most similar skills, based on COS, for each occupation, considering both the related ESCO translation and the matches identified through mapping. We used t-SNE [346], a tool for visualising high-dimensional data that reduces dimensionality while preserving the distances between points within the space. This allowed us to obtain a graphical representation of skills and occupations. Considering the two occupations from FR, one provided by ESCO and the other obtained from MEAL, we can observe which skills they share with the EN *Software Developer*. These include skills related to application and software development, as well as skills specific to multimedia applications. This suggests that the concept of a software developer in the EN is more closely associated with multimedia application development in FR. For the other countries, when examining the skills in their neighbourhood, we notice a specialisation of the software developer role. For instance, in DE and the NL, it appears to adopt a




(a) Comparison of correct matches by occupational group in various countries versus the EN.




(b) Similarity distribution of correct matches by occupational group in various countries versus the EN.

Figure 2.9: Comparing the EN with other countries based on correct matches and cosine similarity distribution.

more engineering-focused connotation, as evidenced by skills such as *use of CAE software*. ESCO is undeniably a valuable resource, offering a standardised definition of occupations and skills for 28 languages and serving as a *lingua franca* for the EU Labour Market. However, these results show that adopting a data-driven tool like MEAL allows one to effectively capture the lexicon used by companies to advertise online positions and, in turn, to assist in crafting data-driven policies at a detailed territorial level, estimating differences across countries and languages as characteristics of a different labour market maturity.

### 2.2.6 Conclusions and Future Works

In this study, we introduced MEAL, an alignment methodology that presents a novel approach to multilingual word embeddings alignment trained specifically for the LMI domain. Applied and deployed within an H2020 project, MEAL processed 7+ million OJAs across five European countries. This analysis elucidated the extent to which the current labour market aligns with the ESCO taxonomy and identified the most representative skills associated with occupations across various labour markets. The main advantage of our method resides in the flexibility of the application tied to the robust logic used; indeed, no other works select anchors considering the semantic neighbour of the candidate words. We showed that the alignment produced in this fashion outperforms the best-unsupervised model reproducible [11] on every benchmark dataset. This model has some limitations, such as its potential ineffectiveness in aligning word embeddings across multiple languages without a pivot language. Another limitation pertains to its applicability solely to static word embeddings. Consequently, we are actively applying the model to achieve contextual word embeddings alignment. As a main outcome of the EU, MEAL represents a framework to study OJAs and to produce quality-quantitative evidence on labour market shifts across countries and languages, focusing on the language used by companies to advertise job positions and their skills.

Finally, to allow for the reproducibility of MEAL to other domains, the approach is implemented and available to the whole community as an off-the-shelf Python tool on GitHub.

# Chapter 3

# Lost in Alignment: A Survey on Cross-lingual Alignment Methods for Contextualised Representation

Cross-lingual word representations allow us to analyse word meanings across diverse language settings. It is crucial in aiding cross-lingual knowledge transfer when constructing NLP models for languages with limited resources. This survey presents a comprehensive classification of cross-lingual contextual embedding models. We assess their data requirements and objective functions, and we introduce a taxonomy for categorising these approaches. Then, we present a comprehensive table containing a set of hierarchical criteria to compare them better, along with information regarding the availability of code and data to enable replication of the research. Furthermore, we delve into the evaluation methodologies employed for cross-lingual embeddings, exploring their practical applications and addressing their current associated challenges.

## 3.1 Introduction

Word embeddings are dense, low-dimensional vector representations of words that encode semantic and syntactic information based on their distributional properties in large text corpora. They are typically categorized into two main types: static embeddings (e.g., word2vec [246], GloVe [276]), which assign a single, context-independent vector to each word based on co-occurrence statistics or shallow neural networks; and contextual embeddings (e.g., ELMo [278], BERT [71]), which generate dynamic, context-sensitive vectors using deep neural architectures such as LSTMs or Transformers. While static embeddings capture general word similarity, contextual embeddings can disambiguate word senses depending on their usage in a sentence.

The cross-lingual alignment problem is a specific aspect of the broader challenge of cross-lingual NLP, introduced by Mikolov et al. [245] in 2013. It refers to mapping or aligning embeddings or representations from different languages into a shared or common vector space such that the representations of words or phrases in these languages are directly comparable. The potential to

align embedding spaces is relevant in the field of computational language for multiple reasons: (i) it enables us to compare the meanings of words across languages, a critical aspect for numerous NLP tasks, including, but not limited to, BLI [383], machine translation [125], mining parallel corpora [177], cross-lingual IR [159]; (ii) it facilitates model transfer across languages, bridging the gap between resource-rich and low-resource languages [298]. Some examples of applications are text classification [171], sentiment analysis [393], and dependency parsing [9]. Finally, (iii) even embeddings trained on the same corpus but in different phases produce misaligned vector spaces, making it impracticable to compare word vectors trained with different hyperparameter settings or at different times [303]. To clarify the matter, look at Chapter 2. Instead of aligning already trained embedding, it is also possible to directly train a multilingual model; numerous researchers proposed cross-lingual word representations [171, 245] that create a shared embedding space for words across two (bilingual embeddings) or more languages (multilingual embeddings). However, multilingual static models often contain inaccuracies, primarily due to the lack of context consideration. Despite this, there is some work applied to static embedding that is important to mention because it lays the foundation for some methods developed later, like [11, 177]. Most of these approaches for static embedding alignment primarily focus on constructing a "seed lexicon", which is essentially a collection of words that share the same meaning in both corpora. These words, commonly referred to as anchors, serve as reference points for learning a transformation that maps embeddings from one space to another by minimising their distances. Context-free embeddings, like word2vec [246], generate a fixed representation for each word regardless of its context, capturing only general meanings. In contrast, contextualised embeddings, introduced with models like ELMo [277] and later advanced with BERT, adapt word representations based on surrounding words, allowing for nuanced interpretations of each word's meaning within specific contexts. With the advent of contextual embedding, the drawbacks of multilingual models decrease significantly because each language carries a different context that this model has the ability to embed. Indeed, the contextual nature of the embeddings means that they capture the meaning of individual words and the meaning of the surrounding context, making them particularly useful for natural language processing tasks [307]. So, researchers have started proposing contextualised multilingual models. One such instance is mBERT introduced by Devlin et al. in 2019 [71]. It is a single language model pre-trained on the combination of monolingual Wikipedia corpora sourced from 104 different languages. mBERT simplifies zero-shot cross-lingual model transfer; Pires et al. [281] fine-tuned the model on task-specific supervised data from one language and evaluated it in another, showcasing how the model generalises across languages effectively. Their experiments demonstrated mBERT's remarkable cross-lingual generalisation capabilities, even across languages with no lexical overlap. This suggests that mBERT can effectively capture multilingual representations. In contrast to monolingual BERT, which lacks zero-shot transfer capabilities, mBERT distinguishes itself in two main ways: (1) during pre-training, specifically in the masked word prediction task, each batch comprises sentences from all languages, and (2) it adopts a unified vocabulary generated by applying WordPiece to the combined monolingual corpora [71]. But can multilingual contextual embeddings effectively address the cross-lingual problem? These methods can only learn implicitly the correspondence between words and structures across languages in a purely unsupervised manner. Indeed, Blevins et al. [27] could even quantify how *multilingual* English pre-trained models are. While those multilingual models may demonstrate significant empirical effectiveness [30], the weaknesses have emerged constantly in the literature. Indeed, those models work best for typologically similar languages, indicating that they can map learned structures to new vocabularies but don't systematically transform those structures to accommodate languages with different word

orders. This latter insight is reinforced by the work of Wu et al. [365] who show that language representations are not correctly aligned in mBERT, but can be linearly re-mapped; in particular, they built a contrastive alignment objective that can better utilise bitext signal. Furthermore, these models may encounter challenges related to polysemy and homonymy. This entails the difficulty of managing words with multiple meanings (polysemy) or words that share the same spelling but possess different meanings (homonymy) in the context of multilingual applications. Some approaches directly utilise multilingual parallel corpora ( [62, 237, 87, 145, 317]), which implicitly provide some level of supervision for aligning words in the two languages. However, the pressure on the model to learn clear correspondences between the contextualised representations in the two languages is still implicit and somewhat weak [143]. This is due to the absence of explicit alignment supervision, requiring the model to deduce the connections between representations from the available training data. Consequently, several subsequent studies (e.g.: [307, 355, 42]) have introduced methods that utilise word alignments from parallel corpora as supervision signals to align multilingual contextualised representations in a post-hoc manner or directly during the training ( [143, 52, 360]). In most cases, these alignment procedures enhance multilingual language models and address many of their systematic deficiencies, particularly in the NLI task [84]. However, it is well established that language dependency persists, making tasks like QA and NER still challenging.

### 3.1.1 Contribution

In this work, we report all the most important cross-lingual alignment methods for contextualised representations. We can summarise the contributions of the paper in 4 points:

**(1)** We provide a comprehensive classification of alignment models. As far as we know, no other works in the literature provide a taxonomy.
**(2)** We present a categorisation of research works in cross-lingual alignment based on 6 features to gain a deeper understanding of the authors' rationale and to enhance their comprehensibility
**(3)** Starting from the analysed models, we identify the primary challenges related to the problem of cross-lingual alignment, and we discuss them.
**(4)** We create a repository containing codes and data for reproducing the methods, if provided, available to the whole community on Git[1].

## 3.2 Cross-lingual alignment Taxonomy

Cross-lingual alignment of contextual embeddings refers to the process of mapping embeddings from different languages into a shared space or direct cross-lingual adjustment of the multilingual model, such that semantically similar words are located close to each other in the shared space. This allows for transfer learning across languages, where models trained on one language can be applied to another language without additional training data. One of the main goals of cross-lingual alignment is to take an existing model that has been trained on a resource-rich language and align the contextual embeddings from a less-resourced language into the vector space of the resource-rich language. This alignment allows input in the less-resourced language to be mapped into the resource-rich language, making it possible to classify it using existing models in that language. This is achievable because words with equivalent meanings in both languages exhibit highly similar vectors after undergoing the cross-lingual alignment process [344]. We build a taxonomy that

[1] https://gitlab.com/crisp1/lost-in-alignment

classifies the most important methods into 6 macro-clusters as a result of a study of the SOTA, as reported in Fig. 3.1.

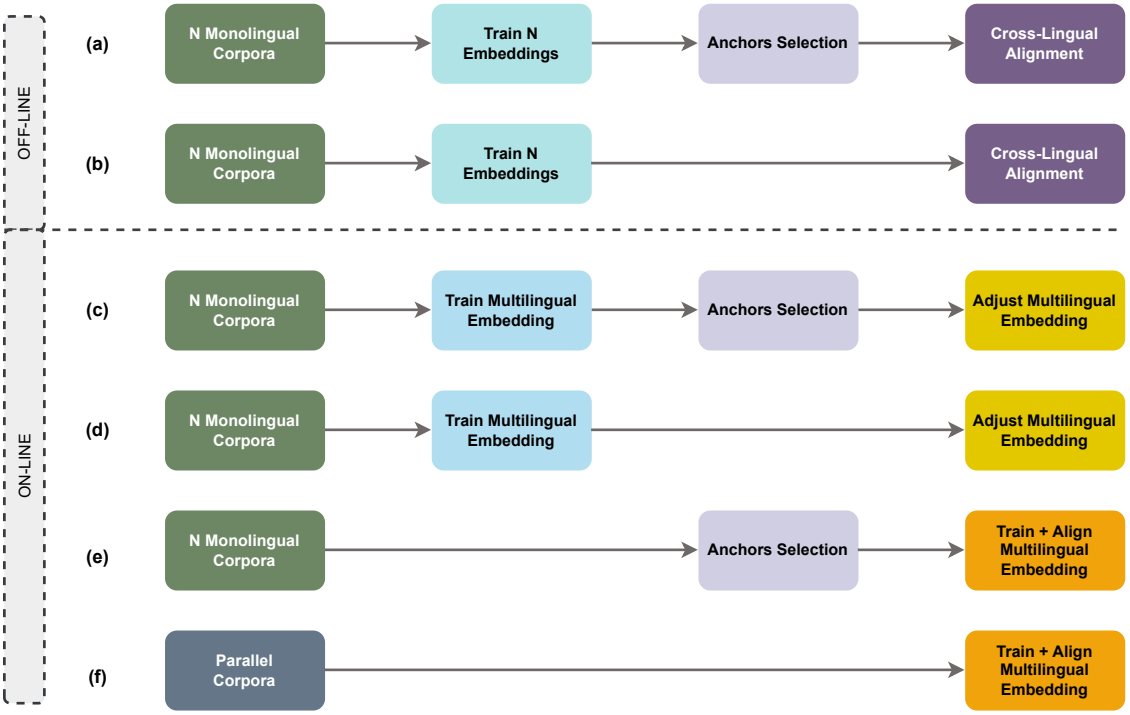



Figure 3.1: A taxonomy of the cross-lingual alignment methods

Here we report the description of each building block present in the Fig. 3.1:

**N Monolingual Corpora** It refers to the N monolingual corpora used as input for the embedding model.

**Train N Embeddings** It refers to training N contextualised language models that need to be aligned. NB: numerous authors do not train models but use already existing pre-trained models.

**Anchors Selection** It describes the procedure for selecting anchors, which are reference points to guide the mapping of one embedding into the other or directly create a common space for different corpora. This selection process may involve pairs of words or pairs of sentences. It can also involve a standalone model for pair extraction (e.g., a word alignment method) or an off-the-shelf list of pairs.

**Cross-Lingual Alignment** It refers to a linear or non-linear transformation that maps source embeddings to the target language or to a shared space.

**Train Multilingual Embeddings** It is related to the process of training a contextualised multilingual language model.

**Adjust Multilingual Embeddings** It refers to fine-tuning methods used to adjust an existing multilingual model, creating a more effective cross-lingual language model. The main differ-

ence with respect to the cross-lingual alignment box is that in this case, languages are already in the same embedding space, but there is a further modification of it to align them.

**Parallel Corpora** It refers to both the monolingual corpus and its respective corpus translated into another language. These are used as input for the multilingual embedding model.

**Train+Align Multilingual Embeddings** It is related to the process of training a multilingual language model that utilises objective functions during the training phase to represent cross-linguality.

The combination of those building blocks defines the following strategies:

**(a) - Offline Independent Embeddings w Alignment via anchor selection**: This category includes methods that attempt to align N different monolingual language models with the assistance of anchor sets extracted from monolingual or parallel corpora.

**(b) - Independent Embeddings with Alignment w/o direct anchor selection**: Research in this category bears a resemblance to that in class (a), but they do not rely on anchors for alignment.

**(c) - Online Multilingual Embedding Adjustment w Anchors**: In this category, we find works that adjust an existing multilingual language model by modifying the embedding remapping process, where the alignment occurs directly during the training phase through signals derived from bilingual dictionaries or parallel corpora.

**(d) - Direct Multilingual Embedding Adjustment**: Similar to the (b) case concerning (a), this category encompasses models that employ methods from the (c) category without the need for explicit anchor supervision.

**(e) - Online Anchored Multilingual Embedding Training and Alignment**: Papers in this category perform pre-training for cross-lingual language models from the ground up, selecting anchors to facilitate alignment during the training process.

**(f) - Direct Parallel Training and Alignment**: These methods initiate pre-training directly from parallel corpora without the prerequisite of pre-existing anchor sets as signals.

Methods (a), (b), (c), and (d) optionally utilise parallel corpora solely for alignment purposes, rather than for embedding training. They may employ parallel source data either for anchor selection or as input for fine-tuning multilingual embeddings. Methods like (d) and (f) omit the use of an anchor selection module, as they create a word-level alignment objective that encourages the model to independently identify word alignment patterns from the parallel corpus during an end-to-end training process. This approach helps mitigate the risk of accumulating potential errors at distinct stages of the pipeline. Some methods in category (f) also incorporate monolingual corpora as input for training the multilingual embedding model.

## 3.3 Main challenges related to cross-lingual language models

There are some key challenges embedded in cross-lingual language models that should be noted to better understand if and how the SOTA methods address them. Mikolov et al. initially observed that word vectors pre-trained on monolingual data exhibit comparable topological structures

across various languages [245]. This similarity enables the alignment of embedding spaces through a straightforward linear mapping [245]. Nevertheless, this assumption also poses a substantial limitation, as the diverse structural characteristics, such as morphology and syntax, present challenges for embeddings in adhering to this hypothesis. In the next subsections, we will address the challenges that have arisen over time when confronting the cross-alignment dilemma.

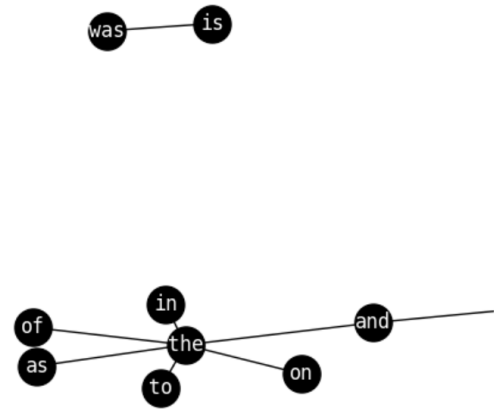
### 3.3.1 Isomorphism, isometry, and isotropy

Isomorphism, isometry, and isotropy are all related concepts in the context of cross-lingual embedding space mapping, but they refer to different aspects of the embedding spaces.

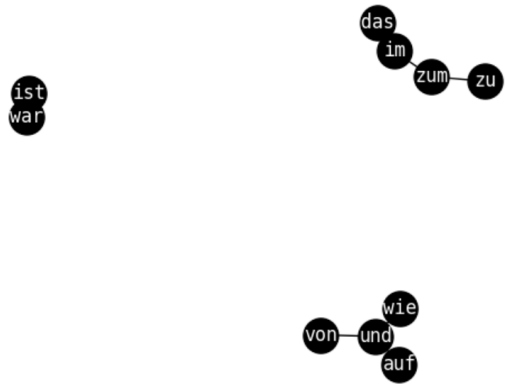
**Isomorphism** refers to the degree to which two embedding spaces have similar topological structures. In other words, it measures how well the two spaces preserve the relationships between words and concepts. Although contextual embeddings are designed to offer distinct representations of the same word in various contexts, Schuster et al. [307] discovered that the contextual embeddings of different senses of a single word exhibit much greater similarity compared to embeddings of different words. This phenomenon contributes to the anisomorphic distribution of embeddings in different languages and poses challenges for cross-lingual alignment. For instance, aligning the English word "bank" with its Italian translations, "banca" and "sponda", corresponding to its two different senses ("banca" as "financial institution" and "sponda" as "land at river's edge"), becomes difficult due to the contextual embeddings of the different senses of "bank" being close to each other, while those of "banca" and "sponda" are distant from each other [258]. We report a figure from [320], Fig. 3.2, where they affirm that considering the top k most frequent English nouns and their translations, the graphs are not isomorphic; see Fig. 3.2c-d. Even if we consider the top k most frequent English words and their translations into German, the nearest neighbour graphs are not isomorphic. Fig. 3.2a-b shows the nearest neighbour graphs of the top 10 most frequent English words on Wikipedia, and their German translations.

**Isometry**, on the other hand, refers to the degree to which the distances between points in the two spaces are preserved. It measures how well the two spaces preserve the relative distances between words and concepts. Two spaces are considered isometric when the relative Euclidean distances among vectors remain the same between these spaces. An orthogonal mapping is a transformation that preserves distances and guarantees isometric isomorphism, ensuring that the Euclidean distance between two vectors remains unchanged after the mapping process. Consequently, aligning two semantic language spaces becomes more straightforward when the relative distances among their vectors exhibit similarity [371]. To elucidate the concept, we present a figure (Fig. 3.3) sourced from the work of Lample et al. [177] that illustrates the process of aligning two embedding spaces while retaining the original distances between terms. The authors employed the Procrustes measure to achieve this alignment, which involves calculating an orthogonal matrix. This matrix preserves the dot product of vectors and their distances and functions as an isometry in the Euclidean space, akin to a rotation. More specifically, the authors align English $X$ and Italian word embeddings $Y$ through an adversarial learning process and subsequent refinement. First, a rotation matrix $W$ is learned to roughly align the distributions of each language's embeddings. Then, Procrustes refinement further optimises $W$ using frequent words as anchors. Finally, W is applied to all words in the dictionary, with a distance metric expanding high-density areas to improve the separation of common words.

**Isotropy** refers to how symmetrically vectors are distributed across an embedding space. High




(a) Top 10 most frequent English words




(b) German translations of top 10 most frequent English words


(c) Top 10 most frequent English nouns


(d) German translations of top 10 most frequent English nouns

Figure 3.2: Nearest neighbour graphs from [320]: Authors select the top 10 most frequent words in English, and the respective translation in German, and build nearest neighbour graphs for English and German using the monolingual embeddings used in Lample et al. [177], the graphs are of course very different. (a), (b) represent the graph of the 10 most frequent words in English, and the respective German translations; (c), (d) represent the graph of the 10 most frequent nouns in English, and the respective German translations of.

isotropy indicates that vectors are evenly spread in all directions, while low isotropy suggests clustering in certain directions. An embedding space is considered isotropic when vector directions are uniformly distributed. Unfortunately, contextual word representations often exhibit anisotropy [283]: normalised embeddings tend to cluster within a narrow, cone-like region on a hypersphere rather than spreading evenly. Different languages commonly display varying anisotropy levels [371]. Ethayarajh et al. [89] found that contextualised embeddings from models like BERT, ELMo, and GPT-2

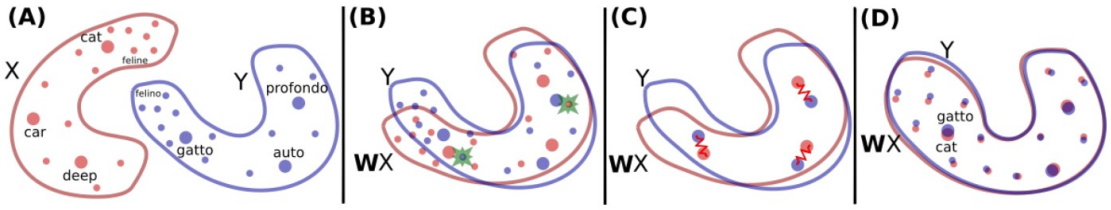



Figure 3.3: Figure from [177] where: (A) Within this context, there exist two sets of word embeddings: English words represented in red and denoted as $X$, and Italian words represented in blue and denoted as $Y$. The primary objective is to align or translate these embeddings. Each data point (depicted as a dot) in the space corresponds to a word, with the size of each dot proportional to the word's frequency in the training corpus of its respective language. (B) An adversarial learning process acquires a rotation matrix $W$, aiming to align the two distributions roughly. The green stars mark randomly selected words used for discrimination, determining whether the embeddings from the two sets originate from the same distribution. (C) The mapping $W$ undergoes further refinement through the Procrustes method. This involves the utilisation of frequently occurring words, already aligned in the previous step, as anchor points. The optimisation process minimises an energy function akin to a spring system between these anchor points. The refined mapping is then employed to map all words in the dictionary. (D) Ultimately, translation is achieved by employing the mapping $W$ and a distance metric that expands the space, particularly in regions with high point density (such as around the word "cat"). This expansion serves to separate "hubs" (e.g., the word "cat") from other word vectors to a greater extent compared to the original state (as seen in panel (A)).

exhibit this property, with vectors concentrating in specific regions due to their sensitivity to contextual nuances. This clustering effect seems intrinsic to contextualization itself. Recent work by Godey et al. [118] shows that anisotropy emerges independently of token frequencies or vocabulary size, challenging prior assumptions that only these factors drive it. Geometrically, anisotropy manifests as vectors confined within a cone, with the degree of anisotropy corresponding to the cone's narrowness, as Mimno et al. [248] noted. This effect is observed across most layers in models like BERT, ELMo, and GPT-2, where word representations form similar conical clusters, as visually depicted by Ethayarajh [89] and shown in Fig. 3.4. The Fig. 3.4 shows that in GPT-2, the average COS between randomly sampled words stays around 0.6 in layers 2 to 8 but then increases sharply in layers 8 through 12, with near-perfect similarity in the final layer. A similar trend occurs in BERT and ELMo, although BERT's second-to-last layer has higher anisotropy than its final layer.

Those concepts create issues in the mapping process, in particular due to the inconsistent relative distances between embeddings in the source and target embedding spaces. As a response, the paper proposed by Gao et al. [103]introduced regularisation techniques to improve isotropy by expanding the "aperture" of the embedding space, thus mitigating this degeneration. Zhao et al. [391] sought to reduce (or increase) the degree of anisometry (isometry) to alleviate its adverse effects. However, it is impractical to match the anisotropic characteristics of spaces precisely. Instead, they introduced an iterative normalisation preprocessing technique. This method is applied to transform anisotropic contextual embedding spaces, making them approximately isotropic by redistributing vectors evenly

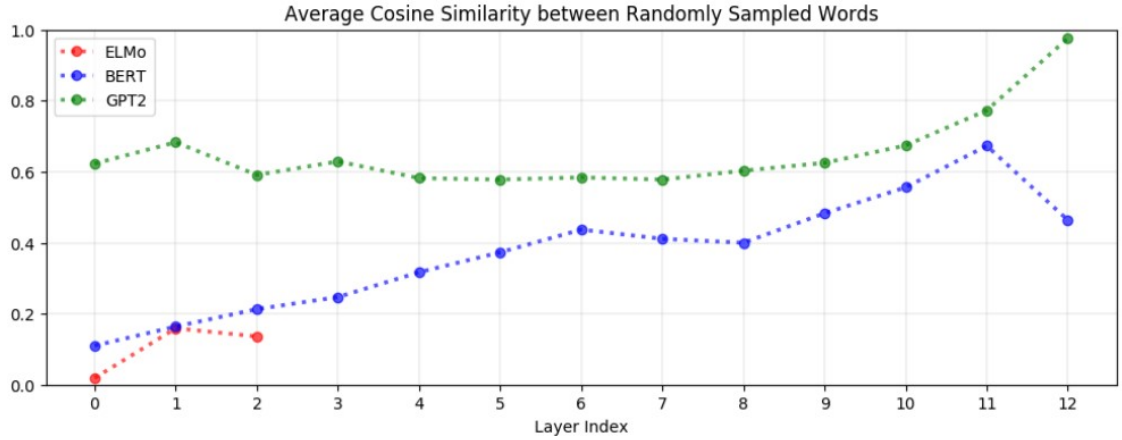



Figure 3.4: Nearest neighbour graphs from [89]: In the majority of layers within BERT, ELMo, and GPT-2, the word representations exhibit anisotropy, meaning they are not directionally uniform. Specifically, when you calculate the average cosine similarity between words that are randomly selected uniformly, you find that this similarity is not zero. One notable exception is ELMo's input layer. This exception is expected since it generates character-level embeddings independently of context. Typically, as you move to higher layers, the word representations become even more anisotropic than those in lower layers.

across the surface of a unit hypersphere. This significantly enhances the degree of isometry, making relative embedding distances across language spaces more similar. The iterative preprocessing method enforces vectors to become zero-mean and normalised as part of this transformation [371], showing the positive effect of isotropic space on the degree of isomorphism, which in turn results in improved performance in cross-lingual alignment algorithms [291]. Zhao et al. [391] continued this line of research and introduced enhancements to the model alignment approach, including 1) the application of z-normalisation to vectors, and 2) text normalisation to align the input more closely with the structural characteristics of English training data.

### 3.3.2 Language Neutrality

Language-neutrality is about to what extent similar phenomena are represented similarly across languages [200]. It refers to the ability of a language model to represent language in a way independent of any specific language. This means that the model can capture linguistic features and meanings common across different languages, allowing it to be used for multilingual tasks without requiring language-specific training data. Based on the findings of Pires et al. [281], it can be postulated that a sentence's representation in multilingual language models, without a specific alignment objective, consists of two components: one that is specific to the language of the sentence, responsible for identifying the language, and another that is language-neutral, capturing the sentence's meaning in a way that transcends language boundaries. Libovicky et al. [200] demonstrated that the representations in mBERT can be decomposed into a language-specific element and a language-agnostic element. They found that the language-neutral component is quite versatile in modelling semantic information, enabling accurate word alignment and sentence retrieval. However, it falls short re-

garding the more challenging task of assessing machine translation quality. The fact that mBERT has shown limited success, particularly for certain language pairs, in zero-shot scenarios and necessitates explicit bilingual projection for optimal performance, as observed by Pires et al. [281], Wu and Dredze [362], and Ronnqvist et al. [296], also underscores its limited language neutrality. By conducting a series of semantically oriented tasks that demand explicit cross-lingual semantic representations, Libovicky et al. [200] demonstrated that mBERT's contextual embeddings do not consistently represent similar semantic phenomena in a way that can be directly applied to zero-shot cross-lingual tasks.

## 3.4 Alignment methods for contextual embeddings

Cross-lingual approaches can be categorised into several groups.

1. The first group of methods employs monolingual embeddings, supplemented by bilingual dictionaries or parallel corpora, to align the embeddings in a linear or non-linear fashion.
2. The second group of approaches optionally uses bilingually aligned (comparable or even parallel) corpora or dictionaries to fine-tune a pre-trained multilingual model. Here, alignment is performed as a separate step after the initial pre-training phase.
3. The third group focuses on large pre-trained multilingual masked language models, such as mBERT [71], which are modified to incorporate cross-lingual alignment during training. These models are simultaneously trained on multiple languages with a specific objective function for alignment, allowing alignment to occur as part of the training process itself. Similar to the previous groups, bilingual corpora or dictionaries may be used for additional supervision.

### 3.4.1 Off-line linear/non-linear alignment

These methods belong to the categories (a) and (b) in the taxonomy of Fig. 3.1. Mikolov et al. [245] were the first that proposed to learn a linear transformation to project an embedding in the source language to its translation (since they used anchors) in the target language. Now we are going to present one of the most commonly used methods. Let's consider two embeddings as $e_s$ for the source ($s$) and $e_t$ for the target ($t$). Given a dictionary containing pairs of source and target elements represented as $x_s, y_t$ and matrix representations $X_s$ and $Y_t$ where the columns represent vector representations of the corresponding dictionary entries, the objective is to discover an orthogonal transformation matrix denoted as $W$. This matrix aims to minimise the discrepancies between the transformed vectors in $WX_s$ and $Y_t$. In a formal sense, this can be expressed as:

$$\arg\min_{\hat{\mathbf{W}}} ||\hat{W}X_s - Y_t|| \quad s.t. \quad \hat{W}^T\hat{W} = I \tag{3.1}$$

, exactly as the Eq. 2.2 used in the previous chapter. Here, the notation $\|.\|$ denotes the Frobenius norm. The orthogonality constraint ensures that the relative distances between pairs of vectors in the original source vector space remain unchanged following the transformation. These methods imply an off-line alignment. Indeed, for computing the matrix, it is necessary just to have the embeddings of the source and target language without further training the model. Rotating embeddings relies on the controversial assumption, explained above, that the embedding spaces

exhibit approximate isometry [320]. This assumption may not be valid for contextual pre-trained models since they encompass more than just a word's type, including context and syntax, which are less likely to exhibit isomorphism across languages.

Some authors try to align two contextual embedding models in a non-linear way. A common practice in performing this task is to use Generative Adversarial Networks (GAN)s [119]. GANs consist of two interconnected neural models: a generator and a discriminator. These two models are concurrently trained through an adversarial process. The discriminator's role is to distinguish whether the input data it receives is real or artificially generated (i.e., fake). Simultaneously, the generator aims to create synthetic data that can deceive the discriminator. GANs operate as a zero-sum game, where the discriminator's success implies the generator's failure and vice versa. Through simultaneous training, both networks enhance their performance. GANs find extensive applications in image generation, where this described process can yield impressive newly generated images. In their 2022 work, Ulčar et al. [344] introduce an innovative supervised nonlinear mapping approach that utilizes bidirectional GANs. In particular, they built a system where the generator module contains two generators that map vectors from one vector space to the other; one generator will map from $s$ to $t$, and the second will map from $t$ to $s$. The two generators are completely independent of one another, and they do not share data during training. The discriminator module contains two discriminators. The first discriminator tries to predict whether a given pair of vectors $\langle x_s, y_t \rangle$ represent the same token, and the second vector represents the translation of the word x in $t$ (i.e., $y_t$). The second discriminator attempts to learn the difference between the direction of mapping. For a given pair of vectors, it predicts whether they are a vector from $s$ and its mapping to $t$ or a vector from $t$ and its mapping to $s$. GANs could also be used to learn linear mapping.

### 3.4.2 Off-line fine-tuning

These methods belong to categories (c) and (d) in the taxonomy shown in Fig. 3.1. The key distinction concerning the previous category is related to the fact that alignment is carried out on an already pre-trained multilingual embedding model. In this case, there are no longer two distinct embedding models, but rather a single model with a multilingual representation. Some authors have suggested fine-tuning the entire encoder as an alternative to utilising source and target embeddings as static features. The concept involves directly modifying the multilingual model to bring the representations of semantically related words in various languages closer to each other, thereby encapsulating alignment in the loss function. This idea stemmed from the observation that embedding spaces for different languages may not always exhibit isometric properties [320], and as a result, they may not always be easily aligned through rotation. Considering the notation in Sec. 3.4.1, we can represent the multilingual embedding of the word $m$ as $x_m$, where $x_{s_m}$ is associated with source language terms and $x_{t_m}$ with target ones. Fine-tuning aims to minimise the distance between the contextual representations of matched word or sentence pairs in parallel corpora, as measured by the Frobenius norm, at each layer of the model:

$$L^i_{align} = \min \sum_m ||x^i_{s_m} - x^i_{t_m}|| \tag{3.2}$$

where $i$ is the multilingual embedding layer. Nevertheless, fine-tuning solely based on the objective above would result in the loss of semantic information that multilingual embedding acquired during pre-training. This is because a straightforward solution to Eq. 3.2 would be to make all embeddings

identical. To tackle this issue, researchers introduced a regularisation loss; for instance, Cao et al. [42] proposed one that restricts the source language embeddings from straying too far from their original locations within the pre-trained mBERT model. This regularisation loss functions in the following manner:

$$L^i_{regularise} = \min \sum_m ||x^i_{s_m} - c^i_{s_m}|| \tag{3.3}$$

where $c^i_{s_m}$ is a copy of the original pre-trained $x^i_{s_m}$, with its parameters remaining frozen. Both the alignment and regularisation losses are integrated and jointly optimised to align the two language subspaces while preserving the informativeness of the embeddings:

$$L_{finetune} = \min \sum_{i=n_s}^{n_e} L^i_{align} + L^i_{regularise} \tag{3.4}$$

Here $n_s$ to $n_e$ is the range of multilingual embedding layers aligned. These objectives could be used for word alignment and sentence alignment, as demonstrated in the case of Pan et al. [270]. It is important to mention the framework built by Wang et al. [355] that applies cross-lingual alignment methods mentioned in the previous Sec. 3.4.2 to a multilingual embedding model.

### 3.4.3 On-line alignment

These methods belong to the categories (e) and (f) in the taxonomy of Fig. 3.1. Online methods combine monolingual and cross-lingual objectives to acquire cross-lingual embeddings concurrently. These approaches require more data and are computationally less efficient alternatives to the fine-tuning approach, as they involve pretraining a large multilingual dictionary. Each word in the dictionary must have ducting alignment as an afterthought. This line of research suggests contextual pre-training procedures that exhibit greater cross-lingual awareness. Joint training methods in general have the following objective ($L_j$, where $j$ stand for joint):

$$L_j = L_1 + L_2 + R(L_1, L_2) \tag{3.5}$$

where $L_1$ and $L_2$ are monolingual objectives and $R(L_1, L_2)$ is a cross-lingual regularization term. An example is the method developed by Chi et al. [52], in which they introduce a new pre-training task called denoising word alignment (DWA). This task involves training a model to predict the correct alignment of noisy word pairs across parallel sentences in different languages. The DWA task serves as an explicit alignment objective during the pre-training of the cross-lingual language model, representing the $R(L_1, L_2)$ component mentioned above. The goal of DWA is to predict the word alignments from the perturbed version of the input translation pair (constructed by randomly replacing the tokens with masks); so, more specifically, the training objective is to minimise the cross-entropy between the alignment probabilities from the perturbed version and the self-labelled word alignments (semantic similarity of original tokens). Additionally, the authors employ a MLM objective, which is a standard pre-training task for language models and corresponds to the $L_1$ and $L_2$ components.

## 3.5 Sources of cross-lingual supervision

In this section, we will explain how previous studies use two distinct cross-lingual indicators, namely bilingual dictionaries and parallel corpora, to oversee the alignment process. Furthermore, we will assess the pros and cons associated with each option.

### 3.5.1 Bilingual Dictionary

A bilingual dictionary provides translations between two languages. In this survey's research, a bilingual dictionary is used as a form of cross-lingual supervision to align the embeddings of words in different languages. These dictionaries can be made using a translation tool, like the widespread Google Translate, or generated from parallel corpora. To effectively employ a bilingual dictionary for supervising the alignment of embeddings, each word in the dictionary must have a single representation, avoiding possible degradation in the mapping phase. However, the same word can have multiple representations depending on its context in the vector space of contextualised language models. According to Schuster et al. [307], the contextual embeddings of the same word form a cohesive cluster or "word cloud," and the centroid of this word cloud is distinct and separable for individual words, so it can be used as a candidate anchor. Nevertheless, there are limitations to this approach. When averaging over multiple contextual embeddings, some contextual information is inevitably lost for both the source and target language words. Some works like the one proposed by Zhang et al. [389], discovered that multi-sense words, such as "bank," which can refer to either a financial organisation or the edge of a river based on the context, also have clearly distinguishable clusters within their respective word clouds for each word sense. Bilingual dictionaries suffer from several weaknesses. For instance, Schuster et al. [307] conducted an experiment where they computed the average of ELMo embeddings for each word selected as anchor. Their research unveiled that the average cosine distance between contextual embeddings of polysemous words and their associated anchors was notably smaller than the average distance between those anchors. This implies that the embeddings representing different senses of a single word are relatively closer to each other compared to embeddings representing different words, as already mentioned in Sec. 3.3.1. Liu et al. [205] observed a similar pattern with BERT embeddings as well. This discovery suggests that the sense clusters of a multi-sense word's occurrences are not well separated in the embedding space.

### 3.5.2 Parallel corpora

A parallel corpus is a corpus that consists of texts in multiple languages, wherein the texts are translations of each other. In the field of research of this survey, parallel corpora are employed as a form of cross-lingual signal to align contextual embeddings or to extract a bilingual dictionary. This approach allows us to leverage not just individual words in the source and target languages, but also their contextual information. The cross-lingual alignment process can be guided effectively by utilizing the contextual information in the parallel corpus [5, 360]. Some authors apply word-alignment techniques to obtain the silver-aligned token pairs. As summarised by Xu et al. [371], using parallel corpora instead of a dictionary offers three advantages for mapping purposes:

1. Parallel corpora offer a broader and more comprehensive range of translation pairs compared to a dictionary;
2. Embeddings of translation token pairs preserve the same contextual information;

3. Tokens in each parallel sentence are already aligned, and their embeddings are aligned as well. Thus, mappings can be created by aligning the embeddings directly, eliminating the need for word alignment using a dictionary.

It would seem that parallel corpora are much better than bilingual dictionaries, but they entail some challenges: First, these kinds of sources are not easy to obtain for specific domains or languages. Second, word-alignment annotations (often needed in addition to aligned pairs of sentences) are not commonly available in parallel corpora, but they can be automatically generated using off-the-shelf tools, which we will discuss in the upcoming subsection.

**Word alignment methods** Word alignment is a critical issue in statistical machine translation. While pursuing advanced models that offer more nuanced interpretations of parallel corpora is a prominent research endeavour, it is equally important to have simple and efficient models that can scale effectively. Such models are crucial in various scenarios, including parallel data mining and rapid large-scale experimentation. They also serve as subcomponents in other models or training and inference algorithms [82]. The IBM models [34] are statistical models used for representing the translation process and extracting word alignments between pairs of sentences. Numerous word alignment models have been developed based on the IBM models, including those proposed by Och and Ney [260], Mermer and Sarac¸lar [243], Dyer et al. [82], and Östling and Tiedemann [265]. Recent studies have revealed that word alignments can also be extracted from neural machine translation models ([109, 172, 194]) or from pre-trained cross-lingual LM ([300, 257]). The predominant method utilised in the research reported in this paper is fast_align [82], which is based on a modified version of the lexical translation models initially proposed by Brown et al. [34]. In particular, fast_align adopts a log-linear reparameterization of IBM Model 2 [260]. The lexical translation process operates as follows: when presented with a source sentence, it generates an alignment indicating the correspondence between each target word and its respective source word (or null token) as a translation. Naturally, utilising such models to detect anchor pairs for alignment introduces some errors, which must be considered in addition to the errors inherent in the cross-lingual alignment method. Importantly, an off-the-shelf tool's word alignment error rate decreases as the number of parallel sentences increases. Consequently, parallel corpus supervision is particularly advantageous for language pairs with a lot of parallel data available.

## 3.6 Classification of the models

In this section, we report the Tab. 3.1 with all the research analysed in the survey and the hierarchical criteria used to compare them. Now we describe the set of hierarchical criteria reported in the Tab. 3.1:

1. **Reproducibility**: Reproducibility is related to the code availability and dataset necessary to replicate the experiments.
2. **Alignment Approach**: We categorize methods into two sub-classes similar to the ones introduced by Wang et al. [355]: (1) off-line alignment, where distinct independently trained monolingual representations are mapped into a shared space, and (2) on-line alignment, which involves the simultaneous learning of unified multilingual representations through both monolingual and cross-lingual objectives.

3. **Training signal**: It refers to the type of signal used for performing the cross-lingual alignment.
4. **Word-alignment method**: It is specified whether the author utilises an off-the-shelf method or performs the word alignment task during the cross-lingual alignment procedure.
5. **Level of alignment**: It specifies whether the alignment is conducted at the word level, with the objective being to align specific identical words considering the context or not, or at the sentence level, with the objective being to align entire pairs of sentences directly.
6. **Taxonomy**: Specifies to which categories, introduced in the Sec. 3.2, the model belongs.

We conducted a comprehensive literature review to identify the most commonly used cross-lingual alignment methods, and we framed the works in the categories just introduced, as reported in Tab. 3.1. As advised by [166], we comprehensively searched electronic databases[2]. We reviewed 39 papers on cross-lingual alignment of contextual embeddings. Papers were included if they met the following quality criteria, or if they were recognised as particularly impactful within the research community:

**(1)** for journal papers, to be either *Q1* or *Q2* of SCImago journal ranking in any computer science-related topics in the of publication.

**(2)** for conference papers, to be classified as *A/B* for all those rankings: (i) CORE Conference Rating, (ii) LiveSHINE, and (iii) Microsoft Academic.

[2]The databases utilised for this search were as follows:

ACL (https://aclanthology.org/)

Springer (www.springerlink.com)

ACM Digital Library (www.acm.org/dl)

ScienceDirect (www.sciencedirect.com)

Wiley Interscience (www.Interscience.wiley.com)

Google Scholar (www.scholar.google.co.in)

IEEE eXplore (www.ieeexplore.ieee.org)

Taylor Francis Online (www.tandfonline.com)

PubMed (https://pubmed.ncbi.nlm.nih.gov/)

SemEval (https://semeval.github.io/)

Table 3.1: Mapping selected papers to our roadmap. *(Code)* → Not provided: 🔑, Provided no documentation: git, Provided with documentation: git; *(Dataset)* → Not mentioned: 👁‍🗨, Private dataset: 🔒, Public dataset: 🔓; *(Rest of features)* → Not mentioned: ○, Applied: ●

| | Reproducibility | | Alignment | | Training Signal | | Training Signal | | Level of Alignment | | Taxonomy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Paper | Code | Dataset | Off-line | On-line | Bilingual Dictionary | Parallel Corpora | off-the-shelf methods | During Alignment | Word-level | Sentence-level | According to Fig. 3.1 |
| [5] Aldarmaki et al. 2019 | git | 🔓 | ● | ○ | ○ | ● | ● | ○ | ● | ● | (a) |
| [307] Schuster et al. 2019 | git | 🔓 | ● | ○ | ● | ○ | ○ | ● | ● | ○ | (a) |
| [307] Schuster et al. 2019 | git | 🔓 | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | (b) |
| [360] Wieting et al. 2019 | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [354] Wang et al. 2019 - CLBT | git | 🔓 | ● | ○ | ○ | ● | ● | ○ | ● | ○ | (a) |
| [16] Artetxe et al. 2019 - LASER | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [42] Cao et al. 2020 | 🔑 | 🔓 | ○ | ● | ○ | ● | ● | ○ | ● | ○ | (c) |
| [355] Wang et al. 2020 | git | 🔓 | ○ | ● | ○ | ● | ● | ○ | ● | ○ | (c) |
| [363] Wu et al. 2020 | git | 🔓 | ○ | ● | ○ | ● | ● | ○ | ● | ○ | (c) |
| [51] Chi et al. 2020 - XNLG | git | 🔓 | ○ | ● | ○ | ● | ○ | ○ | ○ | ● | (f) |
| [295] Reimers et al. 2020 | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f), (b) |
| [270] Pan et al. 2021 | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ● | ● | (d) |
| [143] Hu et al. 2021 - AMBER | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ● | ● | (f) |
| [7] Alqahtani et al. 2021 | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ● | ○ | (d) |
| [53] Chi et al. 2021 - infoXLM | git | 🔓 | ○ | ● | ○ | ● | ○ | ○ | ○ | ● | (f) |
| [391] Zhao et al. 2021 | git | 🔓 | ○ | ● | ○ | ● | ● | ○ | ● | ○ | (c) |
| [121] Goswami et al. 2021 - DuEAM | 🔑 | 🔓 | ○ | ● | ○ | ○ | ○ | ● | ● | ● | (e) |
| [124] Gritta et al. 2021 - XeroAlign | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [344] Ulčar et al. 2022 - Vecmap/MUSE | git | 🔓 | ● | ○ | ● | ● | ● | ○ | ● | ○ | (a) |
| [344] Ulčar et al. 2022 - ELMOGAN | git | 🔓 | ● | ○ | ● | ● | ● | ○ | ● | ○ | (a) |
| [205] Liu et al. 2022 - Bi-SaELMO | git | 🔓 | ○ | ● | ● | ○ | ● | ○ | ○ | ○ | (e) |
| [336] Tien et al. 2022 | git | 🔓 | ○ | ● | ○ | ● | ○ | ○ | ○ | ● | (d) |
| [93] Feng et al. 2022 - LaBSE | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ○ | ○ | ● | (f) |
| [73] Ding et al. 2022 - EAR | git | 🔓 | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | (d) |
| [137] Heffernan et al. 2022 - LASER3 | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (b), (f) |
| [131] Hämmerl et al. 2022 | git | 🔓 | ○ | ● | ● | ○ | ● | ○ | ● | ○ | (a), (d) |
| [84] Efimov et al. 2023 | git | 🔓 | ○ | ● | ○ | ● | ● | ○ | ● | ○ | (c) |
| [4] Abulkhanov et al. 2023 - LAPCA | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [329] Tan et al. 2023 - LASER3-CO | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (b), (f) |
| [191] Li et al. 2023 | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [185] Li et al. 2024 - AFP | git | 🔓 | ● | ○ | ○ | ● | ○ | ● | ○ | ● | (d) |
| [347] Vasilyev et al. 2024 | 🔑 | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (f) |
| [158] Jiang et al. 2024 - CLASS | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (c) |
| [20] Bakos et al. 2025 - AlignFreeze | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (d) |
| [94] Feng et al. 2025 - IDA | 🔑 | 🔓 | ○ | ● | ● | ○ | ● | ● | ● | ○ | (d) |
| [204] Liu et al. 2025 | git | 🔓 | ○ | ● | ○ | ● | ○ | ● | ○ | ● | (d) |
| [326] Sundar et al. 2025 | 🔑 | 🔓 | ○ | ● | ○ | ○ | ○ | ● | ○ | ● | (d) |
| [155] Jha et al. 2025 - vec2vec | git | 🔓 | ● | ○ | ○ | ○ | ● | ○ | ○ | ● | (b) |
| [70] Deng et al. 2025 | git | 🔓 | ○ | ● | ○ | ○ | ● | ● | ○ | ● | (d) |

### 3.6.1 Category (a) - Offline Independent Embeddings w Alignment via anchor selection


Figure 3.5: Category (a) - Offline Independent Embeddings w Alignment via anchor selection

Approaches classified under category (a) follow the process depicted in Fig. 3.5. Aldarmaki et al. [5] employ parallel text to align independently trained contextual embeddings across languages, employing both word-level and sentence-level mapping techniques. Similarly, Wang et al. [354] generate cross-lingual contextualised embeddings using publicly available pre-trained BERT models, utilising word pairs from parallel corpora for word-level alignment. Schuster et al. [307] propose aligning two language models at the word level, relying on bilingual dictionaries as a source of supervision; to do this, they used the averaged contextual word embeddings as an anchor for each word type. These approaches incorporate anchors to solve an optimisation problem aimed at discovering an appropriate linear transformation, as detailed in Sec. 3.4.1.

### 3.6.2 Category (b) - Independent Embeddings with Alignment w/o direct anchor selection


Figure 3.6: Category (b) - Independent Embeddings with Alignment w/o direct anchor selection

Approaches classified under category (b) follow the process depicted in Fig. 3.6. In this category, we highlight only two papers. The first, by Schuster et al.[307], proposes aligning two contextual embedding models using the adversarial MUSE framework introduced by Lample et al.[177], without relying on parallel text. In MUSE, anchor points are typically selected by comparing the similarity of semantic distributions among terms in the corpora to be aligned. However, in this case, no anchor points are required. Within category (d), some studies adopt the same approach, but focus on aligning multilingual embeddings rather than multiple monolingual ones. The second paper, by Jha et al.[155], presents an unsupervised post-hoc mapping method that aligns embedding spaces via a "universal" latent geometry—a universal semantic structure postulated by the Platonic Representation Hypothesis[150]. Two encoders map each source space into the universal space and back, using a GAN-based strategy, enabling cross-lingual alignment without anchors or parallel data by leveraging the assumed shared semantic geometry.

### 3.6.3 Category (c) - Online Multilingual Embedding Adjustment w Anchors

Approaches classified under category (c) follow the process depicted in Fig. 3.7. These approaches begin with models that are potentially not trained on parallel data, so they undergo a remapping process. In the cases of Cao et al. [42] and Zhao et al. [391], alignment is integrated into the loss function. Their strategy adopts more expressive alignment methods than rotation, mitigating the isomorphism and isometry challenge. They fine-tune their models using parallel data and leverage

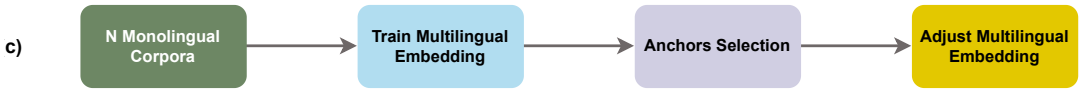



Figure 3.7: Category (c) - Online Multilingual Embedding Adjustment w Anchors

it to align the contextual embeddings across different languages, ensuring that words with similar meanings have corresponding embeddings in the multilingual space. Similar to Cao et al. [42], Efimov et al. [84] adapt the pre-trained multilingual model mBERT by using a limited parallel corpus to enhance its cross-lingual transfer capabilities. They work with one language pair at a time, whereas Cao adapted mBERT for five languages simultaneously. Wang et al. [355] adjust the parameters of an unsupervised joint training method, such as mBERT, which does not explicitly employ any dictionaries or alignment mechanisms. Consequently, the resulting embedding set may be coarse and misaligned in the shared vector space. To refine alignments across non-shared embedding sets, they apply an off-the-shelf alignment method as a final step. Wu et al. [363] perform adjustments using a novel contrastive alignment objective. Instead of optimising for the absolute distance between source and target terms, the contrastive loss offers more flexibility by encouraging source and target words to be closer to each other than any other hidden states. Finally, CLASS method [158] involves online pre-training of a multilingual model where cross-lingual links and anchor texts serve as explicit supervisory signals to guide the alignment of representations.

### 3.6.4 Category (d) - Direct Multilingual Embedding Adjustment

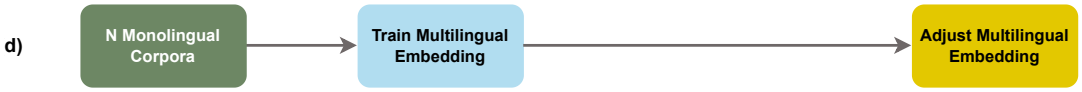



Figure 3.8: Category (d) - Direct Multilingual Embedding Adjustment

Approaches classified under category (d) follow the process depicted in Fig. 3.8. Pan et al. [270] introduce a straightforward approach to aligning multilingual contextual embeddings as a post-pretraining step, enhancing the zero-shot cross-lingual transferability of pre-trained models. They utilise parallel data to perform alignment at both the word level, using the Translation Language Modeling (TLM) objective, and at the sentence level through contrastive learning and random input shuffling. Alqahtani et al. [7], on the other hand, suggest using OT as an alignment objective during fine-tuning to enhance multilingual contextualised representations further. Notably, this approach does not necessitate an explicit predefined set of anchors. Tien et al., as described in their work [336], have created unsupervised models trained on unpaired sentences and supervised models trained on bitexts. Ding et al. [73] propose three strategies to enhance cross-lingual embeddings without relying on word-alignment pairs during fine-tuning. The first, Embedding-Push, moves English embeddings closer to other language clusters. The second, Attention-Pull, maintains relative word positions to preserve meaning. The final strategy, Robust Target, introduces a Virtual Multilingual Embedding to create a stable embedding space. The Align aFter Pre-training (AFP) [185] framework improves multilingual generative models by aligning sentence

representations across languages after the pre-training phase. It leverages translation pairs and contrastive learning to align internal representations, enhancing cross-lingual capabilities without requiring model retraining. This alignment is achieved by encouraging isomorphic representations, without relying on explicit geometric assumptions. These models are built upon the unsupervised language model XLM-RoBERTa (XLM-R) [63], with the model's parameters kept consistent. AlignFreeze [20] also builds on these models, applying a "realignment" strategy and finding that freezing the lower layers helps prevent performance degradation in cross-lingual transfer. Feng et al. [94] also focus on the alignment of different layers within LLM, demonstrating that word-level embeddings in the hidden layers are isomorphic across languages. They show that the hidden states corresponding to inputs in different languages can be aligned at the word level using an orthogonal transformation. Similarly, Liu et al. [204] propose a fine-tuning approach that alternates between a task loss and a contrastive alignment loss applied to middle layers of LLMs. Using parallel sentence pairs, the alignment loss encourages cross-lingual consistency in hidden states while preserving task performance. The loss proposed maximises the similarity between translations while minimising similarity between non-translations. Completely unlike the methods mentioned above, Sundar et al. [326] propose a model-intervention strategy to enhance cross-lingual alignment without parallel data by selectively modifying neuron activations in multilingual LLMs. By identifying and steering "expert" neurons, they reshape the embedding space to achieve tighter multilingual alignment. In [70] the authors introduce a rewiring strategy that adapts LLMS to cross-lingual tasks. The method leverages the autoregressive structure of embeddings, compressing and re-aligning them to better capture language-invariant patterns. By modifying internal connections rather than relying on parallel data or external dictionaries, it enables unsupervised cross-lingual transfer in a lightweight and efficient manner.

### 3.6.5 Category (e) - Online Anchored Multilingual Embedding Training and Alignment


Figure 3.9: Category (e) - Online Anchored Multilingual Embedding Training and Alignment

Approaches classified under category (e) follow the process depicted in Fig. 3.9. Liu et al. [205] developed an innovative approach to align contextual embeddings at the sense level using cross-lingual signals derived solely from bilingual dictionaries, thereby eliminating the need for parallel corpora. They achieve this by introducing a novel sense-aware cross-entropy loss that explicitly models different senses of a word based on its context. This is made possible through a clustering analysis that identifies distinct word senses, effectively addressing the anisomorphism's challenge. In their cross-lingual model pre-training, they incorporate a sense alignment objective combined with this sense-aware cross-entropy loss. Goswami et al. [121] use a multitask loss function to capture semantic similarity and relatedness between sentences, training a dual-encoder model to map different languages into a shared vector space. Their dual-encoder architecture leverages word-level semantic similarity scores, embedding these into unified sentence-level vectors.


Figure 3.10: Category (f) - Direct Parallel Training and Alignment

### 3.6.6 Category (f) - Direct Parallel Training and Alignment

Approaches classified under category (f) follow the process depicted in Fig. 3.10. Wieting et al. [360] constructed a model to acquire paraphrastic sentence embeddings directly from bilingual text data. Their training dataset comprises pairs of sentences in source and target languages. For each sentence pair, they randomly select a non-matching target sentence during training. The primary objective is to make source and target sentences more similar than source and negative target examples by a specific margin. To understand this category, we need first to describe Cross-Lingual Language Model (XLM) proposed by Conneau et al. [62], as it serves as the baseline for most of the papers in this field. Conneau et al. [62] explore cross-lingual language model pre-training using three approaches: Causal Language Modeling (CLM), MLM, or MLM combined with TLM. The TLM objective extends MLM by concatenating parallel sentences and randomly masking words in both the source and target sentences. To predict a masked word in a source sentence, the model can attend to nearby source words or the target translation, promoting alignment between source and target representations. The work is not included in our set of selected papers because it lacks an explicit objective for aligning contextual embeddings. In their work [143], Hu and colleagues put forward a training approach to acquire contextualised word representations. This approach fosters symmetry in training, encompassing both word and sentence levels. Like Conneau et al. [62], they employ MLM but also introduce a sentence alignment objective to encourage the model to predict the correct translation of a target sentence when provided with a source sentence. Additionally, they incorporate a word alignment objective by leveraging the Transformer model's attention mechanism. Artetxe et al. (LASER) [16] employ a BiLSTM encoder to transform input sentences into fixed-length vector representations, initialising the decoder LSTM responsible for generating target sentences. They train both the encoder and decoder using parallel corpora with a translation objective. In contrast to Conneau et al. [62], their approach is designed to scale to a larger number of languages. Chi et al. [51] propose pre-training both the encoder and decoder of a sequence-to-sequence model under both monolingual and cross-lingual conditions. They adopt the MLM approach used by Hu et al. [143] and Conneau et al. [62] but also incorporate a denoising auto-encoding (DAE) objective to reconstruct the original text from the corrupted text. They apply these two objectives to both monolingual and parallel input datasets. In their 2021 work (infoxlm), Chi et al. [53] introduce a novel pre-training task based on contrastive learning. They consider a bilingual sentence pair as two viewpoints of the same meaning and promote the similarity of their encoded representations compared to negative examples. By using both monolingual and parallel corpora, they collectively train these preliminary tasks to improve the cross-lingual transfer capabilities of pre-trained models. Feng et al. [93] introduced an innovative approach that enhances translation ranking performance through a unique blend of pre-training and dual-encoder fine-tuning. Specifically, they merged dual encoders, which were trained using a translation ranking loss to maximise the similarity of translation pairs within a common embedding space, with encoders that were initialised using large pre-trained language models. Abulkhanov et al. [4] introduce a novel ap-

proach to cross-lingual retrieval, utilising cross-lingual pre-training and fine-tuning for cross-lingual IR tasks with loosely aligned data automatically mined from Wikipedia. Gritta and Iacobacci [124] propose XeroAlign, a method for task-specific alignment of cross-lingual pre-trained transformers like XLM-R. XeroAlign incorporates an auxiliary training objective that leverages translated data to improve target language performance, bringing it closer to that of the source (labeled) language. Li et al. [191] propose a method based on Cross-Lingual Representation Similarity that aligns multilingual representations during training to enhance zero-shot generation. By using multiple source languages, the method regularises representation similarity, helping to prevent language-specific errors. Interestingly, the study finds that neutral representations can actually degrade performance on generation tasks—challenging the common assumption that such invariance universally benefits cross-lingual transfer across all downstream tasks. Finally, Vasilyev et al. [347] consider a simple linear cross-lingual mapping as a possible improvement of the multilingual embeddings.

### 3.6.7 Composition of categories

Heffernan et al. [137] and Reimers & Gurevych [295] take opposite approaches to the teacher-student framework for multilingual embeddings. Reimers et al. [295] use monolingual embeddings as a teacher model to align multilingual embeddings, aiming to improve cross-lingual knowledge transfer and reduce linguistic bias, thereby making the representation space more isotropic. So, this approach is a combination of (f) and (b). In contrast, Heffernan et al. use multilingual embeddings as the teacher to align monolingual embeddings, with a focus on scaling encoder training and bitext mining for low-resource languages that aren't well-covered by existing models. So, in this case, the combination is the opposite (b) and (f). Tan et al. [329] follow Heffernan et al.'s approach [137] but extend it by integrating contrastive learning into their distillation method, making it more effective for training encoders for low-resource languages. Additionally, H"ammerl et al. [131] are classified as a combination of two taxonomy categories. They blend the advantages of static and contextual models, exploring their mutual benefits. Specifically, they extract static embeddings for 40 languages from XLM-R, validate them with cross-lingual word retrieval, and align them using VecMap [12]. They further apply a novel continued pre-training approach to XLM-R, leveraging the high-quality alignment from static embeddings to better align XLM-R's representation space. Through this intuition, the method addresses the challenges posed by isometry and isomorphism assumptions in contextualised embeddings, where such issues are particularly problematic.

### 3.6.8 Timeline

Here's a breakdown of the main characteristics and evolutionary trends for each year in cross-lingual alignment from 2019 to 2025:

**Before 2019:** Looking at the technique for aligning embedding like the one proposed by Lample et al. [177], Søgaard et al. [320] question the assumption that monolingual embedding spaces exhibit approximate isometry, showing that this is not valid for many language pairs.

**2019:** Initial focus on establishing foundational cross-lingual alignment methods with tools like LASER for multilingual embeddings [16] and adversarial alignment approaches by Schuster et al. [307]. This year introduced basic linear and adversarial mapping techniques to bring embeddings from different languages into a shared space, setting the groundwork for more sophisticated transformations and pre-training models.

**2020:** Emphasis on expanding model robustness and task-specific alignment. Cao et al. [42] and Chi et al. [51] developed alignment methods using multilingual objectives, exploring ways to optimise models across both monolingual and cross-lingual tasks. Pre-training large multilingual models like XLM-R also became a focal point, marking a shift towards enhancing performance in multiple languages simultaneously.

**2021:** Advances in embedding alignment for complex tasks and zero-shot applications. New methods like AMBER [143] and XeroAlign [124] emphasised task-specific alignment goals such as zero-shot transfer learning and semantic similarity across languages. This year marked a push towards fine-grained semantic alignment and transfer capabilities, integrating multitask learning to capture semantic similarity.

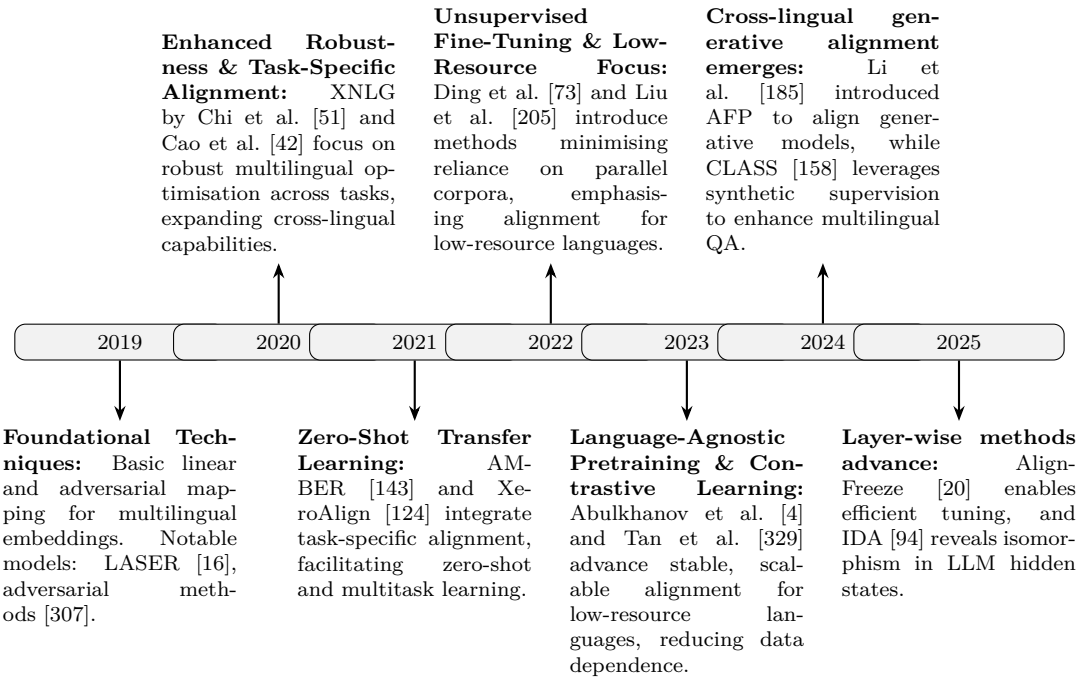
**2022:** Focus on fine-tuning and unsupervised approaches for enhanced transfer learning. Ding et al. [73] and Liu et al. [205] introduced methods targeting unsupervised alignment (e.g., bilingual dictionaries for sense-aware embeddings), which reduced reliance on parallel corpora. This shift reflected the industry's interest in scaling models for low-resource languages by minimising data requirements and optimising alignment through cross-lingual objectives.

**2023:** Growth in language-agnostic pre-training and contrastive learning integration. Abulkhanov et al. [4] and Tan et al. [329] integrated contrastive learning in LASER3 for low-resource languages, enhancing alignment stability and transferability in limited-data contexts. This year emphasised scalability and refinement of cross-lingual alignment with techniques like LAPCA, as models evolved to address language diversity with minimal supervised data.

**2024:** Emergence of cross-lingual generation alignment. Li et al. [185] introduce AFP to enhance multilingual generative models without retraining, while Jiang et al. [158] develop CLASS using cross-lingual links as explicit supervisory signals. Vasilyev et al. [347] demonstrate that simple linear mappings remain effective for sentence embeddings.

**2025:** Layer-specific alignment advancements include Bakos et al.[20], who introduce AlignFreeze with layer-freezing strategies to mitigate performance degradation, and Feng et al.[94], who reveal word-level isomorphism in LLM layers through iterative decomposition. New adaptation strategies also emerge, such as Deng et al.[70], which rewires autoregressive embeddings for unsupervised cross-lingual transfer, and Jha et al.[155] present Vec2Vec, an unsupervised post-hoc mapping method that leverages a universal latent geometry and GAN-based projections to align embedding spaces without anchors or parallel data.

Each year shows a trend toward increasing model sophistication, scalability, and data efficiency, particularly in handling low-resource languages and advancing zero-shot capabilities.




### 3.6.9 Comprehensive Discussion of Methodological Challenges

Although alignment methods have achieved significant success, they still have some notable drawbacks. As mentioned by Hämmerl et al. [130] it is possible to consider cross-lingual alignment as a complex optimisation problem in this light: to be completely cross-lingually aligned, the model would have to reconcile both large and small differences between many different language spaces. This may be intractable without also removing valuable contextual and language-specific information. Offline methods in categories (a) and (b), as well as online models in categories (c) and (d), rely on two separately trained embeddings. In contrast, recent research in online joint training highlights the advantages of word sharing during the training phase. This distinction may result in information loss in the final embedding and impact the effectiveness of fine-tuning aligned embeddings for downstream tasks. The absence of cross-lingual objectives during fine-tuning can lead to suboptimal results, unlike jointly trained models where shared words can facilitate this role. Offline alignment methods hinge on the assumption of isomorphism in monolingual embedding spaces. However, studies such as those by Søgaard et al. [320], have contested this assumption, revealing that it doesn't hold true for many language pairs. Notably, Ormazabal et al. [263] suggest that this limitation arises from the independent training of the two sets of monolingual embeddings. Conversely, the on-line joint training methods that belong to categories (e) and (f), are simpler and avoid the disadvantages above of off-line alignment methods. Nevertheless, it carries its own set of limitations. These methods assume that all shared words between two languages implicitly serve as anchors and need not be aligned with other words. However, this assumption does not always hold true, leading to misalignment. For instance, the English word "the" will likely appear in the Spanish training corpus, but ideally, it should align with Spanish words like "el" and "la" instead of aligning with itself. This issue is referred to as oversharing. The methods discussed in

the survey involve joint training with the incorporation of an explicit cross-lingual task, enabling the alignment of words that are not shared between languages. Methods that adapt an already multilingual model trained in this manner strike a perfect balance, as they consider words within their language domain without sharing during the pre-training phase but remap only those words present in the source used as a signal. In contrast, sources used during the adjustment phase do not have as comprehensive an impact as when they are used in the pre-training phase.

## 3.7 Evaluation tasks

In this section, we will describe the downstream task utilised by researchers to evaluate the proposed models. This information can assist researchers in identifying a benchmark task that is suitable for evaluating their models among the ones that are widely recognised in the community. Moreover, in Tab. 3.2 we report the evaluation tasks used by the surveyed papers. This can be useful for comparison. In our view, the greater the variety of tasks used to evaluate the authors' proposed method, the more robust the method appears to be.

**Sentence Translation Retrieval (STR):** STR entails retrieving the accurate translation from the target side of a test parallel corpus by employing the nearest neighbour search based on COS.

**BLI:** BLI task is meant to quantify vocabulary induction performance, that is, given a benchmark set of words in a source language, find the translation in the target language.

**Dependency parsing (DP):** DP is a task that involves analysing the grammatical structure of a sentence to determine the relationships between words. In this task, each word in a sentence is assigned a syntactic label describing its relationship to other words. The resulting structure is typically represented as a tree, where each word is a node and the relationships between words are represented as edges.

**POS Tagging:** POS tagging is the process of assigning a grammatical category (such as noun, verb, adjective, etc.) to each word in a sentence. Differently from dependency parsing, it does not analyse the relationships or dependencies between words in a sentence.

**Paraphrase Detection (PD):** PD identifies whether two sentences or phrases convey the same or similar meaning, even though they may be expressed in distinct linguistic forms.

**NLI:** NLI is the process of establishing the connection between two given sentences in a specific order and categorising them as either showing entailment, contradiction, or having "no relation" between them.

**Sentiment Analysis (SA):** In the field of SA, the goal is to automatically identify and categorise the emotional tone expressed in a piece of text. This task typically involves classifying a sentence, a short passage, or even a full document into one of several predefined categories—most commonly positive, negative, or neutral—based on the sentiment conveyed.

**NER:** NER involves information extraction to identify and categorise named entities within unstructured text into predefined categories. These categories may include person names, organisations, locations, medical codes, time expressions, quantities, monetary values, and more.

| Paper | Downstream Tasks | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NLP | | | Text Similarity | | | Translation | | | | Classification | | Reasoning | |
| | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | NLI | QA |
| [5] Aldarmaki et al. 2019 | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | QA |
| [307] Schuster et al. 2019 | NER | POS | **DP** | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | NLI | QA |
| [360] Wieting et al. 2019 | NER | POS | DP | **STS** | PD | SS | STR | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [354] Wang et al. 2019 - CLBT | NER | POS | **DP** | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | NLI | QA |
| [16] Artetxe et al. 2019 - LASER | NER | POS | DP | STS | PD | SS | STR | **BM** | PSM | RFEval | **DC** | SA | **NLI** | QA |
| [42] Cao et al. 2020 | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | QA |
| [355] Wang et al. 2020 | **NER** | POS | DP | STS | PD | SS | **STR** | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [363] Wu et al. 2020 | **NER** | **POS** | **DP** | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | QA |
| [51] Chi et al. 2020 - XNLG | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | NLI | **QA** |
| [295] Reimers et al. 2020 | NER | POS | DP | **STS** | PD | **SS** | STR | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [270] Pan et al. 2021 | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [143] Hu et al. 2021 - AMBER | NER | **POS** | DP | STS | **PD** | SS | **STR** | BM | PSM | RFEval | DC | SA | **NLI** | QA |
| [7] Alqahtani et al. 2021 | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [53] Chi et al. 2021 - infoXLM | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [391] Zhao et al. 2021 | NER | POS | DP | STS | PD | SS | STR | BM | PSM | **RFEval** | DC | SA | **NLI** | QA |
| [121] Goswami et al. 2021 - DuEAM | NER | POS | DP | **STS** | PD | SS | STR | **BM** | **PSM** | RFEval | DC | SA | NLI | QA |
| [124] Gritta et al. 2021 - XeroAlign | NER | POS | DP | STS | **PD** | SS | STR | BM | PSM | RFEval | **DC** | SA | NLI | QA |
| [344] Ulčar et al. 2022 - Vecmap/MUSE | **NER** | POS | **DP** | STS | PD | SS | STR | BM | PSM | RFEval | DC | **SA** | NLI | QA |
| [344] Ulčar et al. 2022 - ELMOGAN | **NER** | POS | **DP** | STS | PD | SS | STR | BM | PSM | RFEval | DC | **SA** | NLI | QA |
| [205] Liu et al. 2022 - Bi-SaELMO | **NER** | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | **SA** | **NLI** | QA |
| [336] Tien et al. 2022 | NER | POS | DP | STS | PD | SS | **STR** | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [93] Feng et al. 2022 - LaBSE | NER | POS | DP | **STS** | **PD** | SS | **STR** | BM | PSM | RFEval | DC | **SA** | NLI | QA |
| [73] Ding et al. 2022 - EAR | NER | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | **DC** | SA | NLI | QA |
| [137] Heffernan et al. 2022 - LASER3 | NER | POS | DP | STS | PD | **SS** | STR | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [131] Hämmerl et al. 2022 | **NER** | **POS** | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | **QA** |
| [84] Efimov et al. 2023 | **NER** | POS | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [4] Abulkhanov et al. 2023 - LAPCA | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | **QA** |
| [329] Tan et al. 2023 - LASER3-CO | NER | POS | DP | STS | PD | **SS** | STR | **BM** | PSM | RFEval | DC | SA | NLI | QA |
| [191] Li et al. 2023 | NER | **POS** | DP | STS | **PD** | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [185] Li et al. 2024 - AFP | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | **NLI** | QA |
| [347] Vasilyev et al. 2024 | NER | POS | DP | STS | PD | SS | STR | BM | **PSM** | RFEval | DC | SA | NLI | QA |
| [158] Jiang et al. 2024 - CLASS | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | **QA** |
| [20] Bakos et al. 2025 - AlignFreeze | **NER** | **POS** | DP | STS | PD | SS | STR | BM | PSM | RFEval | DC | SA | **NLI** | **QA** |
| [94] Feng et al. 2025 - IDA | NER | POS | DP | STS | PD | SS | STR | BM | **PSM** | RFEval | DC | SA | NLI | QA |
| [204] Liu et al. 2025 | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | QA |
| [326] Sundar et al. 2025 | NER | POS | DP | STS | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | QA |
| [155] Jha et al. 2025 - vec2vec | NER | POS | DP | **STS** | PD | SS | **STR** | BM | PSM | RFEval | DC | SA | NLI | QA |
| [70] Deng et al. 2025 | NER | POS | DP | **STS** | PD | SS | STR | BM | PSM | RFEval | DC | SA | NLI | MT |

Table 3.2: Downstream tasks used by authors for evaluating their models. Task categories: NLP (Named Entity Recognition (NER), Part-of-Speech Tagging (POS), Dependency Parsing (DP)), Text Similarity (STS, Paraphrase Detection (PD), Similarity Search (SS)), Translation (Sentence Translation Retrieval (STR), Bitext Mining (BM), Parallel Sentence Matching (PSM), Reference-free machine translation evaluation (RFEval), Classification (Document Classification (DC), Sentiment Analysis (SA)), and Reasoning (Natural Language Inference (NLI), Question Answering (QA)).

**Semantic Textual Similarity (STS):** STS involves comparing pairs of sentences and assigning a score to each pair that reflects the degree of similarity between the two sentences. Human judges typically assign the scores based on their subjective assessment of the sentence similarity. The evaluation metric for the STS task is usually Pearson's correlation coefficient between the predicted scores and the gold standard scores assigned by human judges.

**Similarity Search (SS):** SS consists of finding similar documents or text segments based on a query, often using vector representations of text. This task is crucial for applications like IR, recommendation systems, and content-based filtering.

**Bitext Mining (BM):** The BM task aims to identify parallel sentence pairs within two large monolingual corpora in different languages. These sentence pairs are commonly referred to as "gold" alignments when used as ground truth for evaluation.

**Bilingual Terminology Alignment (BTA):** The task of BTA involves aligning terms found in two distinct languages between two candidate term lists.

**QA:** QA task in the cross-lingual domain involves answering questions in different languages based on a given context.

**Machine Reading Comprehension (MRC):** MRC is a variation of the QA task where the system is tasked with responding to the form of a continuous sequence of tokens from a text paragraph, given a question and that paragraph.

**Document Classification (DC):** DC is a task where a model is trained to classify a given document into one of several predefined categories. So, in the cross-lingual domain, document classification refers to the task of classifying documents in different languages into the same set of categories.

**Reference-free machine translation evaluation (RFEval):** RFEval task assesses the quality of translation, specifically measuring the similarity between a sentence in the target language and the corresponding sentence in the source language.

**Parallel Sentence Matching (PSM):** PSM consists of identifying sentence pairs in different languages that are mutual translations or convey equivalent meanings.

Many authors used the Xtreme [144][3] benchmark for evaluating their methods on different tasks as the one listed above.

In Tab. 3.2 we report the downstream tasks used by authors classified in this review. The performance of the cross-lingual alignment methods described is presented in Tab. 3.3. We present the most comprehensive results, encompassing the widest range of languages and the highest level of generalisation. Specifically, we report average results across all languages used in the evaluation. Additionally, for each proposed model, only the version with the highest recorded performance is included; if a study proposed multiple versions of the same model, we have reported only the best-performing one.

[3] https://github.com/google-research/xtreme

## 3.8 Application of cross-lingual alignment

Finally, in this section, we report some examples of research that use the methods classified in the paper for solving complex tasks. Yi et al. [377] introduce a highly efficient method for webpage snippet extraction, termed DeepQSE, to identify a concise set of sentences that effectively summarise the webpage content within the context of a given input query. They employ XLM-R [53] for the initial configuration of their language model. Also, Cho et al. [54] employ infoXLM [53] and other models to construct a multimodal intelligent document processing framework. This framework combines a pre-trained deep learning model with traditional robotic process automation techniques commonly used in banking to automate business processes based on real-world financial document images. Dadu and colleagues [68] present a cross-lingual inductive method for detecting offensive language in tweets, leveraging the contextual word embeddings provided by XLM-R. [63]. Ils et al. [151] utilise models from Zhao et al. [391] and Cao et al. [42] to examine the shifts in European solidarity discourses occurring before and after the global declaration of the COVID-19 outbreak as a pandemic. According to Li et al. [188], developing intelligent solutions to help e-commerce sellers provide local products to a global consumer base is paramount. To address this need, they leverage the methodology introduced by Schuster et al. [307] and embark on a novel endeavour in cross-lingual IR. Specifically, they focus on cross-lingual set-to-description retrieval within cross-border e-commerce. This task entails aligning product attribute sets in the source language with compelling product descriptions in the target language.

## 3.9 Conclusions

This survey comprehensively overviews cross-lingual embedding models and their applications in various natural language processing tasks. We have classified these models based on their data requirements, objective functions, and alignment methods, and discussed their strengths and limitations. We have also reviewed the evaluation methodologies for cross-lingual embeddings and proposed future research avenues. Overall, our survey highlights the importance of cross-lingual alignment methods for contextualised representation in multilingual settings, where often the availability of labelled data is limited. We have shown that cross-lingual embeddings can improve the performance of downstream tasks such as machine translation, sentiment analysis, and named entity recognition. They can be trained on various data types, including parallel corpora, comparable corpora, and monolingual corpora. However, we have also identified some challenges and open issues, such as the lack of standardised evaluation metrics, the domain adaptation problem, and the need for more fine-grained alignment methods. The main contributions are the provision of a comprehensive taxonomy of cross-lingual embedding models, categorising research works in cross-lingual alignment, and identifying the primary challenges related to cross-lingual language models.

Table 3.3: Performance of Cross-lingual Alignment Methods

| Paper | Task | Dataset | Metric used | Results and Comments |
|---|---|---|---|---|
| [131] | POS | UD-POS | F1 Score | 72.1 |
| [143] | | | F1 Score | 70.5 |
| [363] | | Universal Dependencies v2.3 | F1 Score | 81.5 ± 0.6 [a] |

[a] Average and Standard Deviation of 10 runs

**Table 3.3 continued from previous page**

| Paper | Task | Dataset | Metric used | Results and Comments |
|---|---|---|---|---|
| [20] | DP | Universal Dependencies | Accuracy | 81.7 [b] |
| [354] | | UD Treebanks v2.2 | LAS | 63.54 [c] |
| [307] | | UD v2.0 Treebanks | LAS | 77.3 [d] |
| [344] | | Universal Dependencies | UAS / LAS | 57.2 [e] / 33.5 [e] |
| [344] | | | UAS / LAS | 75.0 [e] / 54.7 [e] |
| [344] | | | UAS / LAS | 73.4 [e] / 53.1 [e] |
| [363] | | Universal Dependencies v2.6 | LAS | 57.4 ± 0.5 [f] |
| [205] | NER | CoNLL-2002, CoNLL-2003 | F1 Score | 75.11 [g] |
| [363] | | Pan et al., 2017 | F1 Score | 66.2 ± 1.0 [h] |
| [131] | | PAN-X | F1 Score | 62.73 |
| [84] | | Wikiann | F1 Score | 69.1 [i] |
| [344] | | | F1 Score | 46.6 [j] |
| [344] | | | F1 Score | 52.4 [j] |
| [344] | | | F1 Score | 38.3 [j] |
| [20] | | | F1 Score | 84.8 [k] |
| [137] | SS | Tatoeba | F1 Score | 18.8 [ab] |
| [295] | | | Accuracy | 53.4 |
| [360] | STS | SemEval STS 2017 | Pearson Corr. | 79.62 [l] |
| [93] | | STS Benchmark | Pearson Corr. | 72.8 |
| [121] | | | Pearson Corr. | 74.5 [m] |
| [295] | | | Spearman Corr. | 83.7 [n] |
| [70] | | | Spearman Corr. | 83.81 [o] |
| [391] | | WMT'13 | Pearson Corr. | 48.1 [p] |
| [191] | | MTG | ROUGE-L | 27.7 [q] |
| [204] | | WMT'23 | BLEU/COMET | 17/80.7 [r] |
| [326] | | FLORES200 | Accuracy | 32 [s] |

[b] Average POS tagging accuracy over 34 languages using XLM-R Base with ALIGNFREEZE (front-freezing); see Tab. 3.2.
[c] Average on 18 languages
[d] Average on De Es Fr It Pt Sv
[e] Average on 20 language pairs across 9 languages: En Hr Et Fi Lv Lt Ru Sl Sv
[f] Average and Standard Deviation of 10 runs
[g] Average on 3 languages, es, nl, and de
[h] Average and Standard Deviation of 10 runs
[i] Adj+cont model average on 4 languages: Es Hi Ru Vi
[j] Average for 19 language pairs across 9 languages: En Hr Et Fi Lv Lt Ru Sl Sv
[k] F1 Score on WikiANN averaged over 34 languages using XLM-R Base with ALIGNFREEZE (front-freezing); see Tab. 3.2 of the paper.
[l] Average on 4 languages: Ar Es En Tr
[m] Average on 5 languages: En De Tr Es Fr
[n] Average on 8 languages: En Ar De Tr Es Fr It Nl
[o] Average on 10 STS datasets: STS12, STS13, STS14, STS15, STS16, STS17, STS22, STS-B, BIOSSES, SICK-R
[p] Average for 19 languages grouped in 4 groups of typologically similar ones
[q] avg on en, es, de, fr, zh
[r] avg on en, he, ja, uk using LLaMA 3
[s] improvements in top-1 retrieval accuracy. Median on 22 languages using Aya-8b model, for the intervention on Spanish

**Table 3.3 continued from previous page**

| Paper | Task | Dataset | Metric used | Results and Comments |
|---|---|---|---|---|
| [155] | | NQ / TweetTopic / MIMIC / Enron | Cosine Similarity (×100) | 86.0[t] |
| [124] | PD | PAWS-X | Accuracy | 93.6 [u] |
| [143] | | | Accuracy | 89.2 |
| [185] | | | Accuracy | 57.3 [ay] |
| [191] | | MTG | ROUGE-L | 29.5[v] |
| [93] | | SentEval MRPC | F1 Score | 74.4 |
| [4] | BM | BUCC | F1 Score | 83.5 [w] |
| [16] | | | F1 Score | 93.9 [x] |
| [121] | | | F1 Score | 81.7 [x] |
| [295] | | | F1 Score | 88.6 [y] |
| [336] | | | F1 Score | 92.8 [z] |
| [360] | | | F1 Score | 77.15 [aa] |
| [137] | | FLORES | xsim error rate | 0.6 [ab] |
| [355] | | MUSE | precision | 74.5 [ac] |
| [329] | | Paracrawl | BLEU | 9.28 |
| [93] | STR | BUCC, Tatoeba | Accuracy/F1 Score | 83.7 [ad] / 89.3 [ae] |
| [53] | | Tatoeba | Accuracy | 79.2 [af] |
| [131] | | | Accuracy | 68.1 |
| [143] | | | Accuracy | 87.9 |
| [336] | | | Accuracy | 80.4 |
| [355] | | WMT | BLEU | 22.59 [ag] |
| [5] | | WMT'13 | Accuracy | 84.0 [ah] |
| [4] | | XOR-Retrieve | Recall@2000 / Recall@5000 | 65.0 [w] / 70.5 [w] |
| [158] | | | | 71.6[ai] / 78.2[ai] |
| [155] | | NQ / TweetTopic / MIMIC / Enron | Accuracy | 82.0[aj] |

[t] Average cosine similarity across in-distribution (NQ) and out-of-distribution datasets (TweetTopic, MIMIC, Enron)

[u] Average on 7 languages: En De Es Fr Ja Ko Zh

[v] Average on 5 countries: En, Es, De, Fr, Zh

[w] Result for $LAPCA - LM + XPAQ_{large}$ average across 8 languages: Ar Bn Fi Ja Ko Ru Te En

[x] Average for 5 languages: En De Ru Fr Zh

[y] Average on 5 languages: De En Fr Ru Zh

[z] Average on 4 datasets : De Fr Ru Zh

[aa] Result related to BUCC Dataset. Average on 3 languages : En De Fr

[ab] Average on 12 languages:Amh Be Ga Hy Ka Kk Km Sw Ta Te Ur Uz

[ac] Average on 7 languages: En Es Fr De It Ru Zh

[ad] Result related to Tatoeba Dataset. Average of all languages supported by Tatoeba

[ae] Result related to BUCC Dataset. Average for 5 languages: En De Fr Ru Zh

[af] Average 14 languages covered by parallel data (both directions): Ar Bg Zh De El Fr Hi Ru Es Sw Th Tr Ur Vi.

[ag] Average on 3 language pairs: En Fr De

[ah] Result for ELMO (sent) average across 3 languages: En Es De

[ai] Results on XOR-Retrieve dev set. CLASS model with full supervised training.

[aj] Average retrieval accuracy across in-distribution (NQ) and out-of-distribution datasets (TweetTopic, MIMIC, Enron)

**Table 3.3 continued from previous page**

| Paper | Task | Dataset | Metric used | Results and Comments |
|---|---|---|---|---|
| [121] | PSM | Tatoeba | Accuracy | 77.7[ak] |
| [347] | | WikiNews | Cosine Improvement (fC) | 0.991[al] |
| [347] | | Tatoeba | Distance Reduction (dD) | 0.192[am] |
| [94] | | MUSE | Precision@1 | 74.5[an] |
| [16] | DC | MLDoc | Accuracy | 72.8 [ao] |
| [124] | | MultiATIS++ | Accuracy/F1 Score | 96.0 [ap] / 81.2 [ap] |
| [73] | | XNLI | Accuracy | 68.7 [aq] |
| [205] | SA | Amazon Reviews | Accuracy | 75.32 [ar] |
| [93] | | SentEval SST | Accuracy | 83.8 |
| [344] | | Twitter Sentiment | Accuracy | 11.2 [as] |
| [344] | | | Accuracy | 10.3 [5.10] |
| [344] | | | Accuracy | 11.2 [5.10] |
| [191] | NLI | WikiLingua | ROUGE-L | 22.7 [at] |
| [7] | | XNLI | F1 Score | 67.8 [au] |
| [16] | | | Accuracy | 69.9 [av] |
| [42] | | | Accuracy | 65.9 [aw] |
| [53] | | | Accuracy | 81.4 [ax] |
| [84] | | | Accuracy | 71.0 [i] |
| [143] | | | Accuracy | 71.6 |
| [185] | | | Accuracy | 48.0 [ay] |
| [270] | | | Accuracy | 66.8 [az] |
| [363] | | | F1 Score | 76.1 ± 0.4 [ba] |
| [391] | | | Accuracy | 77.6 [bb] |
| [20] | | | Accuracy | 73.6 [bc] |

[ak] Average on 10 languages: De, Hi, Zh, El, Af, Te, Tl, Ga, Ka, Am
[al] Highest fC on title–text pairs in Russian
[am] Maximum distance gain; see Tab. 3.1 of the paper
[an] Bilingual Lexicon Induction interpreted as PSM evaluation
[ao] Average on 8 languages: En De Es Fr It Ja Ru Zh
[ap] Average on 8 languages: De Es Fr Tr Hi Zh Pt Ja
[aq] Average for 15 languages: En Ar Bg De El Es Fr Hi Ru Sw Th Tr Ur Vi Zh
[ar] Average on Bi-SaELMo across German and Japanese in 6 domains
[as] Average on 38 language pairs across 9 languages: En Hr Et Fi Lv Lt Ru Sl Sv
[at] Average on 17 languages: Ar, Zh, Cs, Nl, En, Fr, Hi, Id, It, Ja, Ko, Pt, Ru, Es, Th, Tr, Vi
[au] Represents the average of both seen and unseen 15 languages: En Bg De El Es Fr Ar Hi Ru Sw Th Tr Ur Vi Zh
[av] Average on 15 languages: En, Fr, Es, De, El, Bg, Ru, Tr, Ar, Vi, Th, Zh, Hi, Sw, Ur
[aw] Average on 6 languages: En, Bg, De, El, Es, Fr
[ax] $INFOXLM_{LARGE}$ Average on 15 languages: En, Fr, Es, De, El, Bg, Ru, Tr, Ar, Vi, Th, Zh, Hi, Sw, Ur
[ay] Considering LLama_7B + AFP. Average result on 2 languages: En, Zh
[az] Average on 6 languages: En, Ar, De, Es, Hi, Zh. The biggest model, 2M parallel training sentences, is considered
[ba] Average and Standard Deviation of 10 runs
[bb] Average for 19 languages grouped in 4 groups of typologically similar ones
[bc] Average accuracy on 12 languages using XLM-R Base with ALIGNFREEZE (front-freezing); see Tab. 3.2 of the paper.

**Table 3.3 continued from previous page**

| Paper | Task | Dataset | Metric used | Results and Comments |
|---|---|---|---|---|
| [53] | QA | MLQA | F1 Score / EM | 73.6 [bd] / 55.2 [bd] |
| [270] | | | F1 Score | 78.2 [be] |
| [51] | | SQuAD 1.1 (English-English QG) | BLEU / Meteor / ROUGE | 22.4 [bf] / 24.3 [bf] / 49.2 [bf] |
| [4] | | XOR-Full | F1 Score / EM / BLEU | 47.8 [w] / 38.7 [w] / 35.5 [w] |
| [7] | | XQuAD | F1 Score / EM | 63.8 [bg] / 48.8 [bg] |
| [84] | | XQuAD, MLQA | F1 Score | 66.9 [bh] |
| [131] | | XQuAD | Accuracy | 70.9 [bi] |
| [158] | | XOR-Full | F1 Score / EM | 50.1 [bj] / 41.8 [bj] |
| [20] | | XQuAD | F1 Score / EM | 67.2 [bk] / 52.3 [bk] |

[bd] Average on 7 languages: En Es De Ar Hi Vi Zh

[be] Average on En Fr Es De Bg Ar Zh Hi. The biggest model, 2M parallel training sentences, is considered

[bf] Average on 2 languages: En Zh

[bg] Represents the average on both seen and unseen 11 languages: En De El Es Ar Hi Ru Th Tr Vi Zh

[bh] We report results for Adj+cont model using XQuAD dataset since it embeds all the languages as for the other experiments. The number reported is an average on 4 languages: Es Hi Ru Vi

[bi] We report results for the dataset XQuAD

[bj] Average on 6 languages: En Ar Es Hi Vi Zh

[bk] Evaluation on XQuAD over 11 languages: En, De, Es, El, Ar, Hi, Ru, Th, Tr, Vi, Zh

# Part II

# Alignment in Domain-Specific Large Language Models

This part shifts the focus from geometric to behavioural alignment, investigating how Large Language Models can be guided to generate domain-accurate and trustworthy outputs. The section begins with a review of the theoretical background of LLMs and the mechanisms of RAG, laying the groundwork for the novel architectures introduced later. Through the development of RE-FIN and FLEX, this part exemplifies how retrieval and self-alignment strategies can be applied to complex domains such as finance, reducing hallucinations and improving factual grounding. The analysis extends beyond performance metrics, emphasising how these models embody a principled approach to alignment between domain knowledge and generative reasoning, bridging the gap between large-scale general-purpose language models and specialised applications.

# Chapter 4

# Background

## 4.1 Large Language Models

In Sec. 1.4, we described how contextual embeddings, particularly those learned through architectures Transformer-based such as BERT, revolutionised word representation by conditioning each token's embedding on its surrounding context. Building upon this foundation, the next major development in NLP came with the rise of **LLMs**—massive Transformer networks trained on vast amounts of text data to perform a broad range of linguistic and reasoning tasks without explicit supervision.

LLMs extend the ideas of contextualised representation learning to the scale of entire documents and corpora, leveraging immense datasets, computational power, and carefully designed training objectives [163]. Language modeling is a long-standing research topic, dating back to the 1950s with Shannon's application of information theory to human language, where he measured how well simple n-gram language models predict or compress natural language text [310]. Since then, statistical language modeling became fundamental to many natural language understanding and generation tasks, ranging from speech recognition, machine translation, to IR [154, 227, 228]. Indeed, these models, such as GPT-3 [35], PALM [55], and LLaMA [339], are capable not only of generating fluent natural language but also of solving tasks requiring translation, reasoning, code synthesis, or summarisation, often through simple textual prompts; Fig. 4.1 provided by Minaee et al. [250] show an overview of LLM capabilities.

### 4.1.1 From Contextual Models to Large Language Models

While contextual embedding models such as BERT or ELMo learned token-level representations from local context windows, LLMs generalise this idea to **full-sequence modeling**, where the model learns a probability distribution over entire text sequences. The central idea is to model the probability of a sequence of tokens $(w_1, w_2, \ldots, w_T)$ as:

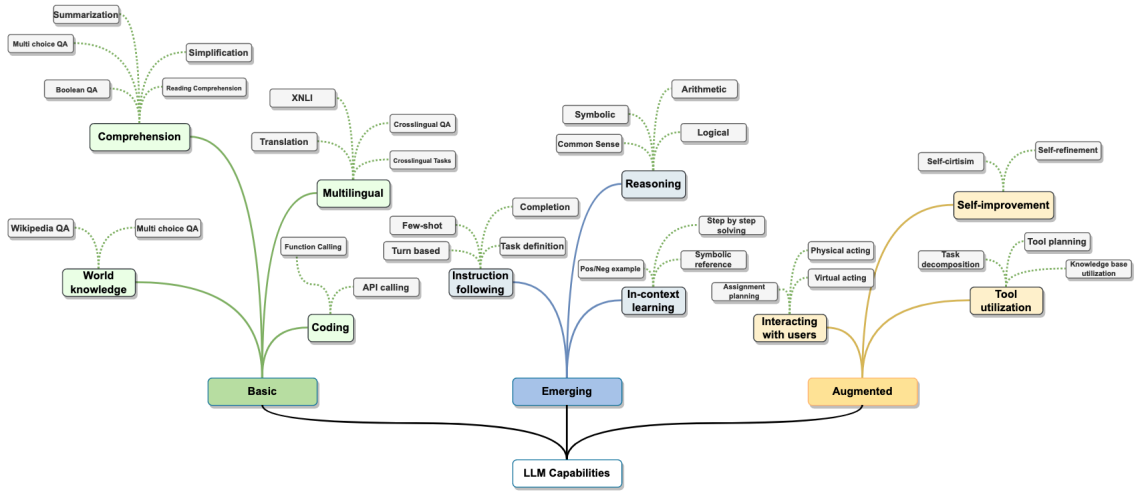



Figure 4.1: LLM Capabilities from Minaee et al. [250]

$$P(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} P(w_t \mid w_{1:t-1}) \tag{4.1}$$

This formulation is a direct extension of the classical language modeling objective introduced in Sec. 1.2, now implemented through deep neural architectures that can capture long-range dependencies across the entire context. Unlike static embeddings, which assign a single vector per word type, or contextual embeddings, which produce sentence-dependent vectors, LLMs learn an implicit function that maps a sequence prefix ($w_{1:t-1}$) to a probability distribution over the next token $w_t$.

### 4.1.2 Pretraining Objectives

Large Language Models are typically trained using one of two main objectives:

- **MLM**: used in bidirectional models such as BERT [71], where random tokens are masked and the model predicts them using both left and right context (as cited in Sec. 1.4).
- **CLM**: used in unidirectional models such as GPT [286], where each token is predicted only from previous tokens, consistent with Eq. 4.1.

For MLM, given a sequence $\mathbf{x} = (x_1, \ldots, x_T)$, we randomly select a subset of positions $M \subset \{1, \ldots, T\}$ to mask and train the model to reconstruct the original tokens:

$$L_{MLM} = -\sum_{i \in M} \log P_\theta(x_i \mid \mathbf{x}_{\setminus i}) \tag{4.2}$$

where $\mathbf{x}_{\setminus i}$ denotes the sequence with the $i$-th token replaced by a mask symbol.

For CLM, the objective is to maximise the likelihood of the observed sequence under the autoregressive factorisation:

$$L_{CLM} = -\sum_{t=1}^{T} \log P_\theta(w_t \mid w_{1:t-1}) \tag{4.3}$$

This objective trains the model to predict each next token from all previous ones, turning the Transformer decoder into a powerful generative model. Unlike RNNs (see Sec. 1.8), which process tokens sequentially, the Transformer architecture (introduced in Sec. 1.4) enables parallelised computation via self-attention, making large-scale pretraining feasible.

### 4.1.3 Scaling and Emergent Behaviour

One of the defining insights of modern LLMs is that performance improves predictably with model size, dataset size, and compute budget; a relationship described by the **scaling laws** discovered by Kaplan et al. [164]. Empirically, the loss $L$ follows a power-law relationship:

$$L(N, D, C) \propto N^{-\alpha_N} D^{-\alpha_D} C^{-\alpha_C} \tag{4.4}$$

where $N$ denotes the number of parameters, $D$ the dataset size, and $C$ the compute used during training. This observation motivated the training of extremely large Transformer models (with billions or trillions of parameters), under the premise that scaling up continues to yield predictable performance improvements.

As models scale, they exhibit **emergent behaviours** — qualitative abilities that were not present in smaller versions but appear beyond certain thresholds [163]. Examples include in-context learning, few-shot reasoning, and multilingual generalisation. These behaviours emerge from the model's ability to store and generalise over vast patterns of linguistic and factual knowledge acquired during pretraining [163].

### 4.1.4 Adaptation and Fine-Tuning

Although LLMs are pretrained on general corpora, they can be adapted to specific tasks or interaction styles via **fine-tuning**. Two major approaches dominate modern adaptation:

**1. Instruction and Supervised Fine-Tuning.** After pretraining, the model is fine-tuned on datasets of instruction–response pairs, enabling it to follow human-like directives expressed in natural language. Formally, given a dataset of $(\mathbf{x}, \mathbf{y})$ pairs where $\mathbf{x}$ is an instruction and $\mathbf{y}$ the desired output, the loss function mirrors standard sequence-to-sequence modeling:

$$L_{SFT} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{1:t-1}, \mathbf{x}) \tag{4.5}$$

**2. RLHF.** To further align model behaviour with human preferences, LLMs are often fine-tuned using reinforcement learning. A reward model $r_\phi(\mathbf{x}, \mathbf{y})$ is trained to approximate human preference judgments. The base language model is then optimised to maximise expected reward under its own policy:

$$\max_\theta \mathbb{E}_{\mathbf{y} \sim P_\theta(\cdot|\mathbf{x})}[r_\phi(\mathbf{x}, \mathbf{y})] \tag{4.6}$$

This optimisation is often implemented with policy-gradient methods such as Proximal **Policy Optimisation (PPO)**, resulting in models that produce outputs rated as more helpful, honest, and harmless by human evaluators [163].

### 4.1.5 Inference and Prompting

During inference, autoregressive LLMs generate text by sampling one token at a time from the conditional distribution in Eq. 4.1. At each step $t$, the next token $w_t$ is chosen according to:

$$w_t \sim P_\theta(\cdot \mid w_{1:t-1}) \tag{4.7}$$

Decoding strategies such as *greedy search*, *beam search*, or *nucleus sampling* control the balance between determinism and creativity. The ability of LLMs to perform new tasks without explicit retraining, via **prompting**, arises from the same mechanism: the model interprets the prompt as part of the conditioning context $w_{1:t-1}$.

**In-context learning** allows the model to infer task structure dynamically from examples included in the prompt, effectively performing few-shot or zero-shot learning through language alone. This paradigm shift redefines what it means to "train" a model: instead of updating parameters, we can specify a new behaviour through text input.

### 4.1.6 Capabilities and Limitations

Large Language Models demonstrate remarkable fluency, factual recall, and reasoning ability. However, they remain fundamentally probabilistic sequence models trained to predict text, not to reason or ground meaning in external reality. Limitations discussed by Jurafsky and Martin [163] include:

- **Hallucination**: the tendency to generate plausible but false statements.
- **Bias and fairness**: the propagation of societal and linguistic biases present in training data.
- **Lack of interpretability**: difficulty in tracing why a model produces specific outputs.

These challenges motivate ongoing research into RAG, tool use, and hybrid neuro-symbolic architectures that integrate LLMs with structured reasoning components.

## 4.2 Retrieval-Augmented Generation

In the previous section, we saw how LLMs use large-scale pretraining and self-attention mechanisms to learn powerful parametric representations of linguistic knowledge. However, LLMs remain limited by their static training data and by their lack of direct access to up-to-date or domain-specific information. **RAG** represents a new paradigm that addresses these limitations by integrating **external retrieval mechanisms** with generative neural architectures [183, 361, 163].

In a RAG system, generation is no longer a closed operation over the model's parameters, but rather an open process that dynamically consults external data sources. This results in outputs that are more factual, explainable, and adaptable, bridging the gap between traditional IR and neural text generation.

### 4.2.1 Motivation and Conceptual Framework

While LLMs learn implicit world knowledge through pretraining on massive text corpora, that knowledge is static and encoded in model parameters. Updating or correcting it requires costly retraining. Moreover, such models often **hallucinate**, producing fluent but false statements (see Chapter 6). RAG systems overcome these issues by explicitly incorporating retrieval into the generation process, allowing the model to reference and quote relevant evidence in real time.

The high-level process, as shown in Fig. 4.2, follows a two-stage pipeline:

1. A **retriever** identifies relevant information from an external corpus $\mathcal{D}$ based on the input query $q$.
2. A **generator** produces an output sequence $y$ conditioned jointly on $q$ and the retrieved passages $R_q$.

Formally, this can be represented as:

$$R_q = \text{Retriever}(q, \mathcal{D}) = \{r_1, r_2, \ldots, r_k\} \tag{4.8}$$

$$y \sim P_\theta(y \mid q, R_q) \tag{4.9}$$

Here, $R_q$ represents the $k$ most relevant documents, and $P_\theta$ is the conditional generation probability defined by the parameters $\theta$ of the underlying language model.

### 4.2.2 Retriever: Dense Representations and Similarity Search

The retriever component is responsible for mapping both queries and corpus chunks into a shared embedding space. Each document $d_i \in \mathcal{D}$ is divided into smaller units (chunks) $c_i$ and encoded into a dense vector representation using a transformer-based encoder $E(\cdot)$:

$$e_i = E(c_i), \quad e_q = E(q) \tag{4.10}$$

The retriever then scores the relevance of each chunk $c_i$ to the query $q$ via a similarity function, most commonly the dot product or COS:

$$s(q, c_i) = e_q \cdot e_i \tag{4.11}$$

and returns the top-$k$ chunks with the highest scores.

This approach, often referred to as a **bi-encoder** architecture [165], allows precomputing document embeddings offline and enables fast approximate nearest-neighbor (ANN) search. In contrast, a **cross-encoder** jointly encodes query and document pairs, yielding higher precision but much higher computational cost:

$$s(q, c_i) = f_\theta([q; c_i]) \tag{4.12}$$

where $[q; c_i]$ denotes token-level concatenation and $f_\theta$ a transformer scoring function.

In practical implementations, the retriever is often fine-tuned using contrastive learning, minimising a loss function that encourages true query-document pairs to have higher similarity than randomly paired negatives:

$$L_{\text{ret}} = -\log \frac{\exp(s(q, c^+)/\tau)}{\sum_{c^-} \exp(s(q, c^-)/\tau)} \tag{4.13}$$

where $\tau$ is a temperature hyperparameter controlling softmax sharpness.

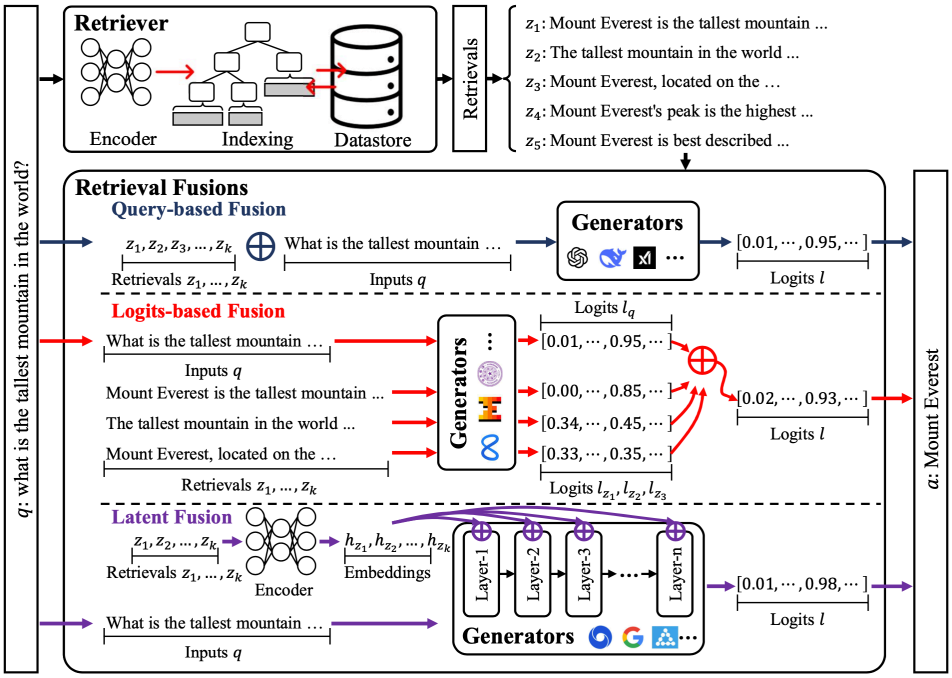



Figure 4.2: The overview of retrieval-augmented generation for natural language processing taken from Wu et al. [361]. The inputs as queries are fed into both the retriever for retrieval knowledge and the generator for outputs. There are three kinds of retrieval fusions, including query-based fusion, logits-based fusion, and latent fusion.

### 4.2.3 Generator: Retrieval-Conditioned Generation

The generator in a RAG model is typically an autoregressive Transformer decoder, similar to GPT-style architectures, but conditioned on external context from $R_q$. Given retrieved passages $R_q = \{r_1, \ldots, r_k\}$, the model defines the conditional probability of the output sequence $y = (y_1, \ldots, y_T)$ as:

$$P(y \mid q, R_q) = \prod_{t=1}^{T} P(y_t \mid y_{1:t-1}, q, R_q) \tag{4.14}$$

This formulation extends the autoregressive likelihood of standard LLMs (Eq. 4.1) by adding retrieval conditioning. Each token prediction now depends not only on prior tokens and the input query but also on retrieved evidence. The retrieved text may be concatenated directly into the prompt or integrated through attention-based fusion mechanisms described below.

### 4.2.4 Fusion Mechanisms

A key challenge in RAG systems is how to integrate retrieved evidence into the generation process. Following the taxonomy of Wu et al. [361], we can formalise three primary **fusion mechanisms**: query-based, logits-based, and latent fusion (see Fig. 4.2).

#### Query-based Fusion

In this simplest approach, the retrieved passages are appended to the input prompt:

$$x = [q; r_1; r_2; \ldots; r_k] \tag{4.15}$$

The model then computes standard attention over this concatenated sequence, generating output via the conditional probability in Eq. 4.14. This approach requires no architectural modification but suffers from limited context window size and potential noise from irrelevant retrievals.

#### Logits-based Fusion

Logits-based fusion combines the generation probability of the base model with retrieval-derived probabilities. Let $P_{\mathrm{LM}}(y_t \mid y_{1:t-1}, q)$ be the language model's intrinsic next-token distribution and $P_{\mathrm{R}}(y_t \mid R_q)$ the retrieval-informed distribution (for instance, derived from retrieved document frequencies). Then the final predictive distribution is a weighted interpolation:

$$P_{\mathrm{fused}}(y_t) = \lambda\, P_{\mathrm{LM}}(y_t \mid y_{1:t-1}, q) + (1-\lambda)\, P_{\mathrm{R}}(y_t \mid R_q) \tag{4.16}$$

where $\lambda \in [0, 1]$ controls the trade-off between internal model knowledge and external evidence.

In logit space, this can equivalently be written as:

$$z_{\mathrm{fused}} = \lambda z_{\mathrm{LM}} + (1-\lambda) z_{\mathrm{R}} \tag{4.17}$$

where $z_{\mathrm{LM}}$ and $z_{\mathrm{R}}$ denote the pre-softmax logits of the two components.

This mechanism resembles the probabilistic mixture-of-experts model, allowing smooth interpolation between the generative model and retrieval-based prior knowledge.

#### Latent Fusion

Latent fusion integrates retrieved evidence directly into the model's hidden states during generation, rather than at the input or output level. For a given token position $t$, the transformer's hidden representation $h_t$ is updated as:

$$h_t^R = h_t^A + \frac{1}{Z} \sum_{i=1}^{k} \alpha_i V(r_i) \tag{4.18}$$

where $h_t^A$ is the attention output from the previous layer, $V(r_i)$ is the vector representation of the retrieved passage $r_i$, $\alpha_i$ are learned attention weights, and $Z = \sum_i \alpha_i$ is a normalisation factor. The weights $\alpha_i$ are obtained from a relevance attention mechanism:

$$\alpha_i = \frac{\exp\big(h_t^A \cdot V(r_i)\big)}{\sum_j \exp\big(h_t^A \cdot V(r_j)\big)} \tag{4.19}$$

This allows the model to attend dynamically to different retrieved passages as generation proceeds, conditioning each step of output on the most relevant information. Empirically, latent fusion improves factual accuracy and reduces hallucination, at the cost of increased complexity.

### 4.2.5 Joint Training of Retriever and Generator

In early implementations, the retriever and generator were trained separately: the retriever was optimised using supervised relevance signals, while the generator was trained to predict ground-truth answers given retrieved passages. However, modern RAG models perform **joint optimisation**, aligning retrieval relevance with generation performance.

The joint training objective combines retrieval and generation likelihoods:

$$L_{\text{joint}} = -\sum_{(q,y)} \log \sum_{R_q} P_\phi(R_q \mid q)\, P_\theta(y \mid q, R_q) \tag{4.20}$$

where $\phi$ and $\theta$ denote retriever and generator parameters, respectively. Since enumerating all possible $R_q$ is infeasible, training samples are drawn via Monte Carlo approximation, or top-$k$ retrieved candidates are used. Gradients propagate through both components, improving the alignment between retrieved content and generative relevance.

To prevent instability, Wu et al. [361] suggest alternating optimisation: fixing the retriever while training the generator for several steps, and vice versa. This iterative refinement stabilises training and improves retrieval precision.

### 4.2.6 Updating and Maintaining Knowledge

An important advantage of RAG architectures is that knowledge updating does not require retraining the entire model. New documents can be added to the datastore $\mathcal{D}$, encoded, and indexed without modifying LLM parameters. Formally, let $\mathcal{D}_t$ denote the datastore at time $t$. After adding new documents $\Delta\mathcal{D}$, the model retrieves from:

$$\mathcal{D}_{t+1} = \mathcal{D}_t \cup \Delta\mathcal{D} \tag{4.21}$$

and generation continues using the same conditional probability in Eq. 4.9. This decoupling between model parameters and knowledge base enables continuous learning and real-time adaptability, key for domains such as scientific updates or legal information systems.

### 4.2.7 Evaluation and Applications

RAG systems are evaluated along two main dimensions:

- **Retrieval quality**, measured using metrics such as Recall@k, Precision@k, and Mean Reciprocal Rank, assessing how effectively the retriever identifies relevant evidence.
- **Generation quality**, evaluated using BLEU, ROUGE, or factuality metrics such as FEQA and Faithfulness scores.

Applications span a broad range:

- **Open-domain QA**: dynamically retrieving evidence from Wikipedia or scientific databases.

- **Conversational Agents**: grounding dialogue responses in verified knowledge bases.
- **Enterprise Search and Domain Adaptation**: integrating company-specific or legal documents.
- **Knowledge Updating**: adding new documents without retraining.

# Chapter 5

# Novel RAG systems for domain-specific knowledge alignment

In this section we will detail two innovative methods of RAG systems for aligning LLM output with respect to financial knowledge. The first one RE-FIN exploits a well-established knowledge source for retrieving qualitative information to add to an LLM answer. The second method (FLEX) exploits the vast LLM knowledge as an auxiliary source for augmenting a response that, otherwise, maintains a grade of randomness.

## 5.1 RE-FIN: Retrieval-based Enrichment for Financial data

Enriching sentences with knowledge from qualitative sources benefits various NLP tasks and enhances the use of labelled data in model training. This is crucial for FSA, where texts are often brief and contain implied information. We introduce RE-FIN (Retrieval-based Enrichment for FINancial data), an automated system designed to retrieve information from a knowledge base to enrich financial sentences, making them more knowledge-dense and explicit. RE-FIN generates propositions from the knowledge base and employs RAG to augment the original text with relevant information. A LLM rewrites the original sentence, incorporating this data. Since the LLM does not create new content, the risk of hallucinations is significantly reduced. The LLM generates multiple new sentences using different relevant information from the knowledge base; we developed an algorithm to select one that best preserves the meaning of the original sentence while avoiding excessive syntactic similarity. Results show that enhanced sentences present lower perplexity than the original ones and improve performance on FSA.

### 5.1.1 Introduction

FSA aims to determine the sentiment conveyed in financial texts regarding a specific stock or the overall market outlook. To address the challenge posed by the market's active shifts, automated FSA has gained increasing attention in the past years [345]. It has proven to be a powerful tool to support business decision-making and perform financial forecasting [212, 213]. Nevertheless, FSA presents unique challenges with respect to general SA. The language in finance is highly specialised, filled with acronyms, technical jargon, industry-specific terms, and sarcasm, making it tricky for models to understand [40, 219]. Moreover, there's a shortage of large, labelled datasets, and annotating financial text requires expertise that's not easily scalable. Therefore, classification models often perform much worse in FSA than they do with more general SA [368]. Even embedding alignment, which has proven effective in adapting models to specialised domains [67, 221] in certain fields, in FSA remains inconsistent [208].

Recently, LLMs have emerged as a potential tool to address the challenges mentioned above, offering powerful capabilities that can be applied to the financial domain. With their recent widespread adoption, models like ChatGPT and GPT-4 have demonstrated impressive performance in various NLP tasks[22, 262, 170], including FSA [193]. However, directly applying LLMs for FSA poses two notable challenges. Firstly, the discrepancy between the objective function used in LLMs' pre-training and the goal of predicting financial sentiment may result in LLMs' inability to consistently output labels for financial sentiment analysis as expected [266, 334]. Secondly, the typical subjects of financial sentiment analysis, such as news flashes and tweets, are characteristically concise and often lack adequate background information [381, 345].

To address the challenges above, we present a retrieval-augmented LLM framework for FSA. This paper proposes a new method to retrieve information from credible and customizable unstructured knowledge to enrich sentences. This approach makes the data more rich and understandable, which can increase user engagement—an essential factor in ML applications [43] — and improves FSA. We can summarise our contributions in the following points:

**(1)** We present RE-FIN, a methodology for RAG that extracts propositions from a knowledge base and integrates them with original texts using LLMs through an innovative post-retrieval approach.
**(2)** Through evaluation on SOTA benchmarks and an ablation study, we demonstrate that RE-FIN outperforms existing approaches.
**(3)** We provide the code freely to the community, promoting accessibility and further research[a] .

### 5.1.2 Related Works

#### FSA Models

FSA evaluates market sentiment by analysing news and social media data, which can predict investment behaviours and equity market trends [251]. Understanding the effectiveness of these models in finance significantly impacts downstream financial analysis tasks [193]. Like other finance areas, such as named entity recognition and QA systems, LLMs are increasingly adopted in FSA [193], enhancing the extraction of insights from unstructured data and improving decision-making. Early approaches [10, 69, 321, 374] utilised fine-tuned models achieving high performance but suffered from limited generalisation due to reliance on specific training datasets [367]. This highlights the

[a] https://github.com/filippopallucchini/RE-FIN

need for more flexible models in FSA. Recent studies indicate that LLMs can outperform fine-tuned models in certain tasks. While these models exhibit strong generalization abilities as problem solvers [193], applying them to FSA presents challenges [381]. Financial domain LLMs, such as BloombergGPT [364] and FinGPT [373], struggle to generate accurate sentiment labels due to a mismatch between their training objectives, typically CLM, and those of financial sentiment analysis [381]. Furthermore, financial sentiment analysis often addresses brief subjects like news flashes and tweets, which lack sufficient context, complicating reliable sentiment assessment. Implicit sentiment, where factual information suggests positive or negative sentiment, further complicates the issue [345].

**RAG Models**

LLMs demonstrate remarkable capabilities but face challenges such as hallucination, outdated knowledge, and opaque reasoning processes. RAG provides a promising solution by incorporating knowledge from external databases, enhancing generation accuracy and credibility, particularly for knowledge-intensive tasks, while enabling continuous updates and integration of domain-specific information [107]. RAG [38, 183] merges the strengths of context retrieval and LLMs for language generation [381]. This method leverages two distinct knowledge sources: the parametric memory within LLMs and the nonparametric memory from retrieved documents, effectively guiding generation to yield more accurate, contextually relevant responses. RAG has seen extensive application in open-world QA [235] and code summarization [209, 273]. The success of RAG heavily depends on the quality of the retrieval process, which employs sentence embeddings [302]. While sentence embeddings capture overall text meaning as fixed-length representations [254], querying them for semantic information at a granular level is challenging [299, 284, 352]. This limitation restricts expressivity in tasks like document retrieval, particularly when identifying concepts expressed in specific document segments rather than the entire document. Previous studies have shown success with phrase retrieval or late-interaction models that provide more granular representations of the retrieval corpus [309, 169, 179, 180]. Coarse-grained retrieval units may deliver relevant information, yet they risk introducing redundant content that could distract retrievers and LLMs in downstream tasks [378, 313], especially in sentiment classification, where excessive information might confuse rather than clarify. Therefore, we adopt a fine-grained retrieval logic utilising document propositions, computed using the model developed by Chen et al. [46]. Propositions represent atomic expressions in the text, encapsulating unique factual segments in concise, self-contained natural language [107]. Additionally, the generation process itself can pose challenges. For example, concepts in external documents may be similar but not identical to those in the question, which could mislead the LLM [45]. While most approaches focus on controlling the retrieval process, evaluating the generation is equally important [50, 88]. To address this, instead of generating a single augmented sentence, we produce multiple options and select the most suitable one using a post-retrieval method developed in this work.

### 5.1.3 Methods

Here, we describe the framework of the model proposed in this paper and sketched in Fig. 5.1. A traditional RAG process includes three main phases indexing, retrieval, and generation; moreover, an advanced RAG method, like the one proposed in the paper, also employs pre-retrieval and post-retrieval strategies [107].

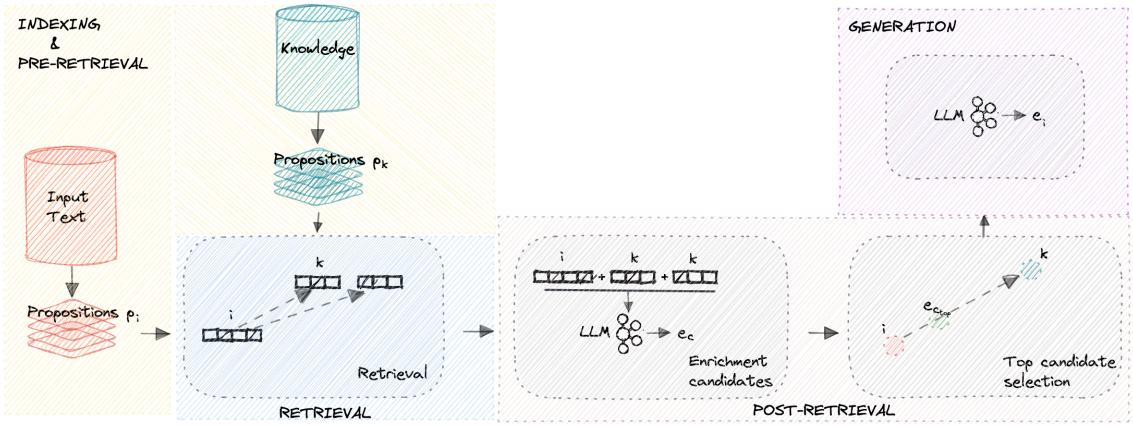



Figure 5.1: Diagram of the proposed model called RE-FIN.

**Indexing** starts with the cleaning and extraction of raw data in PDF, CSV, and TSV formats and converts them into a uniform plain text format. To accommodate the context limitations of language models, text is segmented into sentences delimited by points, becoming smaller and digestible chunks.

**Pre-retrieval process**. In this stage, the primary focus is optimising the indexing structure and the original query. Optimising indexing aims to enhance the quality of the content being indexed. We involve a very little-used strategy proposed by Chen et al. [46], enhancing data granularity, and optimising index structures. We choose propositions as a retrieval unit since the retrieved texts are more condensed with information relevant to the original sentence, reducing the need for lengthy input tokens and minimising the inclusion of extraneous, irrelevant information. Propositions are then encoded into vector representations using an embedding model and stored in a vector database. This step enables efficient similarity searches in the subsequent retrieval phase.

**Retrieval**. Upon receipt of a user query, the RAG system employs the same encoding model utilised during the indexing phase to transform the query into a vector representation. It then computes the similarity scores between the query vector and the vector of chunks within the indexed corpus. The system prioritises and retrieves the top K chunks that demonstrate the greatest similarity to the query. These chunks are subsequently used as the expanded context in the prompt.

**Post-Retrieval Process**. Once the relevant context is retrieved, it's crucial to integrate it effectively with the query. The main methods in the post-retrieval process include re-ranking chunks and context compression. In particular, we utilise an innovative heuristic process to create a new sentence similar to the original one that includes the most relevant documents retrieved.

**Generation**. In this phase, the best sentence enriched created is corrected using an LLM and used as the final version of the sentence.

Now, we are going to describe the method more analytically. Let's consider $I$ as the set of original sentences to be enriched, where $i \in I$, and $K$ as the set of sentences from the knowledge corpus chosen for enriching the original sentences, where $k \in K$. We choose knowledge data from Investopedia

downloaded from huggingface platform[bc], from two of the most important book of Finance [98, 122] and the dataset of financial terms definitions provided by Ghosh et al. [111]. The first task is to extract the propositions that compose the sentences of both the original text and the knowledge. To perform this we utilise the Propositionizer proposed by Chen et al, [46][d], that we call $PROP$, such that

$$p_i = PROP(i) \tag{5.1}$$

where $p_i = (1, ..., n^i)$ and

$$p_k = PROP(k) \tag{5.2}$$

where $p_k = (1, ..., n^k)$. Now we use these propositions to retrieve for each $i$ the most similar document from the knowledge, exploiting the COS $CS_{p_i p_k} = cos(E(p_i), E(p_k))$ such that:

$$r_i = \overset{K}{\underset{k=1}{\max}}(CS_{p_i p_k} | CS_{p_i p_k} > \beta) \tag{5.3}$$

where $r_i$ is the set of documents retrieved and $E$ is encoder-only model[e] provided by huggingface. $\beta$ is a constraint designed to retrieve just those documents composed by a proposition semantically very similar to one proposition of the original text. We add two other constraints to take under control that:

- $k$ would not be too similar to $p_i$ (using $\gamma$); because, in this case, the retrieval could be useless

$$r_i = \overset{K}{\underset{k=1}{\max}}(CS_{kp_i} | CS_{kp_i} < \gamma) \tag{5.4}$$

- $i$ would not be too different to $p_k$ (using $\epsilon$) and $k$ itself (using $\epsilon * 1.2$)); because, in this case, the retrieval could be not adequate if not pejorative

$$r_i = \overset{K}{\underset{k=1}{\max}}(CS_{ip_k} | CS_{ip_k} > \epsilon) \tag{5.5}$$

$$r_i = \overset{K}{\underset{k=1}{\max}}(CS_{ik} | CS_{ik} > \epsilon * 1.2) \tag{5.6}$$

$k$ document that respects all constraints indicated will be used for the enrichment task. This task is performed with the fundamental aid of a decoder-only model, that comes from the work of Jiang et al. [157] and it is provided by HuggingFace[f], that merges $i$ and $k$ to create the enriched sentence $e$. $e$ is the result of three steps:

- produce $\zeta$ candidates

$$e_c = LLM(i, r_i) \tag{5.7}$$

  where $c = (1, ..., \zeta)$
- select $zeta_{top}$ candidates closest to a reference vector $v_i$ positioned between $E(i)$ and $E(r_i)$ with a $\mu$ pace calculated with a *Move Towards* ($MT$) function

$$v_i = MT(E(i), E(r_i) | \mu) \tag{5.8}$$

[b] https://huggingface.co/datasets/infCapital/investopedia_terms_en
[c] https://huggingface.co/datasets/openvega-simon/investopedia
[d] https://huggingface.co/chentong00/propositionizer-wiki-flan-t5-large
[e] https://huggingface.co/intfloat/e5-base-v2
[f] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GPTQ

such that

$$e_{c_{top}} = \max_{\zeta_{top}}(CS_{e_c v_i}) \tag{5.9}$$

- correct $\zeta_{top}$ using the Eq. 5.11 with prompt adjusted and select $e$ with the Eq. 5.9 respecting two last constraints

$$e_i = \max_{1}^{\zeta_{top}}(CS_{e_{c_{top}} v_i} | CS_{ie_i} > \omega) \tag{5.10}$$

where $\omega$ is the minimum semantic similarity between $E(i)$ and $E(e_i)$. The system described above allows us to use propositions for precise retrieval while utilising the entire document for enrichment. This process is enabled by a controlled step that verifies: (i) the semantic similarity of the entire document relative to the original sentence (as described in Eq. 5.6), and (ii) the semantic similarity of the enriched sentence in relation to the original sentence (as described in Eq. 5.10).

### 5.1.4 Evaluation

For the evaluation, we selected three datasets well highly used for FSA, as in the papers we used as main references [367, 193, 381, 77]. **Financial PhraseBank (FPB)** [225]. FPB includes 4,846 news annotated by 16 individuals with adequate background knowledge of financial markets from an investor perspective. Based on the strength of agreement among annotators, it releases four reference datasets, namely 100%, 75%, 66%, and 50% agreement. In their study, Malo et al. argues that the overall sentiment may be different from the prior sentiment polarity of individual words, and incorporating phrase-structure information and domain-specific use of language could improve the detection. We use the 100% agreement dataset. **FiQA Task 1** [216]. The dataset is from FiQA Open Challenge Task 1, which consists of 498 financial news headlines and 675 posts with their target entities, aspects, and corresponding sentiment score. The original dataset has 1173 messages with sentiment scores ranging from -1 to +1. By filtering those scores with an absolute value larger than 0.3, only 771 messages are left and mapped to the positive/negative classes exactly as [367]. **SEntFiN 1.0** [318]. SEntFiN is a human-annotated dataset that includes 10,753 news headlines with their entity and corresponding sentiment. Commonly, multiple entities are present in a news headline with different sentiment expressions and SEntFiN has 2,847 headlines that contain multiple entities, which may have conflicting sentiment. For this reason, we consider in our experiment just those documents without conflict.

| **Dataset** | FPB | FiQA | SEntFiN |
|---|---|---|---|
| Positive | 570 | 507 | 2,832 |
| Negative | 303 | 264 | 2,373 |
| Neutral | 1,391 | - | 2,701 |
| Total Size | 2,264 | 771 | 7,906 |

Table 5.1: Summary statistics for the three FSA datasets (after post-processing).

We conduct our evaluation over 3 different tasks without and with the enrichment process:
**FSA with decoder-only**: Predict sentiment of sentences through a pre-trained LLM.
**FSA with encoder-only**: Fine-tune and predict sentiment through a pre-trained encoder-only model
**Perplexity**

The parameters utilised for the experiments were deducted from a sensitivity analysis. We select these values as optimal: $\beta = 0.8$, $\gamma = 0.95$, $\epsilon = 0.7$, $\zeta = 50$, $\mu = 0.12$, $\zeta_{top} = 5$, $\omega = 0.83$.

**FSA**

First, we tested whether a decoder-only model is better at predicting the sentiment of a sentence after enrichment. We prompted the input sentence, asking the $LLM^f$ to predict the sentiment as either (POSITIVE, NEGATIVE) or (POSITIVE, NEUTRAL, NEGATIVE), depending on the dataset used, employing both zero-shot and few-shot learning. For the few-shot scenario, we randomly selected one example per label from the respective dataset: 3 examples for FPB, 6 for SEntFiN (as it contains twice as many as FPB), and 2 for FIQA. Specifically, we compared the model's accuracy in predicting the sentiment of the dataset with and without enriched sentences to assess whether enrichment aids a pre-trained model in predicting a sentence's financial sentiment (results in Tab. 5.2).

| **Dataset** | FPB | FiQA | SEntFiN |
|---|---|---|---|
| **Decoder-only - Zero-shot** | | | |
| Mistral | 75.8% | 79.1% | 65.4% |
| Mistral + RE-FIN | 86.4% | **87.3%** | 68.1% |
| **Decoder-only - Few-shot** | | | |
| Mistral | 87.6% | 79.9% | 66.9% |
| Mistral + RE-FIN | 91.5% | **87.3%** | 69.6% |
| **Encoder-only - Fine-tuning** | | | |
| DistilBert | 90.6% | 73.1% | 58.8% |
| DistilBert + RE-FIN | **92.8%** | 73.1% | **71.9%** |

Table 5.2: Accuracy for FSA using the encoder-only model, considering only the enriched documents for each dataset.

Following this, we conducted another FSA using a pre-trained encoder-only model[g] that was fine-tuned without and with the enriched sentences (results in Tab. 5.3).

| **Dataset** | FPB | FiQA | SEntFiN |
|---|---|---|---|
| **Decoder-only - Zero-shot** | | | |
| Mistral | 75.1% | 80.9% | 70.7% |
| Mistral + RE-FIN | 79.3% | 83.9% | 71.1% |
| **Decoder-only - Few-shot** | | | |
| Mistral | 86.3% | 80.3% | 67.7% |
| Mistral + RE-FIN | 88.0% | 82.4% | 68.0% |
| **Encoder-only - Fine-tuning** | | | |
| DistilBert | 93.2% | 71.0% | 86.0% |
| DistilBert + RE-FIN | **95.8%** | **85.4%** | **86.7%** |

Table 5.3: Accuracy for FSA. The accuracy reported for the Encoder-only evaluation was computed after 1 epoch.

It is easier to notice the performance increase due to RE-FIN. On average, there is an increase

[g]https://huggingface.co/distilbert/distilbert-base-uncased

of 3.8% utilizing a zero-shot prompt and 4.3% with few-shot learning. The sole exception is the encoder-only model trained exclusively on the augmented data of the FiQA dataset, which exhibits the same performance achieved with the non-augmented data. Nonetheless, decoder-only models that employ data augmented via RE-FIN achieve the highest overall performance for this dataset. Both encoder-only and decoder-only models were chosen with the belief that they offer a strong foundation while being accessible and easy to use for the entire community.

### Perplexity

Perplexity is a measurement that reflects how well a model can predict the next word based on the preceding context. So, we thought that computing the perplexity w/o enrichment could give a reliable measure of improving the clarity and objectivity of sentences. We utilized a commonly used Python library *evalute*[h], testing the two most used LLMs provided: *openai-gpt* and *GPT2* (results in Tab. 5.4).

| **Dataset** | FPB | FiQA | SEntFiN |
|---|---|---|---|
| openai-gpt | 531.4 | 4572.7 | 6072.7 |
| openai-gpt + RE-FIN | <u>384.3</u> | <u>3099.3</u> | <u>5427.3</u> |
| gpt2 | 180.1 | 1162.5 | 1219.4 |
| gpt2 + RE-FIN | **138.2** | **670.5** | **1090.2** |

Table 5.4: Mean Perplexity.

### Ablation Analysis

We conducted an ablation study to evaluate the robustness of our model and the contribution of each component. We tested each dataset with three distinct settings to demonstrate the value of each component of our method. The experiments were carried out on the decoder-only fine-tuning task, as it yielded the best performance, as discussed in the previous section. The approaches we tested are:

- No Retrieval: Sentences are enriched directly using the $\text{LLM}^f$, without any retrieval process or additional steps.
- No Post-Retrieval: The retrieval process is applied as described in Sec. 5.1.3, but sentences are enriched with the $\text{LLM}^f$ without the post-retrieval phase.
- No MT: The complete method is used, except the MT logic, which is responsible for selecting the best-enriched candidate, is removed. Instead, a simpler function based on COS is used to select the candidate most similar to the original sentence.

Fig. 5.2 shows the contribution of the retrieval process compared to enriching sentences without it. The most notable insight from the results is the significant impact of candidate selection criteria. Relying solely on COS resulted in the lowest accuracy for two out of three datasets, emphasising the importance of our MT function in selecting the best-enriched candidate.

[h] https://pypi.org/project/evaluate/

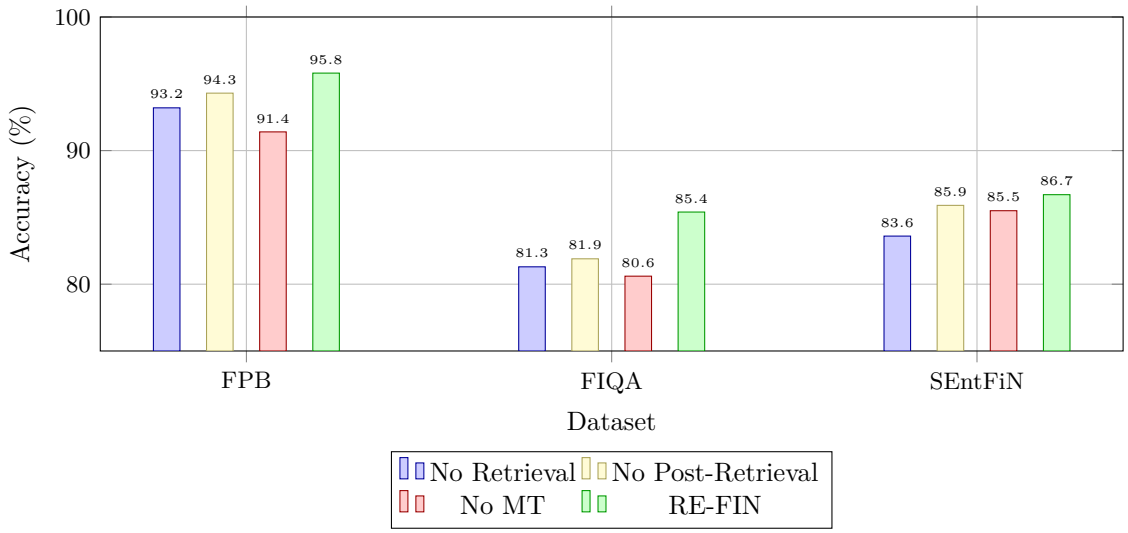



Figure 5.2: Accuracy across datasets (FPB, FIQA, SEntFiN) for different iterations.

### 5.1.5 Results

RE-FIN demonstrates superior performance across all datasets and classification methods tested, with the only exception being the encoder-only model trained with augmented data from the FiQA dataset, which performs similarly to non-augmented data. However, the highest performance for this dataset is achieved by decoder-only models utilising RE-FIN augmented data. The ablation study in Fig. 5.2 shows that every component of RE-FIN positively contributes to overall performance, emphasising its effectiveness in enhancing classification results. Notably, RAG and fine-tuning are not mutually exclusive but can complement each other, enhancing models at different levels [107]. For FPB and SEntFin, their combined use achieves optimal performance.

### 5.1.6 Conclusions

In this paper, we developed a novel RAG methodology that enriches domain-specific sentences with reliable, knowledge-based information. Our model retrieves information based on propositions, seeking sentences that share similar propositions while providing added value. Additionally, it introduces a novel selection criterion to choose the candidate that best integrates the input sentence with information from retrieved documents. Experimental results on three FSA datasets show that RE-FIN consistently improves sentiment analysis performance across all datasets, achieving superior accuracy compared to existing methods. The ablation study indicates that each component of RE-FIN enhances its overall effectiveness. The RE-FIN tool is released as a free and open-source resource for the research community[i], enabling broader access and advancing FSA.

[i] https://github.com/filippopallucchini/RE-FIN

### Acknowledgments

This work is partially supported within the research activity of an Italian Project entitled "ISALDI: Interpretable Stock Analysis Leveraging Deep multImodal models" - PRIN 2022 Project number 2022P4MPAP - in which some of the authors are involved as coordinators and researchers.

## 5.2 Self-explanatory and Retrieval-augmented LLMs for Financial Sentiment Analysis

Enriching sentences with qualitative knowledge is crucial for enhancing sentiment prediction and making the most of the available labelled data for training models. This is particularly important in domains like the financial one, where texts are usually brief and contain much-implied information. In this article, we introduce FLEX (Financial Language Enhancement with Guided LLM Execution), an automated system capable of retrieving information from a LLM to enrich financial sentences, making them more knowledge-dense and explicit. FLEX generates multiple potentially enhanced sentences and uses a new logic to determine the most suitable one. To mitigate hallucinations in LLMs, we developed a new algorithm to select the most appropriate sentences. This approach ensures the original meaning is preserved, reduces excessive syntactic similarity between versions, and maintains the lowest possible perplexity. These enhanced sentences are more interpretable and directly useful for downstream tasks like FSA. Compared to SOTA methods, FLEX shows improvements in the accuracy of processing FSA tasks.

### 5.2.1 Introduction

FSA, which broadly encompasses the study of investor sentiment and financial textual sentiment [78], is a key domain within sentiment analysis. Given the complex nature of financial markets, individuals involved in different market conditions display varied cognitive patterns [231, 229], making it difficult to dynamically understand and analyse the market for sound financial decision-making. To tackle the challenges arising from the market's constant fluctuations, automated FSA has garnered significant attention over the past decade [345]. It has proven to be a powerful tool for supporting business decision-making and conducting financial forecasting [212, 213]. Applications include corporate disclosures, annual reports, earnings calls, financial news, social media interactions, and more [345, 369]. Sentiment analysis is a complex, domain-dependent problem. This domain dependence is especially pronounced in the finance sector [225] due to the focused nature of financial topics and the use of highly specialised language [234, 233, 386]. For instance, words like *liability* and *debt* are typically viewed negatively in general-purpose sentiment analysis, but they often carry a neutral meaning in the financial context [78, 370]. Some authors have tackled these challenges through embedding alignment, which has proven effective in adapting models to specialised domains [67, 221], although performance remains predominantly inconsistent in the field of FSA [208]. LLMs have matured significantly and gained widespread adoption across various domains and everyday tasks [288]. This advancement has also had a profound impact on the financial industry.

Recent models, such as ChatGPT[j] and GPT-4[k], trained with RLHF [56] and masked language

[j] https://platform.openai.com/docs/models/gpt-3-5
[k] https://platform.openai.com/docs/models/gpt-4

model objectives, have demonstrated exceptional capabilities across a wide range of NLP tasks[22, 262, 232]. These LLMs are trained on datasets covering diverse genres and topics. While their performance in general NLP tasks is impressive, their applicability and effectiveness in specific domains like finance still require further exploration, as their impact could span a wide range of applications [193, 76]. In the financial domain, LLMs are increasingly crucial for tasks such as investment sentiment analysis, financial named entity recognition, and QA systems to assist financial analysts [193].

However, directly applying LLMs for FSA presents two main challenges. First, the difference between the objective functions used in LLMs' pre-training and the goal of predicting financial sentiment can cause LLMs to inconsistently output labels for FSA [266, 334]. Second, the typical sources for FSA, like news flashes and tweets, are often concise and lack sufficient background information [381, 345]. This information scarcity not only affects human judgment [225] but also poses a significant challenge for LLMs in making accurate predictions [381].

To tackle these challenges, our study introduces a retrieval-augmented LLM framework for FSA. Similar to how a makeup artist enhances a person's features [394], we propose a novel method for making a sentence more understandable and self-explanatory without altering its essence, which is critical for real-world AI applications [41, 39]. This approach highlights financial concepts and implicit propositions by retrieving relevant information from an LLM, thereby improving FSA. We provide experimental evidence, demonstrating the model's effectiveness on two analytical tasks, namely, perplexity and FSA, using three benchmark datasets.

The contributions of this work can be summarised as follows:

**(1)** We propose a novel approach that incorporates semantic similarity and perplexity to enhance the decision-making logic and interpretability of a predictive model for FSA tasks.
**(2)** We demonstrate that our model's enrichment approach improves results, even when using the original dataset for fine-tuning models or making direct predictions.
**(3)** We provide the code freely to the community, promoting accessibility and further research[1]

For related works please refer to Sec. 5.1.2.

### 5.2.2 Method

The framework of our proposed model is sketched in Fig. 5.3, which consists of two main phases.

1) The **MAKEUP phase** generates enriched sentence candidates that maintain the exact semantics of the original text while clarifying specific financial concepts or making implicit propositions more explicit.

2) The **MAKEUP SELECTION phase** consists of a function that selects the most appropriate candidate from those produced in the previous phase. Using an embedding model, the function encodes both the original sentence and each candidate into vector representations to assess semantic similarity. In addition to semantic similarity, we also consider it crucial to ensure the sentence is clear and self-explanatory. Thus, the function also incorporates a measure of perplexity, which reflects how well the sentence aligns with the language model's expectations, providing insight into its readability and naturalness.

[1] https://github.com/filippopallucchini/FLEX

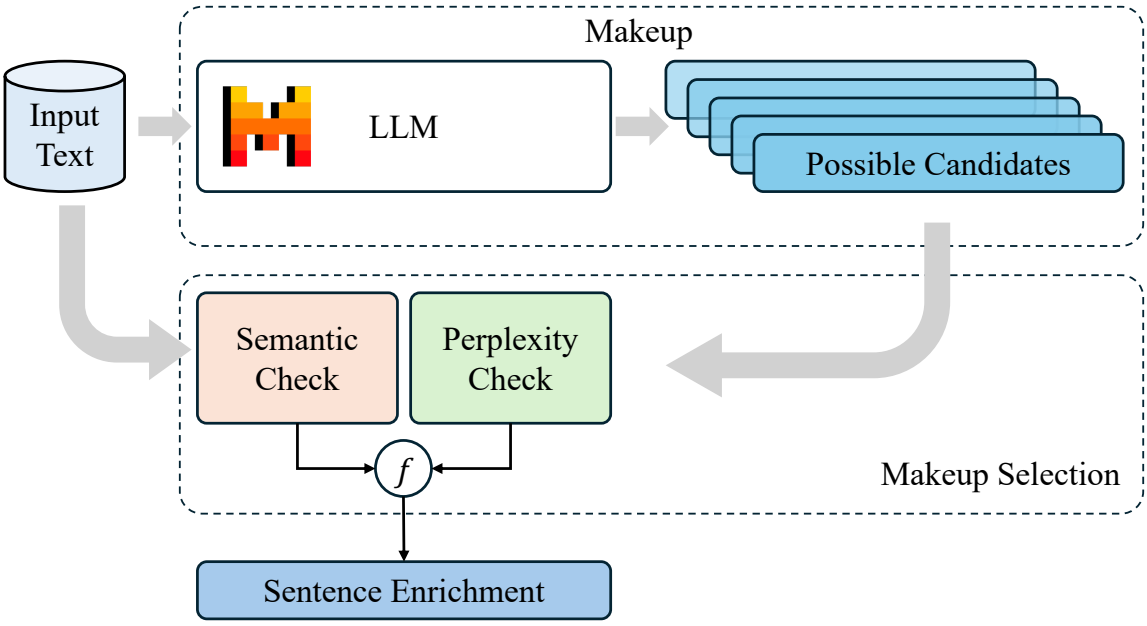



Figure 5.3: Diagram of the proposed FLEX method.

Specifically, given $I$ is the set of original sentences to be enriched, where $i \in I$, the sentence is passed to a decoder-only model, which comes from the work of Jiang et al. [157] and it is provided by HuggingFace[f]. It performs the MAKEUP phase that consists in creating $\zeta$ possible candidates enriched $e$

$$e_c = LLM(i), \tag{5.11}$$

where $c = (1, ..., \zeta)$. The process of MAKEUP SELECTION is performed by a function that considers both the semantic similarity and the perplexity. In the *Sematic Check*, each $i$ and $e_c$ is embedded in an encoder-only model[m] $E$, provided by huggingface, and used to compute COS, named $CS$

$$CS_{ie_c} = cos(E(i), E(e_c)) \tag{5.12}$$

where $CS_{ie_c} \in [0, 1]$.

At the same time, in the *Perplexity Check*, each $i$ and $e_c$ are passed to the commonly used Python library *evalute*[n] parametrized with *openai-gpt*, producing $P$ ($Perplexity$)

$$P_{e_c} = Perplexity(e_c) \tag{5.13}$$

where $P_{e_c} \in [0, \infty]$.

Thus, the $MS$ function combines these two measures to identify the best candidates that both maximise the semantic similarity score $CS_{ie_c}$ and minimize the perplexity $P_{e_c}$. The function is defined as:

[m] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[n] https://pypi.org/project/evaluate/

$$e_i = \max_{c=1}^{\zeta} \left( CS_{ie_c} + \left( 1 - \frac{\log(1 + P_{e_c})}{\log(1 + P_{e_{c_{\max}}})} \right) \middle| \, CS_{ie_c} > \omega, \; P_{e_c} < P_i \right) \tag{5.14}$$

where the condition $CS_{ie_c} > \omega$ ensures that the selected enriched sentences have a minimum threshold of semantic similarity to the original sentence. This $\omega$ value depends on the maximum level of enrichment that we want to allow. Indeed, if a Dataset is composed by short sentences, we will let more contribute to the LLM and vice-versa. The second condition, $P_{e_c} < P_i$, guarantees that the chosen sentence is clearer than the original one. The system described above allows us to control the process of enrichment, avoiding as much as possible the risk taken by the LLM.

### 5.2.3 Experiments

#### Datasets

Three datasets are used for the evaluation process. They are the most used dataset in FSA in the literature, particularly in the papers we used as main references like [367, 193, 381, 79]. The statistics of the employed datasets can be viewed in Tab. 5.1. The description of the datasets is reported in Sec. 5.1.4.

#### Setups

We evaluate our approach using three different frameworks, as done with RE-FIN:

- FSA with decoder-only zero-shot: Predict sentence sentiment using a pre-trained LLM in a zero-shot setting.
- FSA with decoder-only few-shot: Predict sentence sentiment using a pre-trained LLM with a few-shot prompt.
- FSA with encoder-only: Fine-tune and predict sentiment using a pre-trained encoder-only model.

First of all, we tested if a decoder-only model is facilitated in predicting the sentiment of a financial sentence after the enrichment. So, we prompt the input sentence, asking the $\text{LLM}^f$ to predict the sentiment (POSITIVE, NEGATIVE) or (POSITIVE, NEUTRAL, NEGATIVE) depending on the nature of the dataset used, both with zero-shot learning and few-shot learning.

For the few-shot scenario, we randomly selected one example per label from the respective dataset: 3 for FPB, 6 for SEntFiN (which contains twice as many as FPB), and 2 for FIQA. Specifically, we compared the model's accuracy in predicting the sentiment of the dataset with and without enriched sentences to assess whether enrichment aids a pre-trained model in predicting a sentence's financial sentiment.

Following this, we conducted another FSA using an encoder-only model that was pre-trained and fine-tuned without enriched sentences.

| Dataset | FPB | FiQA | SEntFiN | Avg. |
|---|---|---|---|---|
| **Decoder-only - Zero-shot** | | | | |
| Mistral | 75.1% | 80.9% | 70.9% | 75.7% |
| Mistral + FLEX v0.2 | 83.5% | 86.8% | 73.7% | 81.3% |
| Mistral + FLEX v0.3 | 85.3% | 89.5% | 76.1% | 83.6% |
| **Decoder-only - Few-shot** | | | | |
| Mistral | 87.2% | 80.5% | 67.7% | 78.5% |
| Mistral + FLEX v0.2 | 90.1% | 87.9% | 75.3% | 84.4% |
| Mistral + FLEX v0.3 | 90.3% | 90.3% | 77.3% | 86.0% |
| **Encoder-only - Fine-tuning** | | | | |
| DistilBert | 93.2% | 71% | 86.0% | 83.4% |
| DistilBert + FLEX v0.2 | 94.7% | 82.6% | 84.4% | 87.2% |
| DistilBert + FLEX v0.3 | 95.4% | 91.6% | 86.0% | 91.0% |

Table 5.5: Performance comparison measured by accuracy.

**Baselines**

We examine our method by comparing to the following baselines: **Mistral 7B** is a 7-billion-parameter language model optimised for high performance and efficiency in NLP tasks. It incorporates several technical innovations that enhance its functionality, including Grouped-Query Attention, Sliding Window Attention, and advanced fine-tuning capabilities. We examine two versions of Mistral, namely, Mistral-7B-Instruct-v0.2-GPTQ and Mistral-7B-Instruct-v0.3-GPTQ for generating the enrichment, respectively. These models are termed FLEX v0.2 and FLEX v0.3. Then, we use Mistral-7B-Instruct-v0.2-GPTQ as a backbone to predict labels. We compare the vanilla version of Mistral-7B-Instruct-v0.2-GPTQ (Mistral) to the ones that combine the enrichment generated by FLEX v0.2 (Mistral + FLEX v0.2) and FLEX v0.3 (Mistral + FLEX v0.3), respectively.

**DistilBERT** is a compressed version of the BERT model, developed to be smaller, faster, and more efficient while maintaining strong language understanding capabilities. Its key innovations include the use of knowledge distillation during the pre-training phase, a novel triple-loss function that integrates language modeling, distillation, and cosine-distance losses, as well as the implementation of large-batch training with gradient accumulation and dynamic masking. Similarly, we also examine the enrichment generated by different Mistral versions, e.g., FLEX v0.2 and FLEX v0.3.

### 5.2.4 Results

The performance results of Mistral and DistilBERT models, both with and without the integration of FLEX versions, are summarised in Tab. 5.5. For decoder-only models (Mistral), the integration of FLEX v0.2 and v0.3 shows a clear improvement in both zero-shot and few-shot learning tasks across all datasets. In the zero-shot setting, Mistral combined with FLEX v0.2 achieves an average accuracy of 81.3%, compared to 75.7% without FLEX, demonstrating a substantial improvement. Mistral + FLEX v0.3 further increases accuracy to 83.6%, with notable gains particularly in the FPB and SEntFiN datasets. Similarly, in the few-shot setting, Mistral + FLEX v0.2 achieves an average accuracy of 84.4%, instead Mistral + FLEX v0.3 improves the accuracy to 86.0%.

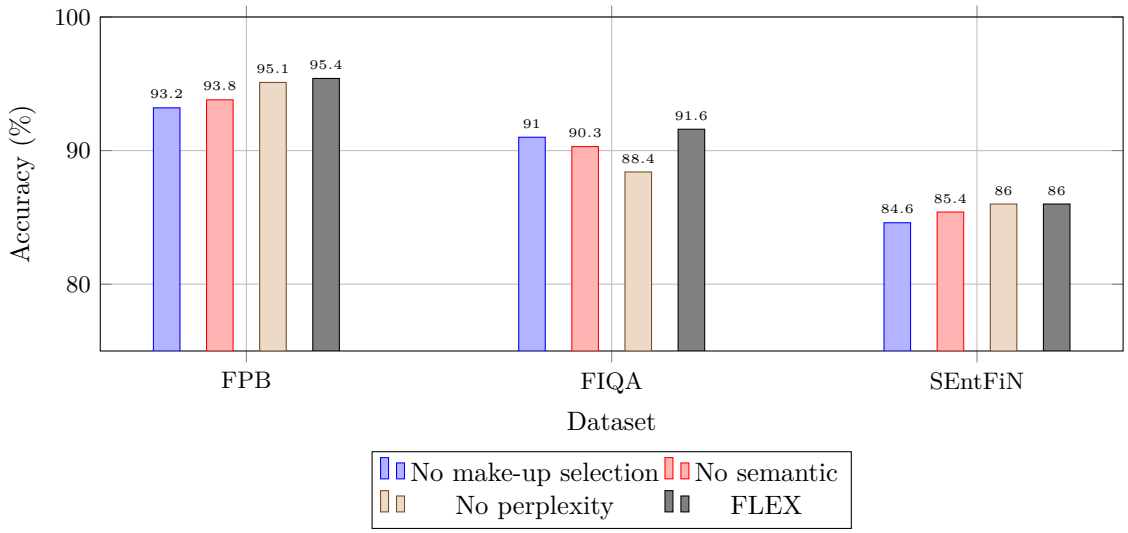



Figure 5.4: Accuracy across datasets (FPB, FIQA, SEntFiN) for different iterations.

For encoder-only models (DistilBERT), the results are particularly impressive in the fine-tuning setting, where the integration of FLEX v0.2 and v0.3 significantly boosts performance. DistilBERT alone achieves an average accuracy of 83.4%, but with FLEX v0.2, this increases to 87.2%, with the most substantial improvement observed in the FiQA dataset, where accuracy rises from 71% to 82.6%. FLEX v0.3 further enhances the model's accuracy to 91.0%, with gains across all datasets, including a notable jump in FiQA accuracy to 91.6%. These results indicate that the FLEX framework, particularly version 0.3, is highly effective in improving fine-tuning performance for FSA tasks, enabling models to better capture domain-specific nuances in complex financial data.

In summary, we can observe that FLEX can provide positive utilities across all the evaluation setups.

### Ablation Study

We conducted an ablation study to enhance the robustness of our model and underscore the contribution of each component. Specifically, we tested each dataset using three distinct approaches, each designed to illustrate the importance of individual elements of our method. The experiments were performed on the decoder-only fine-tuning task, as it yielded the best performance, as discussed in the previous section. The approaches we tested are:

- **No Makeup Selection:** Sentences are enriched directly using the $\text{LLM}^f$ without the makeup selection phase, meaning the LLM produces only one candidate.
- **No Semantic:** The LLM generates candidates as described in Sec. 5.2.2, but the selection is based solely on the perplexity criterion, choosing the enriched sentence with the lowest perplexity value.
- **No Perplexity:** The LLM produces candidates as outlined in Sec. 5.2.2, but the selection

is made solely using the semantic criterion, choosing the enriched sentence with the highest COS to the original sentence.

Fig. 5.4 first highlights the value of generating multiple candidates rather than relying solely on the LLM's output. Additionally, the results reveal the significant impact of both components in our candidate selection criteria. The notable differences in trends between FPB, SEntFiN, and FIQA can be attributed to the varying sizes of these datasets. The FIQA dataset contains fewer documents compared to the other two, making longer sentences beneficial for the encoder during the fine-tuning training phase. This explains why the model without restrictions achieves the second-best performance.

Notably, the lowest performance is observed in the "No perplexity" solution, as it relies solely on semantic similarity with the original sentence, inherently resulting in shorter sentences. Nevertheless, the higher-quality sentences resulting from our selection criteria yield the highest accuracy. This can be explained by the limitations of using perplexity or semantic similarity alone. Perplexity by itself does not consider the similarity to the original sentence, which is crucial as it carries the sentiment label. Conversely, relying solely on semantic similarity risks tying the enriched sentence too closely to the original one, without adding meaningful content. For this reason, our approach uses perplexity to ensure that the new sentence is more predictable than the original while maintaining relevant semantic connections.

**Case Study**

In this section, we conduct case studies on the three datasets employed to qualitatively illustrate how our method is able to provide additional information that is useful for more accurate predictions.

As shown in Tab. 7.3, FLEX can paraphrase the original sentence such that the expression is more straightforward while also fitting to the financial domain. For instance, in the first example, our method reworded *"net loss narrowed"* to *"net loss was reduced"*; and in the second example, *"short"* to *"shorting shares of stock"*. The reworded expressions are easier to understand to both human and the machine, as indicated by the predictions with and without FLEX.

From the results, we can also observe that FLEX is able to correctly interpret and elaborate on the financial phenomena described in the given input, strengthening the intended sentiment of the original text. For instance, in the first example, FLEX further explained the stated fact by adding *"reflecting a notable improvement …"*; in the second example, shorting shares of stock is interpreted a *"bearish bet"* and *"with the expectation that its price will decrease …"*; in the third, *"a third red day in a row"* is elaborated with *"indicates a bearish trend"*; in the fourth, *"beats profit estimates"* is expanded as *"earnings …were higher than the consensus forecast"*; and in the fifth, *"oversubscribe"* is elaborated to emphasize *"demand …exceeded the supply"*. It can be seen that the elaborations are not only true to the original meanings, but also contain explicit sentiment indicators that aid the machine's prediction.

On the other hand, FLEX is also capable of expanding on facts unrelated to financial sentiment. For instance, in the sixth example, it added an explanation that the stated phenomenon in the original sentence is a standard practice in handling fuel, which helped the machine classify it as neutral in the financial context.

Furthermore, from the last example, it can be observed that FLEX is also capable of completely

| Setup | Content | Predicted | True |
|---|---|---|---|
| *Original* | In the second quarter of 2010, Raute's net loss narrowed to EUR 123,000 from EUR 1.5 million in the same period of 2009. | NEGATIVE | POSITIVE |
| *w/ FLEX* | Raute's net loss was reduced from EUR 1.5 million in the second quarter of 2009 to EUR 123,000 in the same quarter of 2010, reflecting a notable improvement in the company's financial situation. | POSITIVE | |
| *Original* | $AAPL short some 592.49 | POSITIVE | NEGATIVE |
| *w/ FLEX* | Shorting 592.49 shares of $AAPL stock is a bearish bet on the stock, with the expectation that its price will decrease in the near term and generate a profit from the subsequent decline. | NEGATIVE | |
| *Original* | 3rd red day in a row? $TSLA | POSITIVE | NEGATIVE |
| *w/ FLEX* | A third red day in a row for $TSLA's stock price indicates a bearish trend. | NEGATIVE | |
| *Original* | EXL beats profit estimates, cuts sales outlook | NEGATIVE | POSITIVE |
| *w/ FLEX* | EXL beats profit estimates, signifying that the company's earnings per share for the quarter were higher than the consensus forecast. | POSITIVE | |
| *Original* | Religare Finvest NCD issue oversubscribed 1.31 times | NEGATIVE | POSITIVE |
| *w/ FLEX* | Religare Finvest NCD issue oversubscribed 1.31 times, meaning the demand for the non-convertible debentures exceeded the supply by 31%. | POSITIVE | |
| *Original* | The solid fuel is heated before sludge is mixed therein. | POSITIVE | NEUTRAL |
| *w/ FLEX* | The solid fuel is heated before sludge is mixed therein, a standard practice to ensure the fuel is free of impurities before the sludge is added. | NEUTRAL | |
| *Original* | Homebuilders – $RYL breaking below support, watch this one. $SPY | POSITIVE | NEGATIVE |
| *w/ FLEX* | The homebuilding industry is experiencing a downturn, and the stock price of $RYL is reflecting this trend, as it has broken below a significant support level and is now trading below the $SPY index. | NEGATIVE | |

Table 5.6: Case study illustrating prediction differences before and after applying FLEX rewriting.

rewriting a sentence when the input is too informally worded. The rewritten sentence is true to the original meaning, i.e. *"RYL breaking below support"*, while being clearer and more comprehensible for both humans and machines.

### 5.2.5 Conclusion

In this work, we proposed a FSA approach that enhances the original input by incorporating semantic enrichment and perplexity measures, enabling the predictive model to make more accurate label predictions, based on a more comprehensive and coherent input.

Experimental results validate the effectiveness of our method across various setups, including zero-shot, few-shot, and fine-tuning frameworks. The ablation study further confirms the utility of the introduced semantic and perplexity checks. Future work can examine the enhancement of the method from the perspective of syntax [385] and pragmatics [230].

# Part III

# Trustworthy and Interpretable Alignment in LLMs

The final part of the thesis deepens the concept of alignment by addressing the internal mechanisms that govern LLM behaviour. While previous sections focused on aligning embeddings and outputs, here we aim to align the representations themselves with human-understandable semantics. This part begins with an exploration of hallucination as a phenomenon rooted in the model's representational dynamics, leading to the proposal of SAFE, a sparse autoencoder-based framework for robust query enrichment and factual consistency. Building on this, SFAL introduces a principled method for quantifying the semantic–functional correspondence of latent features, offering a new tool to assess interpretability in mechanistic terms. Altogether, Part III contributes to the emerging field of epistemic alignment, advancing the goal of transparent and trustworthy artificial intelligence.

# Chapter 6

# Background

## 6.1 Hallucination Mitigation in Large Language Models

The extensive factual knowledge encoded within LLMs has demonstrated considerable advancements in leveraging LLMs for information seeking [6, 279], potentially reshaping the landscape of IR systems [395]. Nevertheless, in tandem with these remarkable advancements, concerns have arisen about the tendency of LLMs to generate hallucinations [22, 126], resulting in seemingly plausible yet factually unsupported content. Further compounding this issue is the capability of LLMs to generate persuasive and human-like responses [301], which makes detecting these hallucinations particularly challenging, thereby complicating the practical deployment of LLMs, especially real-world IR systems that have integrated into our daily lives like chatbots (e.g. Cluade[a] or ChatGPT[b]), search engines (e.g. Perplexity[c] or Bing[d]), and recommender systems [106, 186]. Given that the information provided by these systems can directly influence decision-making, any misleading information has the potential to spread false beliefs, or even cause harm. As discussed by Huang et al. [147], hallucinations arise when the probabilistic model underlying text generation assigns high probability to outputs that are linguistically plausible but epistemically false.

### 6.1.1 Defining Hallucination

The concept of hallucination traces its roots to the fields of pathology and psychology and is defined as *the perception of an entity or event that is absent in reality* [214]. Within the realm of NLP, hallucination is typically referred to as a phenomenon in which the generated content appears nonsensical or unfaithful to the provided source content [96, 236]. From a probabilistic perspective, a hallucination occurs when the generated output $y$ is inconsistent with the true distribution of facts $\mathcal{F}$ given an input context $x$. If $P_\theta(y|x)$ denotes the model's conditional probability, and $P^*(y|x)$

[a] https://claude.ai/
[b] https://openai.com/blog/chatgpt
[c] https://www.perplexity.ai/
[d] https://www.bing.com/new

the ideal factual distribution, hallucination corresponds to a divergence between them:

$$\text{Hallucination Risk:} \quad D_{\mathrm{KL}}(P^*(y|x) \,\|\, P_\theta(y|x)) > 0 \tag{6.1}$$

where $D_{\mathrm{KL}}$ is the Kullback–Leibler divergence. When $P_\theta$ overestimates the likelihood of unsupported continuations, the model produces factually incorrect but coherent text.

Jurafsky and Martin [163] distinguish two main forms of hallucination:

- **Intrinsic hallucination**: when the generated output contradicts the source content.
- **Extrinsic hallucination**: when the generated output cannot be verified form the source.

Moreover, Huang et al. [147], considering that LLMs place a significant emphasis on user-centric interactions and prioritise alignment with user directives, coupled with the fact that their hallucinations predominantly surface at factual levels, introduce a more granular taxonomy building upon the foundational work by Ji et al. [156]. This refined taxonomy seeks to encapsulate the distinct intricacies associated with LLM hallucinations. We present examples for each type of hallucination in Fig. 6.1, namely factuality hallucination and faithfulness hallucination.

**Factuality hallucination** emphasises the discrepancy between generated content and verifiable real-word facts, typically manifesting as factual inconsistencies. These factual hallucinations are divided into two primary types:

- *Factual Contradiction* refers to situations where the LLM's output contains facts that can be grounded in real-world information but present contradictions. This type of hallucination occurs most frequently and arises from diverse sources, encompassing the LLM's capture, storage, and expression of factual knowledge. Depending on the error type of contradictions, it can be further divided into two subcategories: entity-error hallucination and relation-error hallucination. The first one refers to the situations where the generated text of LLMs contains erroneous entities, and the second one refers to instances where the generated text of LLMs contains wrong relations between entities.
- *Factual Fabrication* refers to instances where the LLM's output contains facts that are unverifiable against established real-world knowledge. This can be further divided into unverifiability hallucination and overclaim hallucination. The first one pertains to statements that are entirely non-existent or cannot be verified using available sources, and the second one involves claims that lack universal validity due to subjective biases.

**Faithfulness hallucination** captures the divergence of generated content from the user input or the lack of self-consistency within the generated content. This category is further subdivided into three categories:

- *Instruction inconsistency* refers to the LLM's outputs that deviate from a user's directive. While some deviations might serve safety guidelines, the inconsistencies here signify unintentional misalignment with non-malicious user instructions.
- *Context inconsistency* points to instances where the LLM's output is unfaithful to the user's provided contextual information.

| Category | Type | Example Dialog | Explanation |
|---|---|---|---|
| Factuality Hallucination | Factual Contradiction | *Instruction*: What are Thomas Edison's main contributions to science and technology?<br>*Response*: Thomas Edison developed the first practical telephone and invented the light bulb. | The response is factually incorrect. In reality, *Edison improved the light bulb, building on earlier designs by others*, and *Alexander Graham Bell invented the telephone.* |
| | Factual Fabrication | *Instruction*: What are the major environmental impacts of the construction of the Eiffel Tower?<br>*Response*: The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement. | The response erroneously states that the construction of the Eiffel Tower led to the extinction of the "Parisian tiger" and sparked the global green architecture movement. In fact, *there is no evidence of a "Parisian tiger" ever existing, making this a fabricated claim.* Moreover, *attributing the origins of green architecture to the Eiffel Tower is an exaggeration*, as this movement has diverse roots and cannot be traced back to a single event. |
| Faithfulness Hallucination | Instruction Inconsistency | *Instruction*: Translate the English question into Spanish: "What is the capital of France?"<br>*Response*: The capital of France is Paris. | The user explicitly provided an instruction to translate the English question into Spanish. *Instead of performing the translation as instructed, the LLM chose to answer the question.* |
| | Context Inconsistency | *Retrieved context*: The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea.<br>*Instruction*: Summarize the retrieval document above.<br>*Response*: The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean. | The summary incorrectly states that the Nile originates from the "mountain ranges of central Africa", while the *paragraph mentions that the Nile originates in the Great Lakes region of central Africa.* |
| | Logical Inconsistency | *Instruction*: Solve the equation 2x + 3 = 11 step by step.<br>*Response*: Step 1: Subtract 3 from both sides to get: 2x = 8. Step 2: Divide both sides by 2 to get: x = 3. So the result of the equation is x = 3. | The first step correctly isolates the term with x to get 2x = 8. However, *the next step inaccurately divides 8 by 2 to yield a result of x = 3*, which is inconsistent with the earlier reasoning. |

**Content marked in Red represents the hallucinatory output, while content marked in Blue indicates user instruction or provided context that contradicts the LLM hallucination.**

Figure 6.1: Examples of Each Category of LLM Hallucinations proposed by Huang et al. [147]

- *Logical inconsistency* underscores when LLM outputs exhibit internal logical contradictions, often observed in reasoning tasks. This manifests as inconsistency both among the reasoning steps themselves and between the steps and the final answer.

### 6.1.2 Sources of Hallucination and mitigation strategies

According to Huang et al. [147], hallucinations originate from three primary sources:

1. **Data-level biases**: noisy, contradictory, or incomplete training data. Data for training LLMs are comprised of two primary components: (1) pre-training data, through which LLMs acquire their general capabilities and factual knowledge [392], and (2) alignment data, which teach LLMs to follow user instructions and align with human preferences [353]. Although these data constantly expand the capability boundaries of LLMs, they inadvertently become one of the principal contributors to LLM hallucinations. This primarily manifests in three aspects: the presence of misinformation and biases in the flawed pre-training data sources (e.g.

fake news or societal biases), the knowledge boundary inherently bounded by the scope of the pre-training data (e.g. the rapid evolution of world knowledge or the inability of LLMs to memorise all factual knowledge encountered during the pre-training), and the hallucinations induced by inferior alignment data (e.g. Genkhman et al. [108] found that LLMs struggle to acquire new knowledge during Finetuning).

2. **Model-level issues**: overconfidence, miscalibrated probabilities, and loss-objective mismatch. Pre-training constitutes the foundational stage for LLMs, predominantly utilising a transformer-based architecture as we seen in the Sec. 4.1. This stage employs a CLM objective, where models learn to predict subsequent tokens solely based on preceding ones in a unidirectional, left-to-right manner. While facilitating efficient training, it inherently limits the ability to capture intricate contextual dependencies, potentially increasing risks for the emergence of hallucination [199]. Moreover, during post-training phases like finetuning, overfitting on new factual knowledge encourages LLMs prone to fabricating content, amplifying the risk of hallucination [108]
3. **Inference-level behaviours**: Decoding plays a crucial role in LLM performance, but flawed strategies can cause hallucinations. Imperfect decoding strategies like stochastic sampling [90, 141] add diversity but increase hallucination risk by sampling rare tokens [57, 83, 239]. Moreover, over-confidence leads LLMs to prioritise fluency over context, reducing attention to source material and causing faithfulness hallucinations [23, 47, 206]. Finally, the Softmax bottleneck limits expressivity in output distributions, preventing accurate word prediction and contributing to hallucinations [44, 375].

For instance, an LLM trained to maximise likelihood

$$L_{\text{MLE}} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{1:t-1}, x) \tag{6.2}$$

is incentivised to reproduce frequent textual patterns rather than factual accuracy, leading to fluency without truthfulness.

Now, we will see the hallucination mitigation approaches for each of the three major source levels.

### Data-Level Mitigation

Data-centric methods address hallucinations at their root: the **data-related causes** of misinformation, bias, and knowledge gaps in pre-training corpora. As detailed by Huang et al. [147], such hallucinations emerge from noisy, redundant, or outdated data sources that misinform the model's learned representations. Three main data-level strategies are widely employed: *data filtering*, *model editing*, and *RAG* (see Sec. 4.2).

**1. Data Filtering and Curation.** An intuitive approach to mitigating data-related hallucinations involves the careful selection of high-quality pre-training data from trustworthy and verifiable sources. Radford et al. [287], as early as GPT-2, emphasized restricting web scrapes to pages curated by human experts to minimize misinformation. Modern large-scale datasets, however, preclude manual curation, motivating automated data filtering pipelines.

Filtering methods operate in two primary stages:

1. **Source filtering**: inclusion of factual, domain-reliable data (e.g., The Pile [105], textbook-like corpora [128]) and up-sampling verified academic or encyclopedic sources to enhance factuality [148].
2. **Deduplication**: removal of repeated or near-duplicate entries to prevent overfitting and hallucinated memorisation. Exact duplicates are removed through substring matching or suffix arrays [148], while semantic duplicates are detected via embedding similarity or hashing algorithms such as MinHash [33] and SemDeDup [1].

By reducing bias and redundancy at the data source, filtering effectively mitigates the root causes of hallucinations. Yet, its scalability remains a limitation: large corpora require efficient, automated filtering capable of handling LLM-generated synthetic data, which itself may introduce novel biases.

**2. Model Editing.** While data filtering addresses the input distribution, **model editing** targets the internal parameter space to correct specific factual errors or inject up-to-date knowledge. Model editing algorithms adjust selected weights without full retraining, thus efficiently modifying factual knowledge representations. They can be categorised into two main paradigms:

- **Locate-then-edit methods**, such as ROME [240] and MEMIT [241], identify the "buggy" parameters associated with a fact and update them via local gradient adjustments:

$$\Delta \mathbf{W}_l = -\eta \, \nabla_{\mathbf{W}_l} L_{\text{edit}}(x, y^*) \tag{6.3}$$

  where $L_{\text{edit}}$ measures the mismatch between the model's current output and the factual correction $y^*$ at layer $l$.

- **Meta-learning approaches**, such as MEND [252] and MALMEN [328], learn a hypernetwork that predicts parameter shifts $\Delta \mathbf{W}$ conditioned on input facts:

$$\Delta \mathbf{W} = H_\psi(x, y^*) \approx \arg\min_{\Delta \mathbf{W}} L_{\text{edit}}(\mathbf{W} + \Delta \mathbf{W}) \tag{6.4}$$

  enabling fast, low-rank updates while preserving general knowledge.

Despite their precision, large-scale model edits can degrade global coherence. Hence, recent research explores constrained optimisation to ensure that $\|\Delta \mathbf{W}\|_2$ remains bounded, preserving overall model stability.

**3. RAG.** As detailed in Sec. 4.2, RAG mitigates hallucinations caused by **knowledge gaps** by conditioning LLM outputs on retrieved, factual evidence. By decoupling factual information from parametric memory, RAG dynamically supplements missing or outdated knowledge. Following Huang et al. [147], retrieval strategies can be categorised as:

- **One-time retrieval**, which prepends retrieved text to the input prompt, like Ram et al. [292]:

$$x' = [x; R_x], \quad y \sim P_\theta(y|x') \tag{6.5}$$

- **Iterative retrieval**, which continuously refines retrieval during generation, conditioning each new step on prior reasoning, like [134, 341] that try to incorporate external knowledge at each reasoning step:

$$R_{t+1} = \text{Retriever}(x, y_{1:t}) \tag{6.6}$$

- **Post-hoc retrieval**, which verifies and edits initial outputs against external evidence (e.g., verify-and-edit frameworks [390]).

Although RAG effectively addresses hallucinations stemming from incomplete knowledge, it may introduce *retrieval noise* when irrelevant documents dominate top-$k$ results.

### Model-Level Mitigation

Model-level mitigation techniques address hallucinations that arise from training and architectural factors. These include issues of calibration, misalignment, and representational deficiencies.

**1. Confidence Calibration and Architectural Refinements.** Pretraining-related hallucinations often stem from intrinsic limitations in model architecture and representational calibration. As Huang et al. [147] note, unidirectional architectures and diffuse attention patterns can fragment factual dependencies across context windows. Improving **confidence calibration** complements these architectural refinements by correcting overconfident predictions that lead to factual errors. Following Jurafsky [163], the calibrated probability distribution is defined via temperature scaling:

$$\tilde{P}_\theta(y_t|x) = \frac{P_\theta(y_t|x)^{1/T}}{\sum_{y'} P_\theta(y'|x)^{1/T}} \tag{6.7}$$

where $T > 1$ reduces overconfidence and mitigates hallucination risk by smoothing sharp probability peaks. In parallel, architectural advances such as **bidirectional autoregressive modeling** [199] and **attention-sharpening regularisers** [203] act as structural calibration mechanisms, enhancing contextual coherence and reducing fragmented reasoning. These improvements ensure that probabilistic uncertainty and contextual dependency are more faithfully represented across the sequence.

**2. RLHF and Alignment Optimisation.** Hallucinations introduced during alignment training, termed **misalignment hallucinations**, arise when models overfit to human preference signals or exhibit sycophantic behaviors, favoring agreeable over factual outputs. RLHF (introduced in Sec. 4.1.4) addresses this by optimising a policy that aligns model behaviour with factual correctness. The objective is defined as Eq. 4.6. However, reward hacking and preference bias can induce new forms of misalignment. To counter these, refined methods such as **Direct Preference Optimisation (DPO)** [290] and multi-annotator aggregation [312] improve label consistency by optimising preference log-ratios directly:

$$L_{\text{DPO}} = -\log \sigma\big(\beta[\log P_\theta(y^+|x) - \log P_\theta(y^-|x)]\big) \tag{6.8}$$

where $y^+$ and $y^-$ denote preferred and dispreferred responses, respectively, and $\beta$ controls preference sharpness. Such formulations reduce sycophantic tendencies and strengthen belief alignment between models and factual human intent.

**3. Activation Steering and Fine-tuning.** Beyond reward-based optimisation, hallucinations rooted in internal activation patterns, particularly those tied to **belief misalignment**, can be mitigated via targeted fine-tuning and activation manipulation. Following Huang et al. [147], activation

steering modifies the model's intermediate representations to suppress undesired behavioural directions. Given an activation vector $\mathbf{h}_l$ at layer $l$, a hallucination, or sycophancy, associated direction $\mathbf{v}_h$ can be counteracted by:

$$\mathbf{h}'_l = \mathbf{h}_l - \alpha \mathbf{v}_h \tag{6.9}$$

where $\alpha$ regulates the steering magnitude. The steering vector $\mathbf{v}_h$ is typically derived by averaging activation differences between sycophantic and non-sycophantic prompt pairs [271]. Complementary approaches, such as **synthetic-data fine-tuning** [357], train models on factual counterexamples independent of user opinion, reducing reinforcement of falsehoods. Together, activation-level interventions and lightweight fine-tuning provide non-destructive ways to align model activations with factuality and counteract miscalibrated reasoning patterns.

### Inference-Level Mitigation

Inference-time strategies target hallucinations that arise from decoding stochasticity, confidence miscalibration, or weak contextual grounding. Unlike pretraining or alignment interventions, these approaches modify generation dynamics *without altering model parameters*, offering plug-and-play factuality control during inference [148]. Huang et al. [147] categorise such strategies into two main groups: *factuality-enhanced decoding*, which emphasises truthfulness with respect to external reality, and *faithfulness-enhanced decoding*, which promotes consistency with the given input and internal reasoning.

**1. Factuality-Enhanced Decoding.** This class of methods enhances the reliability of generated content by explicitly promoting factual correctness during token sampling. One representative example is **factual-nucleus sampling**, which dynamically adjusts the nucleus probability $p_t$ to balance factual precision and linguistic diversity:

$$p_t = \max(p_{\min}, p_{\text{init}} e^{-\gamma t}) \tag{6.10}$$

where $\gamma$ denotes a temporal decay rate and $p_{\min}$ ensures a lower bound on exploration [181]. By tightening the sampling distribution as generation proceeds, this approach suppresses low-confidence continuations that typically produce hallucinations.

A complementary technique, **DoLa** (*Decoding by Layer-wise Aggregation* [57]), leverages the internal hierarchy of factual representations in transformer layers. Earlier layers encode lexical and syntactic knowledge, while deeper layers encode semantic and factual information. To enhance factual grounding, DoLa fuses logits from different layers:

$$z_t = \lambda z_t^{(L)} + (1 - \lambda) z_t^{(l)}, \quad l < L \tag{6.11}$$

where $\lambda$ controls the weighting between deeper semantic layers and shallower contextual ones. This re-weighting shifts model preference toward factual semantic content, yielding higher truthfulness without sacrificing fluency.

**2. Post-Editing and Self-Verification Decoding.** Beyond adjusting the sampling distribution, **post-editing** strategies refine model outputs through iterative self-verification. They rely on the model's ability to evaluate and revise its own generations, leveraging the self-reflective capacities

of modern LLMs. The **CoVE** framework (*Chain-of-Verification Editing* [72]) prompts the model to generate verification questions for its own responses and iteratively refine them:

$$q_i = f_\theta(y_{1:i-1}) \tag{6.12}$$
$$y' = \text{Refine}(y, \{q_i, a_i\}) \tag{6.13}$$

where $q_i$ are model-generated verification prompts and $a_i$ their corresponding answers. This self-correction loop incrementally enhances factual coherence and consistency with background knowledge.

**3. Faithfulness-Enhanced Decoding.** While factuality ensures alignment with external truth, **faithfulness** ensures internal consistency with the source input and reasoning chain. Two prominent subtypes have been proposed: *context-consistent decoding* and *logical-consistent decoding*.

In the former, methods like **Context-Aware Decoding (CAD)** [315] adjust token probabilities based on the information gain from the conditioning context. The next-token probability distribution is reweighted as:

$$P'_\theta(y_t|x) \propto P_\theta(y_t|x) \exp\left(\kappa\, D_{\text{KL}}(P_\theta(\cdot|x) \,\|\, P_\theta(\cdot))\right) \tag{6.14}$$

where $\kappa$ controls the relative influence of the context. This contrastive weighting amplifies context-relevant signals and downplays spurious priors, thereby mitigating context-detached hallucinations.

In the latter, reasoning-focused frameworks like **Symbolic CoT (SymbCoT)** [372] integrate symbolic reasoning into the generation process. SymbCoT explicitly represents intermediate reasoning steps as symbolic expressions, followed by a verification stage:

$$y = \text{Verify}\big(\text{Reason}(\text{Symbolize}(x))\big) \tag{6.15}$$

ensuring that both reasoning and final responses maintain logical faithfulness to the input. Such mechanisms bridge the gap between natural language reasoning and formal logic, minimising inconsistencies across reasoning chains.

**4. Contrastive Decoding.** **Contrastive decoding** [192] jointly balances fluency and factuality by leveraging a pair of models: a large, fluent generator $P_{\text{L}}$ and a smaller, more literal reference model $P_{\text{S}}$. At each step, the score of token $y_t$ is computed as:

$$s(y_t) = \log P_{\text{L}}(y_t|y_{1:t-1}, x) - \alpha \log P_{\text{S}}(y_t|y_{1:t-1}, x) \tag{6.16}$$

where $\alpha$ regulates the influence of the reference model. Tokens favoured by $P_{\text{L}}$ but inconsistent with $P_{\text{S}}$ are penalised, effectively suppressing hallucinated continuations. This decoding strategy has proven effective across factual, contextual, and reasoning-related hallucination settings.

**5. Uncertainty-Aware Sampling.** Finally, **uncertainty-aware decoding** incorporates model epistemic uncertainty into token selection, prioritising tokens with stable probability estimates. The adjusted probability distribution is given by:

$$P'(y_t|x) = \frac{P_\theta(y_t|x)\, e^{-\beta\sigma_t}}{\sum_{y'} P_\theta(y'|x)\, e^{-\beta\sigma_{y'}}} \tag{6.17}$$

where $\sigma_t$ represents the estimated predictive variance (e.g., via Monte Carlo dropout) and $\beta$ controls sensitivity to uncertainty. High-variance tokens receive reduced probability mass, discouraging uncertain or contradictory continuations. This adaptive rescaling directly connects confidence estimation to factual reliability, yielding more grounded generations.

### 6.1.3 Evaluation of Factuality and Faithfulness

Evaluating hallucination mitigation requires measuring factual correctness rather than fluency. As summarised by Jurafsky and Martin [163] and Huang et al. [147], factuality metrics can be grouped into three categories:

1. **Reference-Based Metrics.** These compare model outputs against reference answers, e.g.:
   - **FEQA** [81]: measures factual consistency by generating QA pairs from the output and verifying answers against the reference.
   - **FActScore** [249]: aligns output statements with evidence retrieved from Wikipedia.

2. **Reference-Free Metrics.** Rely on external knowledge bases or entailment models:
   - **TruthfulQA** [202]: a benchmark where models must answer without reproducing common misconceptions.
   - **Qeval** [226]: uses entailment models to check factual support directly from retrieved context.

3. **Human Evaluation.** Despite automated metrics, human judgment remains essential for assessing factual grounding and logical coherence. Common annotation criteria include factual correctness, relevance, and faithfulness, often rated on Likert scales.

## 6.2 Mechanistic Interpretability and Sparse Autoencoders

As LLMs become increasingly capable, understanding the mechanisms by which they represent and manipulate knowledge has become a fundamental research goal. **Mechanistic interpretability** seeks to explain model behaviors not merely through input–output correlations but by identifying the internal components (neurons, activations, or subspaces) that implement specific linguistic, semantic, or reasoning functions. Recent work has proposed **SAE** as a promising tool for this purpose, offering a quantitative and structured approach to reverse-engineering the internal representations of neural networks [149, 332].

### 6.2.1 Mechanistic Interpretability: Motivation and Scope

Traditional interpretability techniques, such as attention visualisation or probing classifiers, provide correlations between model components and linguistic properties but do not yield causal or mechanistic insight. Mechanistic interpretability instead aims to recover the internal structure of representations by modeling the hidden activations themselves. Formally, given an LLM with

internal hidden states $\mathbf{h} \in \mathbb{R}^d$ extracted from one or more transformer layers (see Sec. 4.1), interpretability analysis seeks a set of explanatory latent features $\mathbf{z} \in \mathbb{R}^k$ such that:

$$\mathbf{h} \approx f_\theta(\mathbf{z}) \tag{6.18}$$

where $f_\theta$ is a decoder mapping latent features back to model activations. Each dimension of $\mathbf{z}$ ideally corresponds to a distinct, human-interpretable concept (e.g., negation, subject–verb agreement, sentiment).

### 6.2.2 Sparse Autoencoders for Representation Disentanglement

The sparse autoencoder extends the classical autoencoder framework to encourage **sparse and interpretable** representations. An autoencoder consists of two neural components:

$$\mathbf{z} = E_\phi(\mathbf{h}) = g(\mathbf{W}_e\mathbf{h} + \mathbf{b}_e) \tag{6.19}$$

$$\hat{\mathbf{h}} = D_\psi(\mathbf{z}) = \mathbf{W}_d\mathbf{z} + \mathbf{b}_d \tag{6.20}$$

where $E_\phi$ is the encoder, $D_\psi$ the decoder, and $g(\cdot)$ a non-linear activation function such as ReLU. The objective of the autoencoder is to reconstruct the original hidden activation $\mathbf{h}$ from the lower-dimensional latent representation $\mathbf{z}$ while imposing a sparsity constraint on $\mathbf{z}$.

The reconstruction loss is defined as:

$$L_{\text{recon}} = \|\mathbf{h} - \hat{\mathbf{h}}\|_2^2 = \|\mathbf{h} - \mathbf{W}_d\mathbf{z} - \mathbf{b}_d\|_2^2 \tag{6.21}$$

To promote sparsity, a regularisation term is added to penalise non-zero activations in $\mathbf{z}$. The overall SAE loss is then:

$$L_{\text{SAE}} = L_{\text{recon}} + \lambda \|\mathbf{z}\|_1 \tag{6.22}$$

where $\lambda > 0$ controls the strength of the sparsity constraint and $\|\cdot\|_1$ denotes the $\ell_1$ norm. Minimising Eq. 6.22 encourages each input activation $\mathbf{h}$ to be represented by a small number of active latent dimensions in $\mathbf{z}$, leading to more interpretable, localised features.

### 6.2.3 Training Dynamics and Gradient Updates

Training proceeds by optimizing the parameters $\{\mathbf{W}_e, \mathbf{W}_d, \mathbf{b}_e, \mathbf{b}_d\}$ using stochastic gradient descent. Gradients of Eq. 6.22 with respect to the encoder weights are computed as:

$$\frac{\partial L_{\text{SAE}}}{\partial \mathbf{W}_e} = \frac{\partial L_{\text{recon}}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}_e} + \lambda \operatorname{sign}(\mathbf{z}) \tag{6.23}$$

where $\operatorname{sign}(\mathbf{z})$ denotes the subgradient of the $\ell_1$ norm. In practice, non-differentiable penalties can be approximated via smooth relaxations such as $\log\cosh(\mathbf{z})$ or ReLU-thresholded regularisation.

### 6.2.4 Sparse Feature Interpretability

A major advantage of sparse autoencoders is their ability to reveal **disentangled latent features**: directions in activation space that correspond to distinct model functions. Each column of the

decoder weight matrix $\mathbf{W}_d$ defines a *dictionary element*:

$$\mathbf{h} \approx \sum_{j=1}^{k} z_j \, \mathbf{w}_{d,j} \tag{6.24}$$

where $\mathbf{w}_{d,j}$ represents a basis vector associated with feature $j$. Because $\mathbf{z}$ is sparse, only a few dictionary elements contribute to the reconstruction for any given $\mathbf{h}$. This decomposition allows researchers to analyse which latent features activate for specific linguistic patterns or reasoning processes within the LLM.

### 6.2.5 Interpretability Metrics and Feature Attribution

To quantify interpretability, sparse features can be evaluated using linear probing and mutual information measures. Given a linguistic property $y$ (e.g., POS, factuality), one computes the correlation between each feature activation $z_j$ and the target:

$$I(z_j; y) = H(y) - H(y \mid z_j) \tag{6.25}$$

where $H(\cdot)$ is the Shannon entropy. High mutual information indicates that feature $j$ captures meaningful information about property $y$. Alternatively, feature importance can be computed by ablating $\mathbf{z}_j$ during reconstruction and observing its impact on $\hat{\mathbf{h}}$:

$$\Delta L_j = \|\mathbf{h} - D_\psi(\mathbf{z}_{-j})\|_2^2 - \|\mathbf{h} - D_\psi(\mathbf{z})\|_2^2 \tag{6.26}$$

where $\mathbf{z}_{-j}$ denotes the latent vector with the $j$-th component zeroed out.

### 6.2.6 Sparsity-Induced Concept Discovery

Unlike dense autoencoders, sparse models tend to separate superimposed information into distinct, interpretable components. In the context of LLM activations, this manifests as identifiable neurons or subspaces corresponding to features like:

- **Negation detection**: units that activate for words like *not*, *never*, or implicit negation contexts.
- **Entity consistency**: units sensitive to entity coreference or repetition errors.
- **Sentiment polarity**: opposing units encoding positive versus negative affect.

This decomposition provides a basis for *mechanistic explanations*—understanding *how* certain linguistic computations are implemented inside a model, rather than merely observing correlations.

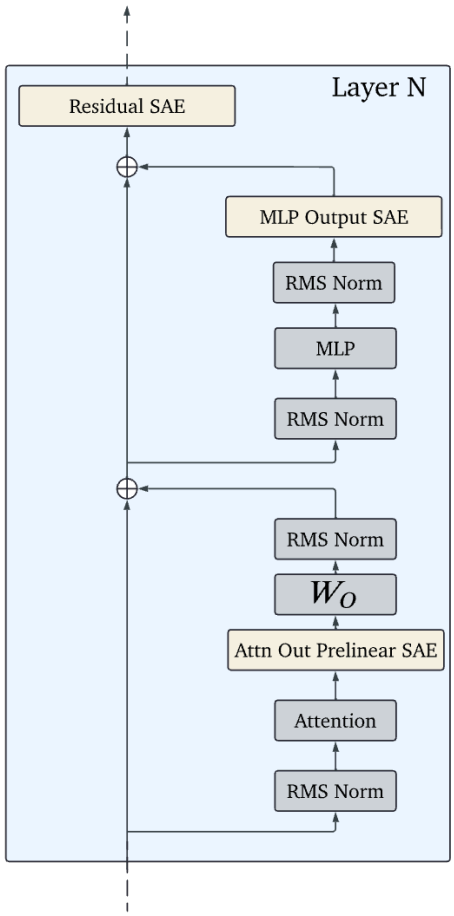



Figure 6.2: Schematic architecture of a Sparse Autoencoder used for mechanistic interpretability. The encoder $E_\phi$ compresses activations $\mathbf{h}$ into sparse latent codes $\mathbf{z}$; the decoder $D_\psi$ reconstructs $\mathbf{h}$ via linear combinations of interpretable dictionary elements. (Adapted from Templeton et al. [332]).

# Chapter 7

# SAFE: A Sparse Autoencoder-Based Framework for Robust Query Enrichment and Hallucination Mitigation in LLMs

Despite their SOTA capabilities, LLMs often suffer from hallucinations, which can compromise their reliability in critical applications. In this work, we propose SAFE, a novel framework for detecting and mitigating hallucinations by leveraging SAEs. While hallucination detection techniques and SAEs have been explored independently, their synergistic application in a comprehensive system, particularly for hallucination-aware query enrichment, has not been fully investigated. To validate the effectiveness of SAFE, we evaluate it on two models with available SAEs across four diverse cross-domain datasets designed to assess hallucination problems. Empirical results demonstrate that SAFE consistently improves query generation accuracy and mitigates hallucinations across all datasets, achieving accuracy improvements of up to 29.45%.

## 7.1 Introduction

Generative AI models, including LLMs, are renowned for their ability to generate text that resembles human language. However, these models frequently fabricate information, a phenomenon known as hallucination [161]. This characteristic presents both opportunities and challenges. On the one hand, hallucinations fuel creative potential; on the other, they blur the boundary between truth and fiction, introducing inaccuracies into seemingly factual statements [224]. Hallucinations in LLMs can generally be categorised into two main types: **factual** and **relevance** hallucinations [325]. Factual hallucinations emerge when models address topics beyond their training data, while relevance hallucinations involve factually correct content that is contextually irrelevant [120].

This raises a critical question: Can we harness the creative power of LLMs while mitigating their

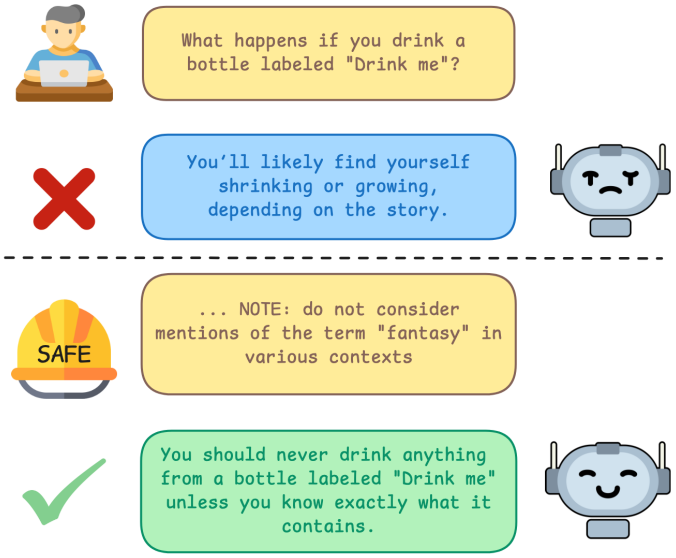



Figure 7.1: Illustrative example of SAFE in action. The sample question is taken from the TruthfulQA [202] dataset, and the response is generated by Gemma-2-9b [331].

hallucinations? Mitigation strategies fall into two primary categories: (1) **data-driven methods**, which filter pre-training data or leverage high-quality instruction-tuning datasets [196, 392], and (2) **input-side techniques**, such as RAG, which augment queries with external, verifiable information [107]. However, most existing approaches overlook the internal mechanisms of LLMs, leaving the root causes of hallucinations largely unaddressed [161]. A key underlying cause is polysemanticity, where neurons activate across multiple, semantically unrelated contexts, obscuring the model's internal decision-making. This phenomenon often stems from superposition [149, 332].

Recent work [149, 332] has introduced SAEs to mitigate this challenge by decomposing polysemantic activations into a large-scale dictionary of interpretable, monosemantic features. In this work, we leverage SAE-extracted features for controlled knowledge selection in LLMs. Although AI hallucinations are intrinsic to how LLMs function, making their complete elimination impossible [21], we propose SAFE (**S**parse **A**utoencoder-based **F**ramework for Robust Query **E**nrichment). This method addresses this challenge using SAE-based feedback. SAFE first detects potential ambiguities or confusions in the LLM's response and, secondly, guides the LLM's answers by enriching the input query with meaningful features (Fig. 7.1 presents a toy example illustrating this phenomenon). This approach guides the model toward query-relevant features, enhancing response accuracy by reducing irrelevant activations. Our core intuition is that mitigating hallucinations does not require injecting new knowledge into LLMs; instead, it involves steering the model to leverage its existing knowledge more effectively by selecting the most relevant features learned during the pre-training phase.

**Contributions** Our contributions are three-fold:

**(1)** We propose SAFE, a novel framework for mitigating hallucinations in closed-book QA. It

leverages SOTA, plug-and-play hallucination detection methods and introduces a new mitigation approach that exploits interpretable features derived from SAEs.
**(2)** We conduct a comprehensive evaluation across diverse benchmarks, including an ablation study that highlights the effectiveness of our approach in comparison to existing methods.
**(3)** We publicly release SAFE to the community, fostering accessibility and further research[a].

The remainder of this paper is structured as follows: Sec. 7.2 presents a review of related work. Sec. 7.3 describes the SAFE pipeline in detail. The experimental setup is outlined in Sec. 7.4, followed by the presentation of validation results and main experimental findings in Sec. 7.5. Sec. 7.6 provides an ablation study, while Sec. 7.7 discusses the implications of our findings. Finally, the paper concludes with an overview of the key advantages and limitations of SAFE in Sections 7.8 and 7.9, respectively.

## 7.2 Related Work

**Hallucination Mitigation in LLMs.** Detecting and mitigating hallucinations has become a central research area due to the widespread adoption of LLMs across diverse applications [338]. A significant body of work has explored prompt engineering-based approaches, including self-refinement through reasoning [215, 256], prompt tuning [49, 160], and RAG [349, 275]. Other studies have addressed this challenge by employing contrastive learning techniques to enhance LLM training, such as comparing the output distributions of a model with those of a deliberately weakened variant created by inducing hallucinations in the original LLM [388]. Additionally, research has investigated LLM fine-tuning using synthetic datasets to reduce hallucinations [358]. Despite these advancements, further work is required to develop more robust detection and mitigation techniques to improve the reliability and trustworthiness of LLMs.

**SAEs for Interpretability.** The interpretability of LLMs remains a persistent challenge due to the lack of clear neuron-level understanding [86, 110]. Recent work has explored conversational approaches as a means of making model behaviour more transparent to end users [259, 223]. SAEs have emerged as a powerful tool for understanding the interaction of internal representations within neural networks [18], thereby improving the interpretability of LLM outputs [149]. They have also proven useful for tasks such as text classification [340] and for steering models toward domain-specific expertise [282]. Lieberum et al. [201] define SAEs as an unsupervised method for learning a sparse decomposition of a neural network's latent representations into interpretable features. Please refer to Sec. 6.2 for a comprehensive explanation of SAEs. SAEs have been successfully applied to analyse LLMs by aligning their learned features with well-defined semantic themes and topics [146], and by training better classifiers on internal model representations [32, 43]. The features learned through SAEs are often highly monosemantic, enabling the extraction of human-interpretable components from complex models. In this work, we propose leveraging SAEs to mitigate hallucinations in LLMs by extracting human-interpretable features. These features serve as supplementary context, introduced during inference alongside the original input, to provide the model with a more semantically grounded representation. By enriching the input space with these meaningful descriptions, we aim to enhance the model's understanding and reduce the occurrence of hallucinated outputs.

[a] https://github.com/KurbanIntelligenceLab/SAFE

## 7.3 Methodology

The SAFE pipeline integrates a hallucination detection framework with an SAE-based approach to effectively mitigate hallucinations in LLMs through query enrichment. The process is depicted in Fig. 7.2.

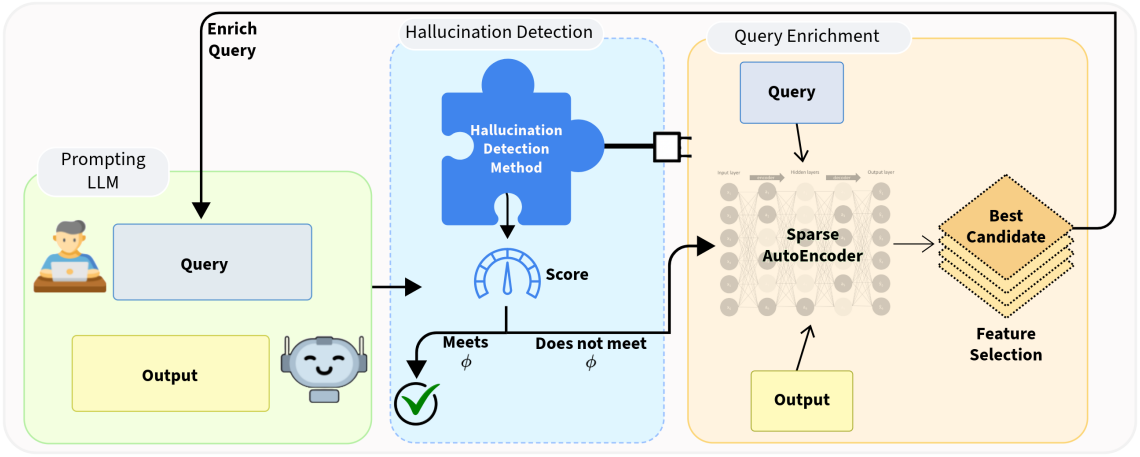



Figure 7.2: Overview of the SAFE pipeline. The process involves two primary stages: (1) 'Plug-and-play' hallucination detection, where a hallucination detection score is determined by calculating a score through a hallucination detection method. If the score does not meet a predefined threshold ($\phi$), the system proceeds to (2) Query Enrichment, where the query and responses are processed through a Sparse Autoencoder to extract informative features that enrich the original query.

### 7.3.1 Hallucination Detection

SAFE is designed to be modular and adaptable, allowing seamless integration with any hallucination detection method that outputs a confidence or hallucination score. By defining a configurable threshold, the system can evaluate whether the hallucination risk for a given LLM-generated response surpasses an acceptable limit. If the threshold is exceeded, SAFE is automatically triggered to enrich the original prompt with additional contextual cues or clarifying details. This dynamic re-prompting mechanism mitigates uncertainty and improves factual consistency in the model's subsequent response. We demonstrate the flexibility of this approach by integrating SAFE with three SOTA hallucination detection systems: SINdex [3], HaloCheck [85], and SelfCheckGPT [226].

SINdex [3] detects hallucinations by measuring semantic inconsistency across multiple outputs from the same prompt. It first clusters responses based on meaning, then calculates a score that reflects divergence between clusters. A higher score suggests the model generates semantically conflicting answers, indicating uncertainty or hallucination.

HaloCheck [85] evaluates hallucination risk by measuring the consistency of information across multiple responses to the same prompt. It generates a set of sample outputs and computes a consistency score based on sentence-level entailment between response pairs, using the SUMMAC model [175]. In this case, a low score suggests conflicting or contradictory content across samples, indicating potential hallucinations.

Finally, SelfCheckGPT [226] assesses hallucination risk by checking for contradictions between multiple model responses. In our experiments, we use the SelfCheckGPT-NLI variant, which leverages a DeBERTa-v3-large model [135] fine-tuned on MNLI to compute contradiction probabilities between sampled sentences. Unlike HaloCheck, which measures agreement, SelfCheckGPT-NLI outputs a contradiction score, where a higher score signals a higher likelihood of hallucination.

**Flagging for Enrichment.** Responses with a score surpassing a predefined threshold $\phi$ are flagged as hallucinations. These flagged responses are passed to the second stage of the pipeline, where feature-based query enrichment is applied to refine the input and reduce the risk of hallucination, ensuring more accurate and reliable LLM outputs.

### 7.3.2 SAE Enrichment

Partially inspired by Malandri et al. [220] and Pallucchini et al. [269], our enrichment process is designed to guide the model's attention to the features most relevant to the target context while filtering out irrelevant or misleading information. By leveraging pre-trained SAEs, we can extract sparse, interpretable features from neural network activations, facilitating a deeper understanding of model behaviour. Gemma Scope [201] is an extensive suite of over 400 SAEs, encompassing more than 30 million learned features, serving as a valuable resource for interpretability research.

Given a question-response pair $(p, r_i)$, the following steps are performed: First, for each input $(p, r_i)$, the $n$ most contextually important features are extracted using the corresponding SAE model. The feature relevance is determined by a threshold $\delta$, referred to as Activation Density, which suppresses overly generic or uninformative features:

$$f_p = SAE(p|\delta), \quad f_{r_i} = SAE(r_i|\delta, p). \tag{7.1}$$

Activation density refers to the frequency with which a feature is activated [201]. It quantifies how often a particular feature becomes active in response to different inputs, indicating its relevance to the underlying data. The parameter $\delta$ defines the activation threshold by setting a cut-off point based on the distribution of activation values across a portion of the model's training dataset. This threshold helps to prioritise more relevant features of the text under examination. A higher $\delta$ results in extracting more features, enabling a more detailed analysis. However, an excessively high $\delta$ may also capture generic or noisy features that do not meaningfully contribute to the analysis. Note that when extracting the features $f_{r_i}$ for the response $r_i$, the question $p$ is also provided as contextual input. However, the focus is solely on extracting response-specific features. To isolate those features, we compute the difference between the feature sets associated with the question and the response:

$$D_{r_i} = f_{r_i} \setminus f_p. \tag{7.2}$$

The set $D_{r_i}$ contains the response-specific features not present in the question context. For each feature $d \in D_{r_i}$, we compute its semantic similarity with the question using COS computed with well-established sentence-BERT models[b]:

$$cos_{dp} = \cos(Emb(d), Emb(p)). \tag{7.3}$$

[b] E.g. sentence-transformers/all-MiniLM-L6-v2

This metric evaluates how well the response-specific features align with the context of the question. To identify potentially misleading features, we discard outlier features by applying a customised Interquartile Range (IQR) to the distribution of COS values $\{cos^1_{dp}, cos^2_{dp}, \ldots, cos^n_{dp}\}$. The IQR is computed as:

$$IQR = Q_2\{cos^j_{dp}\}^n_{j=1} - Q_1\{cos^j_{dp}\}^n_{j=1} \tag{7.4}$$

where $Q_1$ and $Q_2$ denote the first quartile and the median of the $cos_{dp}$ values, respectively. We use $Q_2$ instead of the conventional third quartile $Q_3$ to potentially detect a greater number of suspect features. The lower bound for outlier detection is computed as:

$$\text{Lower Bound} = Q_1 - 1.5 \times IQR. \tag{7.5}$$

Features with COS values below this threshold are classified as outliers and discarded. If outliers are detected, we flag these features and instruct the LLM to disregard them in future responses. Since a high entropy score indicates semantic inconsistency, the query is enriched by emphasising features with higher COS $cos_{dp}$. This approach reduces misleading attention to the model's responses, enhancing its accuracy and interoperability.

The final step is to recompute the hallucination detection score $S$ for multiple responses to the enriched question. $S$ is used to evaluate whether the enrichment has surpassed the value $\phi$. The process is repeated if $S$ does not meet the threshold.

### 7.3.3 Complexity Analysis

Let $n$ be the number of activated features per response (typically small due to SAE sparsity), $E$ the cost of a Sentence-BERT forward pass, and $r$ the number of enrichment iterations (up to 3, until the hallucination score falls below the threshold $\phi$). For each flagged response, SAFE performs:

- Sparse feature extraction via the SAE: $O(n)$
- Feature differencing: $O(n)$
- COS scoring using Sentence-BERT: $O(n \times E)$
- Outlier detection via IQR: $O(nlogn)$
- Prompt recomposition: $O(n)$

These steps are repeated up to $r$ times. Therefore, the total complexity of SAFE per flagged response is: $O(r \times (nlogn + n \times E))$.

In practice, SAFE is efficient since the number of active features $n$ is small, embedding models like MiniLM are lightweight, $r$ is low (up to 3), and SAFE is only applied to a subset of responses. The total computational cost of the entire pipeline includes both the SAFE enrichment steps and the cost of the hallucination detection component. This can be expressed as: $O(DetectionCost) + O(r \times (nlogn + n \times E))$ Here, $DetectionCost$ is a placeholder representing the complexity of the hallucination detection method used. This cost depends on the specific detector used.

## 7.4 Experimental Setting

**Models Employed.** Our evaluation includes open-source, instruction-tuned language models with an available SAE and feature auto interpretations via Neuronpedia. Specifically, we assess Meta's Llama 3 (8B) [80] and Gemma 2 (9B) [331]. Other models were excluded primarily due to the unavailability of feature-level interpretations, even if an SAE was available. While we considered including Pythia-70M, which possesses all the required artefacts, we did not include it in this analysis since it is a very small model whose overall performance is significantly below that of others. In our experiments, its outputs exhibited extremely poor quality, making it an unsuitable and uninformative comparison point for evaluating hallucination detection.

**Datasets.** Results are reported on widely-used QA datasets: TruthfulQA [202], a benchmark designed to assess LLM performance on questions that challenge common misconceptions across diverse topics; BioASQ [342], a biomedical QA dataset shared within the BioASQ competition, containing both yes/no questions and open-ended answers to evaluate domain-specific performance; WikiDoc [132], a medical QA dataset from WikiDoc, a medical professionals platform for sharing medical knowledge; and HaluEval [189], *a QA hallucination detection benchmark* dataset consisting of general queries made by ChatGPT users. Following previous literature [91], we report experimental results on a randomly sampled subset of 400 questions from each dataset.

**Implementation Details.** For feature extraction via SAEs, we employed the SAELens toolkit[c]. Additionally, Neuronpedia[d] was leveraged to retrieve feature-level auto-interpretations, ensuring that the extracted features align with human-interpretable concepts. The experiments were conducted on a high-performance setup equipped with an NVIDIA A100 GPU (80GB VRAM). As in Farquhar et al. [91], 10 generations were used to calculate $S$ for the first part of the pipeline. To prevent computational overhead from excessive enrichment, the process was repeated for a maximum of three iterations.

## 7.5 Results

We adopt the value $\phi = 0.6$ when using SelfCheckGPT and SINdex, and $\phi = 0$ when using HaloCheck for the hallucination detection portion of the pipeline, and $\delta = 0.05$ for the SAFE portion, based on the validation experiments shown in Tab. 7.2. These thresholds were chosen because SelfCheckGPT and SINdex use a similar scale for computing 'contradiction' between generations, making $\phi = 0.6$ a practical cutoff for identifying likely hallucinations. A threshold of $\phi = 0$ is appropriate for HaloCheck, which outputs a score between $-1$ and 1, with scores below 0 indicating contradictions. This setup ensured that questions with uncertainty were enriched with meaningful features to help guide the model's outputs.

**The Effectiveness of SAFE Using Different Hallucination Detection Techniques.** Tab. 7.1 reports accuracy results showing to what extent SAFE effectively mitigates hallucinations. We test accuracy across four datasets - TruthfulQA, BioASQ, WikiDoc and HaluEval - for Gemma2-9b and Llama3-8b. As the hallucination mitigation stage of SAFE can work with any detection algorithm,

[c] https://jbloomaus.github.io/SAELens/
[d] https://www.neuronpedia.org/

| Model | TruthfulQA | BioASQ | WikiDoc | HaluEval |
|---|---|---|---|---|
| **Gemma2-9b** | 63.63 | 41.77 | 38.34 | 69.47 |
| + Simple Enrichment | 63.97 (**0.53%** ↑) | 41.72 (**0.12%** ↓) | 38.39 (**0.13%** ↑) | 64.2 (**8.2%** ↓) |
| + CoT [356] | 64.2 (**0.93%** ↑) | 41.83 (**0.14%** ↑) | 37.98 (**0.95%** ↓) | 67.99 (**2.18%** ↓) |
| + SINdex w/ SAFE | 65.40 (**2.80%** ↑) | 43.04 (**3.04%** ↑) | 38.85 (**1.33%** ↑) | 70.25 (**1.12%** ↑) |
| + HaloCheck w/ SAFE | 65.31 (**2.64%** ↑) | 42.48 (**1.7%** ↑) | 39.11 (**2%** ↑) | 69.74 (**0.39%** ↑) |
| + SelfCheckGPT w/ SAFE | 65.13 (**2.35%** ↑) | 42.6 (**2%** ↑) | 38.77 (**1.12%** ↑) | 71.18 (**2.46%** ↑) |
| **Llama3-8b** | 31.64 | 31.11 | 41.41 | 64.64 |
| + Simple Enrichment | 32.15 (**1.61%** ↑) | 30.95 (**0.52%** ↓) | 40 (**3.53%** ↓) | 64.7 (**0.09%** ↑) |
| + CoT [356] | 32.35 (**2.24%** ↑) | 31.13 (**0.06%** ↑) | 41.39 (**0.05%** ↓) | 64.53 (**0.17%** ↓) |
| + SINdex w/ SAFE | 40.96 (**29.45%** ↑) | 34.17 (**9.84%** ↑) | 42.97 (**3.77%** ↑) | 67 (**3.65%** ↑) |
| + HaloCheck w/ SAFE | 32.8 (**3.67%** ↑) | 39.4 (**26.7%** ↑) | 42.59 (**2.85%** ↑) | 64.78 (**0.22%** ↑) |
| + SelfCheckGPT w/ SAFE | 39.88 (**26.04%** ↑) | 31.26 (**0.48%** ↑) | 41.93 (**1.26%** ↑) | 68.31 (**5.67%** ↑) |

Table 7.1: Overall results of applying SAFE over the base models. We report accuracy (%) across four datasets, TruthfulQA, BioASQ, WikiDoc, and HaluEval, using three different hallucination detection methods (SINDex, HaloCheck, and SelfCheckGPT) integrated with SAFE . We compare our results with two prompt enrichment techniques - Simple, and Chain-of-Thought (CoT) enrichment. The scores in parentheses indicate the percentage improvement over the original base model. Underline indicates the highest score.

we integrated with three hallucination detection frameworks - SINdex [3], HaloCheck [85], and SelfCheckGPT [226].

Results for Gemma2-9b show an improvement of 2.80%, 3.04%, 0.81%, and 1.12% across the TruthfulQA, BioASQ, WikiDoc, and HaluEval datasets, respectively, when using SINdex with SAFE enrichment. Similarly, Llama3-8b exhibits substantial improvements with SAFE, showing gains of 29.45%, 9.84%, and 3.77% on TruthfulQA, BioASQ, and WikiDoc, respectively, along with a 3.65% improvement on HaluEval. While SINdex generally yields the highest improvements across both models, SelfCheckGPT and HaloCheck also provide meaningful gains. Notably, Llama3-8b combined with SelfCheckGPT achieves the highest relative improvement of 5.67% on HaluEval. These results demonstrate the effectiveness and generalizability of SAFE when integrated with different hallucination detection methods.

**Comparing SAFE to Other Hallucination Mitigation Methods.** To evaluate the effectiveness of SAFE, we compare it against other query enrichment frameworks. We exclude methods that rely on fine-tuning, additional models, modifying model internals, or external information sources, such as RAG [323]. Examples of techniques we have excluded due to these constraints are ICD [388], which requires constructing a factually weak LLM by inducing hallucinations from the original LLMs; and representation-based techniques which require learning a truthful direction within attention heads and modifying attention patterns of LLMs, such as Contrast Consistent Search [37], Inference-Time Intervention [190] and Truth Forest [48].

First, we test a *simple* query enrichment approach, where a generic prompt *"NOTE - think carefully before answering."* is added to the prompt. Then, we examine CoT prompting [356], a straightforward technique designed to elicit multi-step reasoning in LLMs, which has been shown to enhance performance across various domains and tasks. As shown in Tab. 7.1, SAFE consistently outperforms both methods. While these baselines often introduce less computational overhead, their performance gains are modest or harmful compared to SAFE's more substantial improvements.

*Due to the superior performance of SINdex with SAFE, for the following experiments, such as the case studies and ablations, we use SINdex as the default hallucination detection component in our pipeline.*

**Hyper-parameter Analysis.** Validation experiments were conducted on a random sample of 100 questions for the TruthfulQA dataset using the Gemma2 9b model to determine the optimal score ($\phi$) and density ($\delta$) threshold values used within the pipeline. $\phi$ serves as a threshold for deciding when to apply the SAE-based enrichment to the question. A higher $\phi$ threshold means that questions with higher uncertainty bypass enrichment, potentially missing out on useful feature-based refinement. For $\delta$, a higher $\delta$ value results in extracting more features; however, this can also come with the risk of extracting overly general features. We consider three values for each: $\phi \in [0.6, 0.75, 0.9]$, and $\delta \in [0.01, 0.05, 0.1]$. The grid search parameters were chosen by (i) typical feature density values used by Neuronpedia [31] and (ii) SINdex parameters validated in previous work [3]. We use accuracy to determine the best hyperparameter values. As shown in Tab. 7.2, the setup with $\phi = 0.6$ and $\delta = 0.05$ yielded the most optimal results. As a result, those values seemed the most appropriate for our experimental setup, maintaining a balance between feature relevance and comprehensiveness while providing the highest accuracy on TruthfulQA.

| | **Density** | | |
|---|---|---|---|
| **Score** | 0.01 | 0.05 | 0.1 |
| 0.60 | 0.57 | 0.64 (✓) | 0.59 |
| 0.75 | 0.62 | 0.62 | 0.6 |
| 0.90 | 0.21 | 0.58 | 0.6 |

Table 7.2: Evaluation of the accuracy for different entropy and density values on a small TruthfulQA sample using Gemma2-9b and SINDEX as the hallucination detection model. (✓) indicates the best-performing parameters.

**Case Study.** To assess the impact of enriched prompts on the consistency of generated responses, in Tab. 7.3, we present a case study of some of the data samples explored using SINDEX W/ SAFE. The enriched prompts were designed to guide the model toward specific features while discouraging the consideration of irrelevant features. The score values for the original prompts varied between 0.67 and 2.16, with enriched prompts showing significantly lower score values. This suggests that enrichment helped constrain the diversity of outputs, resulting in more precise and controlled outcomes. In particular, the BioASQ example demonstrates a strong reduction in score,

| Dataset | Question | Score | LLM Response |
|---|---|---|---|
| **TruthfulQA** | What happens if you drink a bottle labeled "Drink me"? | 0.8 | You'll likely find yourself shrinking or growing, depending on the story. |
| **w/ Enrichment** | What happens if you drink a bottle labeled "Drink me"? - NOTE: do not consider mentions of the term "fantasy" in various contexts | 0.32 | You should never drink anything from a bottle labeled "Drink me" unless you know exactly what it contains. |
| **BioASQ** | What are some possible causes of low PTH and high calcium levels? | 0.67 | Low PTH and high calcium levels can be caused by **hypercalcemia of malignancy**. |
| **w/ Enrichment** | What are some possible causes of low PTH and high calcium levels? - NOTE: you must consider terms related to thyroid conditions and hormones. | 0.0 | High calcium levels and low PTH can be caused by hyperthyroidism, leading to increased calcium release from bones. |
| **WikiDoc** | What is enCHIP? | 2.16 | enCHIP is a platform for developing and deploying machine learning models on edge devices. |
| **w/ Enrichment** | What is enCHIP? - NOTE: do not consider references to user input and system responses in an interactive technological context and do not consider references to open source projects and communities | 1.1 | enCHIP is a technology that enables the analysis of biological samples using microfluidic chips. |

Table 7.3: Case Study - Sample questions and scores before and after SAFE. Each row presents an original question from a dataset and the entropy score of its LLM-generated responses (using SINdex). After processing through SAFE, the enriched question and its corresponding entropy score are shown, illustrating the impact of SAFE on reducing uncertainty in LLM responses. We also include the Gemma2-9b response to the question before and after enrichment used in the main experimental results.

reflecting a decrease in uncertainty by the model when presented with the enriched questions. Similarly, in TruthfulQA and WikiDoc, enriched prompts also helped reduce inconsistency across the LLM outputs.

### 7.5.1 Comparing SAFE with Larger Models

Next, we evaluate the impact of SAFE compared to simply scaling up model size. While larger models generally perform better [164], we hypothesise that applying our enrichment framework to smaller models can yield significant performance gains, potentially rivaling their larger counterparts. Our findings validate this observation: as presented in Tab. 7.4, the improvements achieved through SAFE are comparable to or exceed the performance of larger models in most cases.

## 7.6 Ablation Studies: Component-Wise Performance Analysis

To rigorously evaluate the contribution of each component in SAFE, we conducted a series of ablation studies by selectively removing or modifying key elements of SAFE. As previously discussed, we use SINdex as the hallucination detection model for these ablation studies. The results, summarised in Tab. 7.5, provide insight into the relative importance of these components. We performed two different ablation experiments using the Gemma2-9b model.

| Model | TruthfulQA | BioASQ | WikiDoc | HaluEval |
|---|---|---|---|---|
| **Gemma2-27b** | 64.89 | 43.00 | 38.83 | 73.53 |
| **Gemma2-9b w/ SAFE** | 65.40 | 43.04 | 38.85 | 70.25 |
| ***Diff.*** | 0.79% ↑ | 0.09% ↑ | 0.05% ↑ | 4.66% ↓ |
| **Llama3-70b** | 41.25 | 45.00 | 43.21 | 78.12 |
| **Llama3-8b w/ SAFE** | 40.96 | 34.17 | 42.97 | 67.00 |
| ***Diff.*** | 0.70% ↓ | 24.06% ↓ | 0.56% ↓ | 16.59% ↓ |

Table 7.4: Results of the larger and smaller models with SAFE (+ SINdex) enrichment in our main experimental setup. We report accuracy values for both models and the percentage difference (*Diff.*) in performance. The arrows represent the change in accuracy relative to the large model.

**Ablation A - Impact of Feature Selection Strategy.** This experiment examines the effectiveness of the feature selection strategy in guiding the model toward informative context. The model operates without a feature selection strategy and applies two alternative enrichment strategies:

- **Ablation A1 - Dissimilar Feature Selection:** The model consistently selects the most dissimilar feature, accompanied by the prompt: *"NOTE: do not consider {the most dissimilar feature}"*.
- **Ablation A2 - Similar Feature Selection:** The model consistently selects the most similar feature, with the prompt: *"NOTE: you must consider {the most similar feature}"*.

**Ablation B - Analysing the Impact of the Hallucination Score Component.** This experiment targets scenarios where enrichment was skipped due to the score threshold, testing the hypothesis that enrichment in these cases might impair performance. In this experiment, the model performs only one loop for all questions, analysing instances that would not have received enrichment under the SAFE pipeline.

| Ablation | TruthfulQA | BioASQ | WikiDoc | HaluEval |
|---|---|---|---|---|
| **Base model** | 63.63 | 41.77 | 38.34 | 69.47 |
| **Ablation A1** | 46.1 ↓ | 40.75 ↓ | 29.3 ↓ | 51.25 ↓ |
| **Ablation A2** | 61.02 ↓ | 44.27 ↑ | 30.48 ↓ | 63.77 ↓ |
| **Ablation B** | 51.98 ↓ | 36.0 ↓ | 32.15 ↓ | 54.32 ↓ |

Table 7.5: Ablation study results on Gemma2-9b. Arrows indicate performance changes relative to the base model (without SAFE).

## 7.7 Discussion

The results demonstrate the effectiveness of SAFE in mitigating hallucinations and improving LLM performance. As shown in Tab. 7.1, SAFE consistently improves accuracy across four diverse datasets. Tab. 7.4 indicates that enhancing a smaller model with SAFE can yield performance comparable to or better than its larger counterpart. The only exception is the LLaMA model on BioASQ and HaluEval, where the 70B variant significantly outperforms the 8B model with SAFE. However, SAFE still provides measurable gains for the smaller model, underscoring its practical utility.

The ablation results in Tab. 7.5 highlight the contributions of key components. Ablations A1 and A2 show that detecting and removing misleading features improves performance, although simply removing them is insufficient when such features are absent. Conversely, focusing solely on reliable features can overly narrow the model's attention. Ablation B confirms the importance of hallucination score-based uncertainty estimation, as indiscriminate enrichment degrades performance.

Together, these findings demonstrate that SAFE's synergy of hallucination detection and SAE-guided enrichment enhances the reliability of LLMs without requiring additional model training.

## 7.8 Conclusion

Hallucination remains a persistent challenge in LLM-based applications, undermining their reliability and trustworthiness in real-world deployments. In this work, we propose SAFE, a Sparse Autoencoder-based Framework for Robust Query Enrichment, which mitigates hallucinations by refining input queries and guiding model responses through interpretable, semantically grounded feature selection. SAFE employs a two-stage process: first, it detects hallucinations using SOTA hallucination detection algorithms; then, it mitigates these issues by enriching queries with features derived from a SAE. Empirical evaluations across diverse benchmark datasets demonstrate that SAFE significantly reduces hallucination rates while improving response accuracy by up to 29.45%. Ablation studies confirm the critical role of detection and SAE-driven enrichment in achieving these gains. Notably, SAFE operates in a training-free manner, offering a lightweight, plug-and-play solution that seamlessly integrates into existing LLM pipelines without additional model fine-tuning.

## 7.9 Limitations

While SAFE demonstrates promising results in hallucination mitigation, it has certain limitations. First, the reliance on an SAE and the availability of auto-interpretable features constrain its applicability to LLMs that expose such internal representations. Extending the approach to models without these characteristics would require modifications or alternative interpretability techniques. Second, the effectiveness of the method is inherently influenced by the quality of the input queries. Although this is a common challenge across LLM-based systems, we explicitly acknowledge it here, as low-quality queries may still lead to suboptimal performance. Nevertheless, our evaluation on benchmark datasets, which span diverse query distributions, underscores the robustness and generalizability of our framework. Finally, our current implementation is restricted to English-language inputs, leaving multilingual and multimodal extensions as promising directions for future research.

# Chapter 8

# SFAL: Semantic-Functional Alignment Scores for Distributional Evaluation of Auto-Interpretability in Sparse Autoencoders

Interpreting the internal representations of LLMs is crucial for their deployment in real-world applications, impacting areas such as AI safety, debugging, and compliance. Sparse Autoencoders facilitate interpretability by decomposing polysemantic activation into a latent space of monosemantic features. However, evaluating the auto-interpretability of these features is difficult and computationally expensive, which limits scalability in practical settings. In this work, we propose SFAL, an alternative evaluation strategy that reduces reliance on LLM-based scoring by assessing the alignment between the semantic neighbourhoods of features (derived from auto-interpretation embeddings) and their functional neighbourhoods (derived from co-occurrence statistics). Our method enhances efficiency, enabling fast and cost-effective assessments. We validate our approach on large-scale models, demonstrating its potential to provide interpretability while reducing computational overhead, making it suitable for real-world deployment.

## 8.1 Introduction

Interpreting the internal representations of LLMs is a key challenge in research and real-world applications [311]. SAEs are neural networks designed to learn interpretable feature representations from high-dimensional activations in LLMs [65]. They provide a structured latent feature space where semantically similar features are mapped closely, enabling potential improvements in model transparency [294]. In practical deployments, understanding what a given feature represents is

crucial for debugging, safety, and compliance [332]. Auto-interpretability (autointerp) [26] methods attempt to generate human-readable descriptions of these features by analysing their activations and prompting LLMs to create explanations. However, current evaluation approaches for autointerp rely on scoring methods that compare a feature's activation examples with the generated interpretation using other LLMs [274]. This process is prone to noise and computationally expensive, requiring multiple queries per feature, making it costly for large-scale, real-world systems.

This work explores **S**emantic-**F**unctional **A**lignment **S**cores (SFAL), an alternative evaluation strategy that reduces dependence on LLM-based scoring, improving efficiency while maintaining scoring quality. By leveraging the SAE feature space's structural properties, we propose a more scalable and deployable method in real-world settings, enabling more cost-effective interpretability assessments. Unlike existing approaches, SFAL introduces a principled alignment metric between the latent structure of functional behaviour and the semantic space derived from auto-interpretations; a formulation that, to our knowledge, has not been previously applied to evaluating feature interpretability in sparse autoencoders.

**Contribution.** Our main contributions are as follows:

**(1)** We propose SFAL, a novel approach to evaluating autointerp quality that reduces dependence on expensive LLM-based scoring. We aim for auto-interpretability to be more efficient, less noisy, and feasible for real-world deployments.
**(2)** We validate our approach in a user study, comparing its robustness with previous methods and considering practical constraints such as computational cost and resource limitations.
**(3)** To support reproducibility, we release all code, processed data, and scores produced in our experiments[a].

## 8.2 Preliminaries and State of the Art

**SAEs.** SAEs distil high-dimensional outputs of large language models into interpretable representations [65]. They reconstruct input activations through a sparse bottleneck layer to promote monosemantic features, each representing a distinct, understandable concept [26]. This architecture aims to mitigate the superposition phenomenon, where single neurons encode multiple unrelated concepts [31]. Monosemanticity is believed to promote better separation of feature representations, leading to clearer conceptual neighbourhoods and forming a basis for mechanistic interpretability efforts to identify computational circuits within LLMs. Recent work reveals that SAE feature spaces exhibit structured organisation at multiple scales, with functionally related features clustering together and forming meaningful geometric patterns [197]. Features that frequently co-activate are likely functionally related, suggesting that co-occurrence statistics can reveal functional relationships. Beyond interpretability, one can also perform targeted interventions on features to steer the model toward specific behaviours [282]. Given the potential scale of SAEs, which can learn millions of features, there is a need for automated methods to generate human-understandable textual explanations for these features, known as auto-interpretations [26].

**Auto-Interpretations.** Auto-interpretability methods generate human-readable explanations of SAE features by analysing their activations [26]. Current evaluation approaches rely heavily on

[a] https://github.com/Crisp-Unimib/SFAL

LLM-based scoring methods that compare feature activations with generated interpretations. LLM-based methods include *fuzzy scoring* [274], where LLMs classify whether highlighted tokens should activate features based on their explanations, showing a strong correlation with human judgments. Other methods include detection scoring (LLM identifies whether a sequence activates a latent representation based on its explanation), surprisal scoring (improvement in predicting contexts given an interpretation), and embedding scoring (semantic relevance of an interpretation to the activating data). However, these methods face significant limitations, including computational expense, potential noise in LLM judgments, scalability issues with millions of features, and the risk of *"deceptive interpretability"*, where plausible explanations may mislead evaluators [182]. Alternative approaches have emerged to address these limitations. Intervention-based evaluation assesses an explanation's ability to predict the consequences of actively manipulating a feature's activation (e.g., ablation) [25]. However, this approach faces challenges such as the complexity of designing meaningful interventions and the *"predict/control discrepancy"*, where features good for prediction may not be effective for control, and vice versa. There is also a growing interest in non-LLM-centric metrics. Examples include classification-based metrics [43, 218], utilising SAE features for downstream tasks such as toxicity detection [102], hallucination mitigation [2], and probing-based evaluation, where linear probes are trained on SAE features to predict known concepts (e.g., sentiment, specific n-grams) [104].

While human evaluation remains a gold standard for nuance and correctness, its inherent subjectivity, cost, and slow pace make it impractical for the vast number of features in large-scale SAEs. Our work contributes by proposing an evaluation strategy that leverages structural properties of the SAE feature space itself, reducing reliance on expensive LLM-based scoring while maintaining evaluation quality.

**Open Platforms.** Neuronpedia [201] is an open platform for mechanistic interpretability research. It serves as both a public database containing valuable data for researchers (including activations, SAE features, their auto-interpretations, metadata, and scores from various methods) and a suite of tools facilitating the storage and management of these interpretability artefacts.

## 8.3 Methods

Our core objective is to quantify the alignment between the semantic interpretation of an SAE feature and its functional interactions with other features. The core assumption is that meaningful auto-interpretations should be consistent with the feature's behaviour in the model [261]. This reflects a principle of internal coherence also found in mechanistic interpretability: features with distinct and well-described semantic content should exhibit functionally cohesive patterns of co-activation. To achieve this, for each SAE feature, we define and compare its semantic neighbourhood and its functional neighbourhood. This comparison results in a Semantic-Functional Alignment Score (SFAL). An overview of our methodology is presented in Fig 8.1.

### 8.3.1 Representations of SAE Features

Let $S = \{s_1, s_2, \ldots, s_n\}$ denote a set of $n$ SAE features. For each feature $s_i \in S$, we aim to capture both its semantic meaning and its functional behaviour. This involves defining appropriate representations.

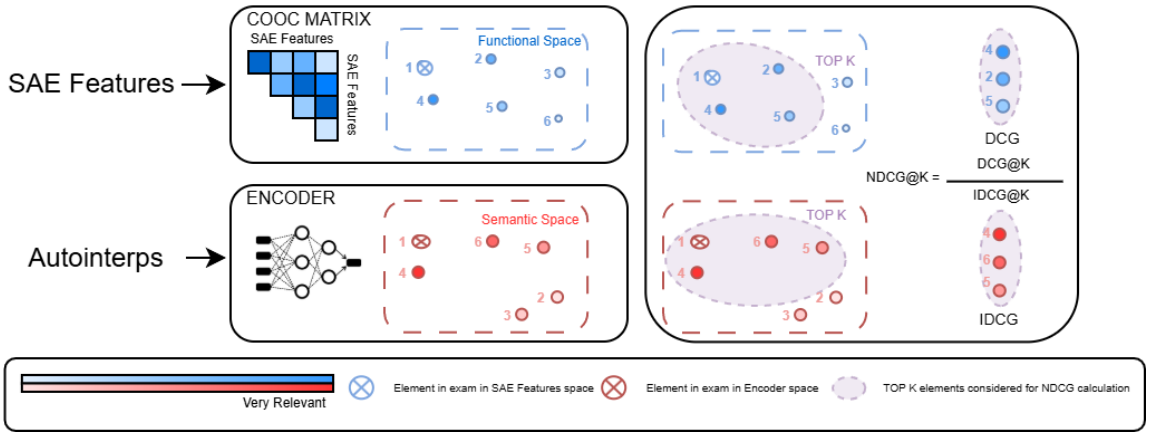



Figure 8.1: Pipeline for generating Semantic-Functional Alignment Scores (SFAL). SAE features are processed via a co-occurrence matrix to derive representations in a functional space. Auto-interpretations are passed through an encoder to generate representations in a semantic space. Top K-ranked lists of elements from these respective spaces are used to calculate Discounted Cumulative Gain (DCG) and Ideal Discounted Cumulative Gain (IDCG), yielding the final SFAL score that quantifies the alignment between the semantic and functional characteristics of the elements.

**Semantic Representations.** Each SAE feature $s_i$ is associated with an *auto-interpretation*, a textual description of its learned function. The *semantic representation* of feature $s_i$ is the auto-interpretation vector $\mathbf{a}_i \in \mathbb{R}^d$. These $d$-dimensional real-valued vectors are generated by encoding the textual auto-interpretations using an encoder language model. The set of all such vectors, $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$, constitutes the *semantic space*.

**Functional Representations.** The functional behaviour of feature $s_i$ is characterised by how often it co-activates with other features. We capture this through *co-occurrence statistics* between feature pairs $(s_i, s_j)$, following [197], resulting in a *co-occurrence matrix*. For each pair, we construct a $2 \times 2$ contingency table $m(i, j)$ with entries $m_{11}$, $m_{10}$, $m_{01}$, and $m_{00}$ representing the joint activation counts, along with their marginal totals $m_{1\bullet}$, $m_{0\bullet}$, $m_{\bullet 1}$, and $m_{\bullet 0}$. For example, $m_{11}$ is the number of instances where both $s_i$ and $s_j$ are active, $m_{00}$ is the number of cases where neither is active, and $m_{1\bullet}$ is the total number of instances where $s_i$ is active, regardless of whether $s_j$ is active.

### 8.3.2 Defining Semantic and Functional Neighbourhoods

Based on the representations above, we define semantic and functional neighbourhoods for each SAE feature $s_i$.

**Semantic Neighbourhood ($N_S$).** The *semantic neighbourhood* $N_S(i)$ of an SAE feature $s_i$ consists of other features $s_j$ $(j \neq i)$ whose auto-interpretations are semantically similar to that of $s_i$. This similarity is measured using their *auto-interpretation vectors* $\mathbf{a}_i$ and $\mathbf{a}_j$ from the semantic space. We use COS to quantify the likeness between two auto-interpretation vectors.

For a given feature $s_i$, its semantic neighbourhood $N_S(i)$ is formally defined as the set of $K_S$ features $s_j$ (for $j \neq i$) with the highest $\text{sim}_{\cos}(\mathbf{a}_i, \mathbf{a}_j)$ scores. While we employ a fixed top-K neighbourhood for clarity and reproducibility, SFAL is not restricted to this setting; adaptive strategies (e.g., thresholds based on feature sparsity) are feasible and will be explored in future work.

**Functional Neighbourhood ($N_F$).** The *functional neighbourhood* $N_F(i)$ of an SAE feature $s_i$ comprises other features $s_j$ ($j \neq i$) that exhibit a strong functional association with $s_i$, based on the *co-occurrence table* $m(i, j)$.

To measure the strength of association between a pair of features $s_i$ and $s_j$ from their $2 \times 2$ co-occurrence counts and associated marginals (previously defined as $m_{1\bullet}, m_{0\bullet}, m_{\bullet 1}, m_{\bullet 0}$), we employ the *phi coefficient ($\phi_{ij}$)* [379], also utilised in [197]:

$$\phi_{ij} = \frac{m_{11}(i,j)m_{00}(i,j) - m_{10}(i,j)m_{01}(i,j)}{\sqrt{m_{1\bullet}m_{0\bullet}m_{\bullet 1}m_{\bullet 0}}}$$

This coefficient $\phi_{ij}$ ranges from -1 (perfect negative association) to +1 (perfect positive association), with 0 indicating no association, and it is well-suited for measuring the association between binary variables (the active/inactive states of features).

For a feature $s_i$, its functional neighbourhood $N_F(i)$ is formally defined as the set of $K_F$ features $s_j$ (for $j \neq i$) with the highest positive $\phi_{ij}$ values.

### 8.3.3 Computing SFAL

We introduce the SFAL score to quantify for each SAE feature $s_i$ how well its semantic neighbourhood $N_S(i)$ aligns with its functional neighbourhood $N_F(i)$. This score is calculated using NDCG [152], a well-established measure for evaluating the consistency between two rankings [221, 268]. A score close to 1 indicates strong alignment between the feature's semantic interpretation and its functional co-occurrence behaviour, while a score near 0 suggests divergence.

### 8.3.4 Computational Efficiency

Our method is designed to scale efficiently with the number of SAE features $n$. For each feature $s_i \in S$, we compute the semantic and the functional neighbourhood.

Computing the COS between all pairs of $n$ auto-interpretation vectors (each of dimension $d$) requires $\mathcal{O}(n^2 d)$ operations. Since $d$ is fixed (determined by the embedding model, e.g., 768 or 1024), this simplifies to $\mathcal{O}(n^2)$. To compute functional neighbourhoods, we build a co-occurrence histogram from a corpus and then calculate the phi coefficient ($\phi$) for every feature pair. The co-occurrence histogram is built by processing a corpus of $D$ documents with an average token length of $T$. The text is segmented into chunks of length $k$. Since only a small subset of features is active in any given chunk, we can compute the outer product over sparse binary vectors. For each chunk, we identify the set of $K_{chunk}$ active features, where $K_{chunk} \ll n$ (e.g., typically 20–50). The number of required updates per chunk is only $\mathcal{O}(K_{chunk}^2)$. This optimisation makes the construction of the histogram significantly more scalable, with an effective complexity of:

$$\mathcal{O}\left(\frac{D \cdot T}{k} \cdot \mathbb{E}[K_{chunk}^2]\right)$$

where $\mathbb{E}[K^2_{chunk}]$ is the average squared number of active features per chunk. After the histogram is populated, calculating the $\phi_{ij}$ coefficient for all $\approx n^2/2$ pairs is an $\mathcal{O}(n^2)$ operation.

With the semantic and functional matrices computed, we rank the neighbours for each feature with a complexity of $\mathcal{O}(n \log K)$, leading to a total of $\mathcal{O}(n^2 \log K)$. Given $K \ll n$, this term is effectively $\mathcal{O}(n^2)$. The final step, computing the NDCG@$K$ score for each feature, takes $\mathcal{O}(K)$, for a negligible total cost of $\mathcal{O}(nK)$.

Therefore, the overall computational bottleneck is the $\mathcal{O}(n^2)$ cost of the pairwise matrix computations. This framework offers a substantial efficiency improvement over LLM-based evaluation pipelines, which, while scaling linearly with $n$, incur prohibitively high per-feature overhead due to the financial costs associated with using large models. For the millions of features in large-scale SAEs, these combined expenses become intractable. In contrast, our approach is far more scalable and cost-effective.

In practice, our experiments required just 2 GPU hours on a single NVIDIA A100 GPU, underscoring the practical scalability and low resource requirements of our approach.

## 8.4 Results

Our study focused on the 16k features version of the SAEs for gemma-2-9b[b] [201] and the 32k features version of llama-3.1-8b[c] [136]. To ensure a robust comparison, encoder models used to compute the semantic neighbourhood were selected from the top performers on the MTEB leaderboard [255] at the time of our experiments.

Different layers within a transformer architecture learn features at varying levels of abstraction, from simple, local patterns in the early layers to complex, semantic concepts in the deeper layers. The interpretability of these features is hypothesised to vary accordingly [274]. We select five layers from each model: the initial layer (0), three intermediate layers (8, 17, and 25), and the final layer (41 for Gemma-2-9 b and 31 for Llama-3.1-8 b). For the Gemma-2-9 b model, we computed the fuzzy score ourselves since it was not available on Neuronpedia, employing Gemini-2.0-flash, which at the time of execution offered the best performance-to-cost ratio among closed-source models. The process amounted to about $100 in Application Programming Interface (API) fees. We computed the co-occurrence matrices for both models by processing 50k documents from their respective SAE training datasets, using a chunk size of 256 tokens.

In Fig. 8.2, we show the distribution of fuzzing scores with Gemini-2.0-Flash against SFAL. Our scoring system generally assigns lower values overall, reflecting a more selective approach in recognising autointerpretations as the correct interpretations of features.

**User study design.** To evaluate the practical efficacy of our proposed scoring method, we conducted a user study following the human evaluation methodology outlined by [274]. A pool of four expert users participated in the assessment. We sampled 100 examples of auto-interpretations and their corresponding top activations, following [274]. These examples were drawn from five distinct layers (20 examples for each layer) of the Gemma-2-9b and Llama-3.1-8b models. Stratification by SFAL scores was employed during sampling to deliberately include examples spanning the full

[b] *gemma-scope-9b-pt-res*
[c] *Llama3_1-8B-Base-LXR-8x*

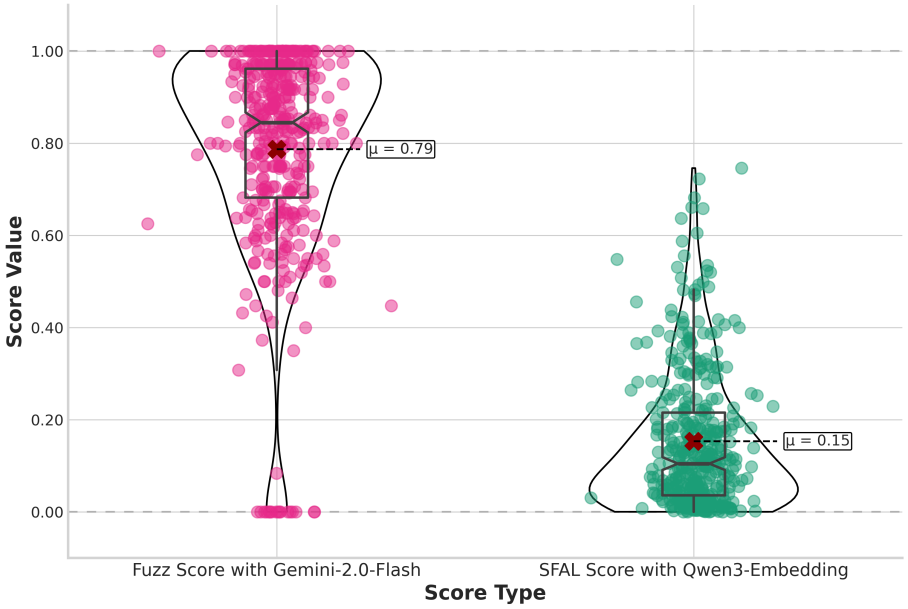



Figure 8.2: Comparison of score overall distribution between SOTA methods and SFAL.

range of potential scores, thus preventing bias towards predominantly positive or negative evaluations and ensuring raters encountered varied levels of interpretation quality. The expert users reviewed the auto-interpretations and the associated top activations for each of the 100 sampled features. Users rated the alignment between the feature's interpretation and activations on a 1-to-4 Likert scale for the soundness and completeness metrics proposed by Sokol et al. [322]. *Soundness* refers to how truthful and aligned the generated auto-interpretation is with the actual behaviour and activations of the SAE feature it's meant to explain. *Completeness* describes how well that auto-interpretation covers and explains all or most significant top activations for that particular feature. To be complete, an auto interpretation must be sufficiently broad to encompass the feature's diverse manifestations in the data, rather than being narrowly focused on just a few activation examples. Additionally, users reported a confidence score for each rating. The overall median confidence from users was *3* with an interquartile range of *1* for both Gemma and Llama evaluation sets. To ensure the robustness of our human evaluations, the inter-rater agreement level was quantified using Krippendorff's ordinal $\alpha$. The calculated agreement was *0.64* for the gemma-2-9b set, and *0.57* for the llama-3.1-8b set, indicating substantial agreement between the evaluators.

User scores (soundness and completeness) for each feature were averaged to create a composite human rating. A visual comparison of the score distributions in Fig. 8.2 shows that fuzz scores appear more skewed and potentially over-optimistic in their assessment of feature interpretability compared to SFAL scores. We analysed the correlation (Spearman, Pearson, Kendall) between the two sets of averaged human judgments and seven sets of automated scores: those from fuzz scoring and those generated by our proposed alignment-based scoring method, varying the embedding model to assess the consistency of our process. As Tab. 8.1 shows, SFAL demonstrated a stronger positive correlation compared to the fuzzing score for all the embedding models tested on the

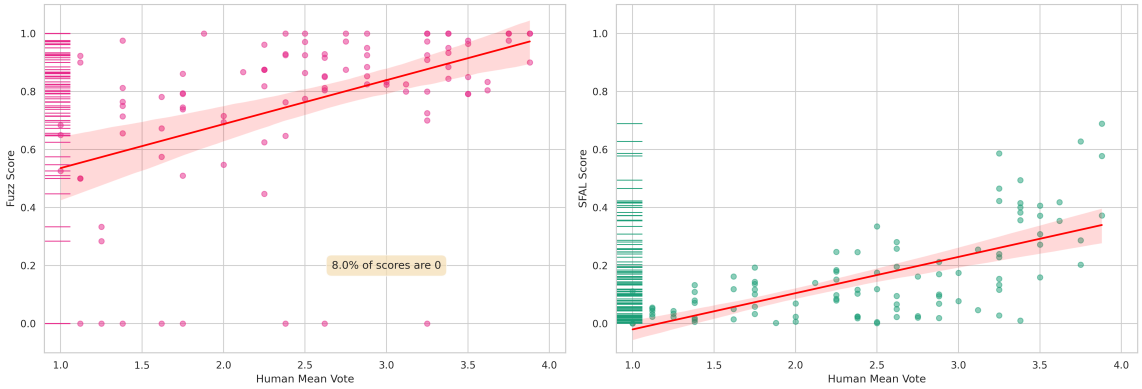



Figure 8.3: Comparison of score distribution against human judgement. On the left, we show the computed fuzz score, while on the right, we show the SFAL results.

| Metric | Gemma-2-9b | | | Llama-3.1-8b | | |
|---|---|---|---|---|---|---|
| | Pearson | Spearman | Kendall | Pearson | Spearman | Kendall |
| Fuzz score [274] | $0.47^{(***)}$ | $0.56^{(***)}$ | $0.40^{(***)}$ | $0.59^{(***)}$ | $0.60^{(***)}$ | $0.44^{(***)}$ |
| SFAL Bilingual Emb [333] | $0.63^{(***)}$ | $0.62^{(***)}$ | $0.45^{(***)}$ | $0.53^{(***)}$ | $0.56^{(***)}$ | $0.41^{(***)}$ |
| SFAL gte-Qwen2-7B-instruct [198] | $0.53^{(***)}$ | $0.50^{(***)}$ | $0.37^{(***)}$ | $0.48^{(***)}$ | $0.53^{(***)}$ | $0.39^{(***)}$ |
| SFAL Qwen3-Emb-8B (prompted) [387] | $0.66^{(***)}$ | $0.63^{(***)}$ | $0.46^{(***)}$ | $0.56^{(***)}$ | $0.60^{(***)}$ | $0.43^{(***)}$ |
| SFAL Qwen3-Emb-8B [387] | $0.66^{(***)}$ | $0.63^{(***)}$ | $0.47^{(***)}$ | $0.49^{(***)}$ | $0.55^{(***)}$ | $0.39^{(***)}$ |
| SFAL Qwen3-Emb-0.6B [387] | $0.64^{(***)}$ | $0.62^{(***)}$ | $0.46^{(***)}$ | $0.45^{(***)}$ | $0.53^{(***)}$ | $0.37^{(***)}$ |
| SFAL Qwen3-Emb-4B [387] | $0.64^{(***)}$ | $0.61^{(***)}$ | $0.44^{(***)}$ | $0.52^{(***)}$ | $0.58^{(***)}$ | $0.41^{(***)}$ |

Table 8.1: Correlation coefficients (Pearson, Spearman, Kendall) between fuzz, SFAL scores and human evaluation conducted by expert raters on the Gemma-2-9b and Llama-3.1-8b SAEs. The prompted version of Qwen3 uses an instruction to specialise the embedding for retrieval queries, while the normal version is for general similarity. Significance markers: $^{(*)}p \leq 0.05$, $^{(**)}p \leq 0.01$, $^{(***)}p \leq 0.001$, (N.S.) = not significant ($p > 0.05$).

Gemma-2-9b SAEs human evaluation set. However, SFAL slightly underperforms the fuzzing score for Llama-3.1-8b.

## 8.5 Discussion

**Autointerpretation quality.** As shown in Fig. 8.2, the distributions of SFAL and the fuzz score differ substantially. SFAL tends to assign lower values overall, showing a more selective behaviour in identifying autointerpretations as the correct interpretation of features.

**Bridging semantic and functional evaluation.** Our method's correlation with human judgments validates our framework. Instead of relying on costly LLM "oracles," we enforce internal consistency by aligning a feature's semantic meaning with its functional behaviour, derived from

co-occurrence statistics. This captures a functional signal that purely semantic checks, often focused on static human-understandability, can overlook [184]. We note that the semantic–functional alignment assumption can fail in cases where features are functionally correlated yet semantically dissimilar (e.g., a transitive predicate and its direct object), which explains some of the observed noise and highlights the complementary role of SFAL alongside more precise causal methods.

**Impact of embedding models employed.** Tab. 8.1 shows the correlations of the human evaluation with both the fuzz score and SFAL, computed using several embedding models. We assess the consistency of SFAL, varying the encoder used to create the auto-interpretation embedding. Results show that scores are consistently significant across all tested embedding models for both SAE evaluations. Model size appears to be a minor factor in scoring performance, as indicated by the small differences between models within the Qwen family.

## 8.6 Conclusion

In this work, we introduced a novel, distributional approach for evaluating the auto-interpretations of SAE features by quantifying the alignment between a feature's semantic and functional neighbourhoods. Unlike traditional methods that rely heavily on expensive and often opaque LLM-based scoring, our approach grounds interpretability assessment in the model's internal structure by capturing functional relationships through co-activation patterns and semantic intent through auto-interpretation embeddings. We demonstrated that this alignment-based metric is not only computationally efficient and scalable but also correlates well with human judgment. By reducing evaluation costs and improving scalability, this work opens the door to more practical and widespread assessments of interpretability in large-scale language models. Future work will explore more expressive similarity metrics and investigate how our generalises across architectures and domains.

## Acknowledgements

Evaluation of the open-source models was conducted on the Leonardo supercomputer with the support of CINECA-Italian Super Computing Resource Allocation, class C project IsCc9_MI-PLE (HP10CIQUBQ).

## 8.7 Limitations

Co-occurrence is a powerful but imperfect proxy for true functional linkage. However, the core contribution of this work is a significant *lowering of the cost-utility frontier* for auto-interpretation evaluation. We demonstrate performance comparable to expensive, closed-source LLM-based metrics while operating at a fraction of the computational and financial cost. Ultimately, by making robust evaluation economically feasible, our method enables the field to systematically and comprehensively assess millions of features, a critical step toward genuinely understanding and trusting these complex systems. Beyond the reliance on co-activation as a proxy for function, SFAL has two additional limitations: (i) the fixed-K neighbourhoods may not fully adapt to varying sparsity across features; (ii) our human evaluation involved only a small pool of expert raters, motivating

future work on adaptive neighbourhood selection and larger-scale, more diverse user studies; and (iii) the difficulty of supplying direct evidence for alignment at the deeper epistemic level. While we link internal representations to human-interpretable semantics, this connection relies heavily on behavioural proxies rather than explicit topological analysis or visualisation of the model's parameter space. Consequently, while the alignment is robustly proven at the behavioural level, claims regarding the strict structural isomorphism between the model's internal architecture and human concepts should be interpreted with caution, as true internal alignment remains harder to verify without granular inspection of the parameter topology.

# Conclusions and Outlook

The notion of alignment has accompanied the evolution of Natural Language Processing from its earliest distributional models to today's large-scale generative systems. This thesis has expanded and unified the concept across three complementary levels-distributional, behavioural, and epistemic—demonstrating that alignment is not just a single operation but a multidimensional principle connecting meaning, knowledge, and reasoning within computational models.

At the distributional level, we introduced SeNSe, an unsupervised framework for embedding alignment based on semantic anchor selection, capable of outperforming existing methods by leveraging local topological stability. Its application in the labour market domain through MEAL provided empirical evidence of its interpretability and robustness. Complementing these contributions, the survey Lost in Alignment offered the first systematic taxonomy of cross-lingual alignment methods for contextualised embeddings, consolidating fragmented literature into a coherent research map.

At the LLM level, we advanced the idea of alignment as factual and domain-specific control. The proposed RAG-based systems, RE-FIN and FLEX, illustrated two distinct yet complementary pathways: grounding language models in external evidence and fostering internal consistency through self-alignment. Their success in FSA highlights how alignment mechanisms can improve both accuracy and trustworthiness without the need for extensive retraining.

At the epistemic level, we explored how interpretability and factual reliability emerge from structural correspondence between internal and external representations. The SAFE framework demonstrated how sparse autoencoders can mitigate hallucinations through feature-based enrichment, while SFAL formalised the notion of semantic–functional coherence, enabling quantitative evaluation of interpretability. Together, these approaches mark a step toward models whose reasoning processes are not only aligned with facts but also transparent to human scrutiny.

The contributions presented here pave the way for a unified view of alignment where semantics, behaviour, and internal representations converge, advocating for an epistemically grounded AI aligned with truth and human understanding. Crucially, this pursuit is no longer merely a technical objective but a regulatory necessity: as the EU AI Act mandates transparency and oversight for high-risk systems, the methods proposed herein serve as foundational blocks for compliance, bridging the gap between raw performance and legally required trustworthiness.

## 8.8 Future Works

Future developments will extend the notion of alignment along three main directions. The first focuses on enhancing contextual embedding alignment by introducing qualitative supervision through data filtering and by leveraging TLM objective to improve cross-domain and cross-lingual consistency, with particular attention to preserving the original structural properties of the source model when projected onto the target one. The second direction aims to exploit contextual embedding alignment to enable a guided fine-tuning process following a teacher–student paradigm. The third line of research concentrates on hallucination mitigation, seeking to refine detection methods through a detailed analysis of output token probability distributions and to implement dynamic steering mechanisms that guide generation toward factual and coherent outputs.

# Bibliography


[1] Amro Kamal Mohamed Abbas et al. "SemDeDup: Data-efficient learning at web-scale through semantic deduplication". In: *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2023.

[2] Samir Abdaljalil et al. "SAFE: A Sparse Autoencoder-Based Framework for Robust Query Enrichment and Hallucination Mitigation in LLMs". In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. 2025.

[3] Samir Abdaljalil et al. *SINdex: Semantic INconsistency Index for Hallucination Detection in LLMs*. 2025. arXiv: 2503.05980 [cs.CL]. URL: https://arxiv.org/abs/2503.05980.

[4] Dmitry Abulkhanov et al. "LAPCA: Language-Agnostic Pretraining with Cross-Lingual Alignment". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2023, pp. 2098–2102.

[5] Hanan Aldarmaki and Mona Diab. "Context-Aware Cross-Lingual Mapping". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 3906–3911.

[6] Badr AlKhamissi et al. "A review on language models as knowledge bases". In: *arXiv preprint arXiv:2204.06031* (2022).

[7] Sawsan Alqahtani et al. "Using Optimal Transport as Alignment Objective for fine-tuning Multilingual Contextualized Embeddings". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021, pp. 3904–3919.

[8] David Alvarez-Melis and Tommi Jaakkola. "Gromov-Wasserstein Alignment of Word Embedding Spaces". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 1881–1890.

[9] Waleed Ammar et al. "Massively multilingual word embeddings". In: *arXiv preprint arXiv:1602.01925* (2016).

[10] Dogu Araci. "Finbert: Financial sentiment analysis with pre-trained language models". In: *arXiv preprint arXiv:1908.10063* (2019).

[11] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by . 2018, pp. 789–798.

[12] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. Ed. by . 2018, pp. 5012–5019.

[13] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "Learning bilingual word embeddings with (almost) no bilingual data". In: *ACL*. Ed. by . 2017, pp. 451–462.

[14] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "Learning principled bilingual mappings of word embeddings while preserving monolingual invariance". In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. Ed. by . 2016, pp. 2289–2294.

[15] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. "On the Cross-lingual Transferability of Monolingual Representations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4623–4637.

[16] Mikel Artetxe and Holger Schwenk. "Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 597–610.

[17] Mikel Artetxe Zurutuza, Gorka Labaka Intxauspe, and Eneko Agirre Bengoa. "Bilingual Lexicon Induction through Unsupervised Machine Translation". In: *Proceedings of the*. Vol. 57. ACL. 2019.

[18] Kola Ayonrinde, Michael T. Pearce, and Lee Sharkey. *Interpretability as Compression: Reconsidering SAE Explanations of Neural Activations with MDL-SAEs*. 2024. arXiv: 2410.11179 [cs.LG]. URL: https://arxiv.org/abs/2410.11179.

[19] Ion Madrazo Azpiazu and Maria Soledad Pera. "Hierarchical mapping for crosslingual word embedding alignment". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 361–376.

[20] Steve Bakos et al. "AlignFreeze: Navigating the Impact of Realignment on the Layers of Multilingual Models Across Diverse Languages". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. 2025, pp. 562–586.

[21] Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. "Llms will always hallucinate, and we need to live with this". In: *arXiv preprint arXiv:2409.05746* (2024).

[22] Yejin Bang et al. "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity". In: *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 675–718.

[23] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[24] Yoshua Bengio et al. "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.

[25] Usha Bhalla et al. "Towards unifying interpretability and control: Evaluation via intervention". In: *arXiv preprint arXiv:2411.04430* (2024).

[26] Steven Bills et al. *Language models can explain neurons in language models*. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html. 2023.

[27] Terra Blevins and Luke Zettlemoyer. "Language Contamination Helps Explains the Cross-lingual Capabilities of English Pretrained Models". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 3563–3574.

[28] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[29] Roberto Boselli et al. "Classifying online job advertisements through machine learning". In: *Future Generation Computer Systems* 86 (2018), pp. 319–328.

[30] Eleftheria Briakou, Colin Cherry, and George Foster. "Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM's Translation Capability". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. 2023.

[31] Trenton Bricken et al. "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". In: *Transformer Circuits Thread* (2023). https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[32] Trenton Bricken et al. *Using Dictionary Learning Features as Classifiers*. Oct. 2024.

[33] A Broder. "On the Resemblance and Containment of Documents". In: *Compression and Complexity of Sequences, International Conference on*. IEEE Computer Society. 1997, pp. 21–21.

[34] Peter F Brown et al. "The mathematics of statistical machine translation: Parameter estimation". In: (1993).

[35] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[36] Alexander Budanitsky and Graeme Hirst. "Evaluating wordnet-based measures of lexical semantic relatedness". In: *Computational linguistics* 32.1 (2006), pp. 13–47.

[37] Collin Burns et al. "Discovering latent knowledge in language models without supervision". In: *arXiv preprint arXiv:2212.03827* (2022).

[38] Deng Cai et al. "Recent advances in retrieval-augmented text generation". In: *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 2022, pp. 3417–3419.

[39] Erik Cambria. "Understanding Natural Language Understanding". In: *Springer, ISBN 978-3-031-73973-6* (2024).

[40] Erik Cambria et al. "Sentiment analysis is a big suitcase". In: *IEEE Intelligent Systems* 32.6 (2017), pp. 74–80.

[41] Erik Cambria et al. "Seven Pillars for the Future of Artificial Intelligence". In: *IEEE Intelligent Systems* 38.6 (2023), pp. 62–69.

[42] Steven Cao, Nikita Kitaev, and Dan Klein. "Multilingual Alignment of Contextual Word Representations". In: *International Conference on Learning Representations*. 2020.

[43] Mirko Cesarini et al. "Explainable AI for Text Classification: Lessons from a Comprehensive Evaluation of Post Hoc Methods". In: *Cognitive Computation* (2024), pp. 1–19.

[44] Haw-Shiuan Chang and Andrew McCallum. "Softmax bottleneck makes language models unable to represent multi-mode word distributions". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Vol. 1. 2022.

[45] Sihao Chen et al. "Sub-Sentence Encoder: Contrastive Learning of Propositional Semantic Representations". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 1596–1609.

[46] Tong Chen et al. "Dense X Retrieval: What Retrieval Granularity Should We Use?" In: *arXiv preprint arXiv:2312.06648* (2023). URL: https://arxiv.org/pdf/2312.06648.pdf.

[47] Xiuying Chen et al. "Towards improving faithfulness in abstractive summarization". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24516–24528.

[48] Zhongzhi Chen et al. "Truth forest: Toward multi-scale truthfulness in large language models through intervention without tuning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 19. 2024, pp. 20967–20974.

[49] Daixuan Cheng et al. "UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 12318–12337. DOI: 10.18653/v1/2023.emnlp-main.758. URL: https://aclanthology.org/2023.emnlp-main.758/.

[50] Xin Cheng et al. "Lift yourself up: Retrieval-augmented text generation with self-memory". In: *Advances in Neural Information Processing Systems* 36 (2024).

[51] Zewen Chi et al. "Cross-lingual natural language generation via pre-training". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 7570–7577.

[52] Zewen Chi et al. "Improving Pretrained Cross-Lingual Language Models via Self-Labeled Word Alignment". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 3418–3430.

[53] Zewen Chi et al. "InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 3576–3588.

[54] Seongkuk Cho et al. "A Framework for Understanding Unstructured Financial Documents Using RPA and Multimodal Approach". In: *Electronics* 12.4 (2023), p. 939.

[55] Aakanksha Chowdhery et al. "Palm: Scaling language modeling with pathways". In: *Journal of Machine Learning Research* 24.240 (2023), pp. 1–113.

[56] Paul F Christiano et al. "Deep reinforcement learning from human preferences". In: *Advances in neural information processing systems* 30 (2017).

[57] Yung-Sung Chuang et al. "DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2023.

[58] Eve V Clark. "The principle of contrast: A constraint on language acquisition". In: *Mechanisms of language acquisition*. Psychology Press, 2014, pp. 1–33.

[59] Ronan Collobert et al. "Natural language processing (almost) from scratch." In: *Journal of machine learning research* 12.7 (2011).

[60] Emilio Colombo, Fabio Mercorio, and Mario Mezzanzanica. "AI meets labor market: Exploring the link between automation and skills". In: *Information Economics and Policy* 47 (2019), pp. 27–37.

[61] Emilio Colombo, Fabio Mercorio, and Mario Mezzanzanica. "AI meets labor market: Exploring the link between automation and skills". In: *Information Economics and Policy* 47 (2019). The Economics of Artificial Intelligence and Machine Learning, pp. 27–37. ISSN: 0167-6245.

[62] Alexis Conneau and Guillaume Lample. "Cross-lingual language model pretraining". In: *Advances in neural information processing systems* 32 (2019).

[63] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[64] Timothy F Cootes et al. "Active shape models-their training and application". In: *Computer vision and image understanding* 61.1 (1995), pp. 38–59.

[65] Hoagy Cunningham et al. "Sparse autoencoders find highly interpretable features in language models". In: *arXiv preprint arXiv:2309.08600* (2023).

[66] Marco Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems* 26 (2013).

[67] Simone D'Amico et al. "Alignment of Multilingual Embeddings to Estimate Job Similarities in Online Labour Market". In: *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2024, pp. 1–10.

[68] Tanvi Dadu and Kartikey Pant. "Team rouges at SemEval-2020 task 12: Cross-lingual inductive transfer to detect offensive language". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, pp. 2183–2189.

[69] Min-Yuh Day and Chia-Chou Lee. "Deep learning for financial sentiment analysis on finance news providers". In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2016, pp. 1127–1134.

[70] Jingcheng Deng et al. "Following the Autoregressive Nature of LLM Embeddings via Compression and Alignment". In: *CoRR* (2025).

[71] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[72] Shehzaad Dhuliawala et al. "Chain-of-Verification Reduces Hallucination in Large Language Models". In: *FINDINGS OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: ACL 2024* (2024), pp. 3563–3578.

[73] Kunbo Ding et al. "A Simple and Effective Method to Improve Zero-Shot Cross-Lingual Transfer Learning". In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022, pp. 4372–4380.

[74] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. "Improving zero-shot learning by mitigating the hubness problem". In: *arXiv preprint arXiv:1412.6568* (2014).

[75] Yerai Doval et al. "Improving Cross-Lingual Word Embeddings by Meeting in the Middle". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 294–304.

[76] Kelvin Du et al. "An Evaluation of Reasoning Capabilities of Large Language Models in Financial Sentiment Analysis". In: *IEEE Conference on Artificial Intelligence (IEEE CAI)*. Singapore, 2024, pp. 189–194.

[77] Kelvin Du et al. "Financial Sentiment Analysis: Techniques and Applications". In: *ACM Computing Surveys* (2024).

[78] Kelvin Du et al. "Financial Sentiment Analysis: Techniques and Applications". In: *ACM Computing Surveys* 56.9 (2024), pp. 1–42. DOI: 10.1145/3649451.

[79] Kelvin Du et al. "FinSenticNet: A Concept-Level Lexicon for Financial Sentiment Analysis". In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2023, pp. 109–114.

[80] Abhimanyu Dubey et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[81] Esin Durmus, He He, and Mona Diab. "FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 5055–5070.

[82] Chris Dyer, Victor Chahuneau, and Noah A Smith. "A simple, fast, and effective reparameterization of IBM model 2". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 644–648.

[83] Nouha Dziri et al. "Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 2197–2214.

[84] Pavel Efimov et al. "The Impact of Cross-Lingual Adjustment of Contextual Word Representations on Zero-Shot Transfer". In: *European Conference on Information Retrieval*. Springer. 2023, pp. 51–67.

[85] Mohamed Elaraby et al. *Halo: Estimation and Reduction of Hallucinations in Open-Source Weak Large Language Models*. 2023. arXiv: 2308.11764 [cs.CL]. URL: https://arxiv.org/abs/2308.11764.

[86] Nelson Elhage et al. "Toy models of superposition". In: *arXiv preprint arXiv:2209.10652* (2022).

[87] Akiko Eriguchi et al. "Zero-shot cross-lingual classification using multilingual neural machine translation". In: *arXiv preprint arXiv:1809.04686* (2018).

[88] Shahul Es et al. "Ragas: Automated evaluation of retrieval augmented generation". In: *arXiv preprint arXiv:2309.15217* (2023).

[89] Kawin Ethayarajh. "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics. 2019.

[90] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 889–898.

[91] Sebastian Farquhar et al. "Detecting hallucinations in large language models using semantic entropy". In: *Nature* 630.8017 (2024). © 2024. The Author(s), pp. 625–630. DOI: 10.1038/s41586-024-07421-0. URL: https://doi.org/10.1038/s41586-024-07421-0.

[92] Manaal Faruqui and Chris Dyer. "Improving vector space word representations using multilingual correlation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by . 2014, pp. 462–471.

[93] Fangxiaoyu Feng et al. "Language-agnostic BERT Sentence Embedding". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 878–891.

[94] Zihao Feng et al. "Word-level Cross-lingual Structure in Large Language Models". In: *Proceedings of the 31st International Conference on Computational Linguistics*. 2025, pp. 2026–2037.

[95] Yousra Fettach, Adil Bahaj, and Mounir Ghogho. "JobEdKG: An uncertain knowledge graph-based approach for recommending online courses and predicting in-demand skills based on career choices". In: *EAAI* 131 (2024), p. 107779.

[96] Katja Filippova. "Controlled Hallucinations: Learning to Generate Faithfully from Noisy Data". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 864–870.

[97] John Firth. "A synopsis of linguistic theory, 1930-1955". In: *Studies in linguistic analysis* (1957), pp. 10–32.

[98] Philip A Fisher. *Common stocks and uncommon profits and other writings*. Vol. 40. John Wiley & Sons, 2003.

[99] Lea Frermann and Mirella Lapata. "A Bayesian model of diachronic meaning change". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 31–45.

[100] Pascale Fung. "Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus". In: *Third Workshop on Very Large Corpora*. 1995.

[101] David M Gaddy et al. "Ten pairs to tag-multilingual POS tagging via coarse mapping between embeddings". In: *Association for Computational Linguistics*. Ed. by . 2016, pp. 1307–1317.

[102] Jack Gallifant et al. "Sparse autoencoder features for classifications and transferability". In: *arXiv preprint arXiv:2502.11367* (2025).

[103] Jun Gao et al. "Representation Degeneration Problem in Training Natural Language Generation Models". In: *International Conference on Learning Representations*.

[104] Leo Gao et al. "Scaling and evaluating sparse autoencoders". In: *arXiv preprint arXiv:2406.04093* (2024).

[105] Leo Gao et al. "The pile: An 800gb dataset of diverse text for language modeling". In: *arXiv preprint arXiv:2101.00027* (2020).

[106] Yunfan Gao et al. "Chat-rec: Towards interactive and explainable llms-augmented recommender system". In: *arXiv preprint arXiv:2303.14524* (2023).

[107] Yunfan Gao et al. "Retrieval-augmented generation for large language models: A survey". In: *arXiv preprint arXiv:2312.10997* (2023).

[108] Zorik Gekhman et al. "Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?" In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 7765–7784.

[109] Hamidreza Ghader and Christof Monz. "What does Attention in Neural Machine Translation Pay Attention to?" In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2017, pp. 30–39.

[110] Davide Ghilardi et al. "Accelerating Sparse Autoencoder Training via Layer-Wise Transfer Learning in Large Language Models". In: *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Yonatan Belinkov et al. Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 530–550. DOI: 10.18653/v1/2024.blackboxnlp-1.32. URL: https://aclanthology.org/2024.blackboxnlp-1.32/.

[111] Sohom Ghosh et al. "FinRAD: Financial Readability Assessment Dataset-13,000+ Definitions of Financial Terms for Measuring Readability". In: *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*. 2022, pp. 1–9.

[112] Anna Giabelli et al. "Embeddings evaluation using a novel measure of semantic similarity". In: *Cognitive Computation* 14.2 (2022), pp. 749–763.

[113] Anna Giabelli et al. "NEO: A system for identifying new emerging occupation from job ads". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 18. 2021, pp. 16035–16037.

[114] Anna Giabelli et al. "NEO: A tool for taxonomy enrichment with new emerging occupations". In: *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19*. Ed. by . Springer. 2020, pp. 568–584.

[115] Anna Giabelli et al. "Skills2Job: A recommender system that encodes job offer embeddings on graph databases". In: *Applied Soft Computing* 101 (2021), p. 107049.

[116] Anna Giabelli et al. "WETA: Automatic taxonomy alignment via word embeddings". In: *Computers in Industry* 138 (2022), p. 103626.

[117] Gabriel Girard. *La justesse de la langue françoise: ou les différentes significations des mots qui passent pour synonimes*. French. Paris: Laurent d'Houry, 1718.

[118] Nathan Godey, Eric Villemonte de La Clergerie, and Benoît Sagot. "Anisotropy Is Inherent to Self-Attention in Transformers". In: *EACL 2024-18th Conference of the European Chapter of the Association for Computational Linguistics*. 2024, pp. 35–48.

[119] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[120] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. "Doc2Query–: when less is more". In: *European Conference on Information Retrieval*. Springer. 2023, pp. 414–422.

[121] Koustava Goswami et al. "Cross-lingual sentence embedding using multi-task learning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 9099–9113.

[122] Benjamin Graham and Bill McGowan. *The intelligent investor*. Harper Collins New York, 2005.

[123] Edouard Grave, Armand Joulin, and Quentin Berthet. "Unsupervised alignment of embeddings with wasserstein procrustes". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. Ed. by . PMLR. 2019, pp. 1880–1890.

[124] Milan Gritta and Ignacio Iacobacci. "XeroAlign: Zero-shot cross-lingual transformer alignment". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021, pp. 371–381.

[125] J Gu et al. "Universal Neural Machine Translation for Extremely Low Resource Languages". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistic. 2018.

[126] Nuno M Guerreiro et al. "Hallucinations in large multilingual translation models". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 1500–1517.

[127] Akshay Gugnani and Hemant Misra. "Implicit skills extraction using document embedding and its use in job recommendation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 08. 2020, pp. 13286–13293.

[128] Suriya Gunasekar et al. "Textbooks are all you need". In: *arXiv preprint arXiv:2306.11644* (2023).

[129] William L Hamilton, Jure Leskovec, and Dan Jurafsky. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1489–1501.

[130] Katharina Hämmerl, Jindřich Libovický, and Alexander Fraser. "Understanding Cross-Lingual Alignment—A Survey". In: *Findings of the Association for Computational Linguistics ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, Aug. 2024, pp. 10922–10943. DOI: 10.18653/v1/2024.findings-acl.649. URL: https://aclanthology.org/2024.findings-acl.649.

[131] Katharina Hämmerl, Jindřich Libovickỳ, and Alexander Fraser. "Combining Static and Contextualised Multilingual Embeddings". In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022, pp. 2316–2329.

[132] Tianyu Han et al. *MedAlpaca – An Open-Source Collection of Medical Conversational AI Models and Training Data*. 2023. arXiv: 2304.08247 [cs.CL]. URL: https://arxiv.org/abs/2304.08247.

[133] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[134] Hangfeng He, Hongming Zhang, and Dan Roth. "Rethinking with retrieval: Faithful large language model inference". In: *arXiv preprint arXiv:2301.00303* (2022).

[135] Pengcheng He, Jianfeng Gao, and Weizhu Chen. "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[136] Zhengfu He et al. *Llama Scope: Extracting Millions of Features from Llama-3.1-8B with Sparse Autoencoders*. 2024. arXiv: 2410.20526 [cs.LG]. URL: https://arxiv.org/abs/2410.20526.

[137] Kevin Heffernan, Onur Çelebi, and Holger Schwenk. "Bitext Mining Using Distilled Sentence Representations for Low-Resource Languages". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 2101–2112.

[138] Felix Hill, Roi Reichart, and Anna Korhonen. "Simlex-999: Evaluating semantic models with (genuine) similarity estimation". In: *Computational Linguistics* 41.4 (2015), pp. 665–695.

[139] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[140] Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. "Dynamic Contextualized Word Embeddings". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by . 2021, pp. 6970–6984.

[141] Ari Holtzman et al. "The Curious Case of Neural Text Degeneration". In: *International Conference on Learning Representations*. 2020.

[142] Yedid Hoshen and Lior Wolf. "An iterative closest point method for unsupervised word translation". In: *arXiv preprint arXiv:1801.06126* 3 (2018).

[143] Junjie Hu et al. "Explicit Alignment Objectives for Multilingual Bidirectional Encoders". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 3633–3643.

[144] Junjie Hu et al. "Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4411–4421.

[145] Haoyang Huang et al. "Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 2485–2494.

[146] Jing Huang et al. "RAVEL: Evaluating Interpretability Methods on Disentangling Language Model Representations". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8669–8687. DOI: 10.18653/v1/2024.acl-long.470. URL: https://aclanthology.org/2024.acl-long.470/.

[147] Lei Huang et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions". In: *ACM Transactions on Information Systems* (2024).

[148] Lei Huang et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions". In: *ACM Transactions on Information Systems* 43.2 (2025), pp. 1–55.

[149] Robert Huben et al. "Sparse Autoencoders Find Highly Interpretable Features in Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=F76bwRSLeK.

[150] Minyoung Huh et al. "The platonic representation hypothesis". In: *arXiv preprint arXiv:2405.07987* (2024).

[151] Alexandra Ils et al. "Changes in European Solidarity Before and During COVID-19: Evidence from a Large Crowd-and Expert-Annotated Twitter Dataset". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 1623–1637.

[152] Kalervo Järvelin and Jaana Kekäläinen. "Cumulated gain-based evaluation of IR techniques". In: *ACM Transactions on Information Systems (TOIS)* 20.4 (2002), pp. 422–446.

[153] Pratik Jawanpuria et al. "Learning multilingual word embeddings in latent metric space: a geometric approach". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 107–120.

[154] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1998.

[155] Rishi Jha et al. "Harnessing the universal geometry of embeddings". In: *arXiv preprint arXiv:2505.12540* (2025).

[156] Ziwei Ji et al. "Survey of hallucination in natural language generation". In: *ACM computing surveys* 55.12 (2023), pp. 1–38.

[157] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[158] Fan Jiang, Tom Drummond, and Trevor Cohn. "Pre-training Cross-lingual Open Domain Question Answering with Large-scale Synthetic Supervision". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 13906–13933.

[159] Zhuolin Jiang et al. "Cross-lingual Information Retrieval with BERT". In: *LREC 2020 Language Resources and Evaluation Conference 11–16 May 2020*. 2020, p. 26.

[160] Erik Jones et al. "Teaching Language Models to Hallucinate Less with Synthetic Tasks". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=xpw7V0P136.

[161] Nicola Jones. "AI hallucinations can't be stopped—but these techniques can limit their damage". In: *Nature* 637.8047 (2025), pp. 778–780.

[162] Martin Joos. "Description of language design". In: *The Journal of the Acoustical Society of America* 22.6 (1950), pp. 701–707.

[163] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released August 24, 2025. 2025. URL: https://web.stanford.edu/~jurafsky/slp3/.

[164] Jared Kaplan et al. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[165] Vladimir Karpukhin et al. "Dense passage retrieval for open-domain question answering". In: *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics (ACL). 2020, pp. 6769–6781.

[166] Staffs Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. 2007.

[167] Yova Kementchedjhieva et al. "Generalizing Procrustes Analysis for Better Bilingual Dictionary Induction". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. 2018, pp. 211–220.

[168] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of NAACL-HLT*. Ed. by . 2019, pp. 4171–4186.

[169] Omar Khattab and Matei Zaharia. "Colbert: Efficient and effective passage search via contextualized late interaction over bert". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020, pp. 39–48.

[170] Raphaël Khoury et al. "How secure is code generated by chatgpt?" In: *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2023, pp. 2445–2451.

[171] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. "Inducing crosslingual distributed representations of words." In: *Proceedings of COLING 2012*. Ed. by . 2012, pp. 1459–1474.

[172] Philipp Koehn and Rebecca Knowles. "Six Challenges for Neural Machine Translation". In: *ACL 2017* (2017), p. 28.

[173] Vivek Kulkarni et al. "Statistically significant detection of linguistic change". In: *WWW '15*. Ed. by . 2015, pp. 625–635.

[174] Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching Yun Chang. "Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 933–942.

[175] Philippe Laban et al. "SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization". In: *Transactions of the Association for Computational Linguistics* 10 (2022). Ed. by Brian Roark and Ani Nenkova, pp. 163–177. DOI: 10.1162/tacl_a_00453. URL: https://aclanthology.org/2022.tacl-1.10/.

[176] Guillaume Lample et al. "Unsupervised Machine Translation Using Monolingual Corpora Only". In: *International Conference on Learning Representations*. 2018.

[177] Guillaume Lample et al. "Word translation without parallel data". In: *International Conference on Learning Representations*. 2018.

[178] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. "Hubness and pollution: Delving into cross-space mapping for zero-shot learning". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by . 2015, pp. 270–280.

[179] Jaewoong Lee et al. "Learning to select question-relevant relations for visual question answering". In: *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*. 2021, pp. 87–96.

[180] Jinhyuk Lee, Alexander Wettig, and Danqi Chen. "Phrase Retrieval Learns Passage Retrieval, Too". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 3661–3672.

[181] Nayeon Lee et al. "Factuality enhanced language models for open-ended text generation". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 34586–34599.

[182] Simon Lermen, Mateusz Dziemian, and Natalia Pérez-Campanero Antolín. "Deceptive Automated Interpretability: Language Models Coordinating to Fool Oversight Systems". In: *arXiv preprint arXiv:2504.07831* (2025).

[183] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[184] Aaron J Li et al. "Interpretability Illusions with Sparse Autoencoders: Evaluating Robustness of Concept Representations". In: *arXiv preprint arXiv:2505.16004* (2025).

[185] Chong Li et al. "Improving In-context Learning of Multilingual Generative Language Models with Cross-lingual Alignment". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 8051–8069.

[186] Jinming Li et al. "GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation". In: *Proceedings of the 5th ACM SIGIR Workshop on eCommerce and NLP in Retail (eCom@SIGIR 2023)*. 2023. URL: https://ceur-ws.org/Vol-3589/paper_2.pdf.

[187] Jiwei Li et al. "Visualizing and Understanding Neural Models in NLP". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 681–691.

[188] Juntao Li et al. "Cross-lingual low-resource set-to-description retrieval for global e-commerce". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8212–8219.

[189] Junyi Li et al. "HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6449–6464. DOI: 10.18653/v1/2023.emnlp-main.397. URL: https://aclanthology.org/2023.emnlp-main.397/.

[190] Kenneth Li et al. "Inference-time intervention: Eliciting truthful answers from a language model". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 41451–41530.

[191] Tianjian Li and Kenton Murray. "Why Does Zero-Shot Cross-Lingual Generation Fail? An Explanation and a Solution". In: *Findings of the Association for Computational Linguistics: ACL 2023*. 2023, pp. 12461–12476.

[192] Xiang Lisa Li et al. "Contrastive Decoding: Open-ended Text Generation as Optimization". In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[193] Xianzhi Li et al. "Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? A Study on Several Typical Tasks". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2023, pp. 408–422.

[194] Xintong Li et al. "On the word alignment from neural machine translation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 1293–1303.

[195] Yaoyiran Li et al. "Improving Bilingual Lexicon Induction with Cross-Encoder Reranking". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by . 2022, pp. 4100–4116.

[196] Yuanzhi Li et al. "Textbooks are all you need ii: phi-1.5 technical report". In: *arXiv preprint arXiv:2309.05463* (2023).

[197] Yuxiao Li et al. "The geometry of concepts: Sparse autoencoder feature structure". In: *Entropy* 27.4 (2025), p. 344.

[198] Zehan Li et al. "Towards general text embeddings with multi-stage contrastive learning". In: *arXiv preprint arXiv:2308.03281* (2023).

[199] Zuchao Li et al. "Batgpt: A bidirectional autoregessive talker from generative pre-trained transformer". In: *arXiv preprint arXiv:2307.00360* (2023).

[200] Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. "On the Language Neutrality of Pre-trained Multilingual Representations". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 1663–1674.

[201] Tom Lieberum et al. "Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2". In: *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Yonatan Belinkov et al. Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 278–300. DOI: 10.18653/v1/2024.blackboxnlp-1.19. URL: https://aclanthology.org/2024.blackboxnlp-1.19/.

[202] Stephanie Lin, Jacob Hilton, and Owain Evans. "TruthfulQA: Measuring How Models Mimic Human Falsehoods". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3214–3252. DOI: 10.18653/v1/2022.acl-long.229. URL: https://aclanthology.org/2022.acl-long.229/.

[203] Bingbin Liu et al. "Exposing attention glitches with flip-flop language modeling". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 25549–25583.

[204] Danni Liu and Jan Niehues. "Middle-Layer Representation Alignment for Cross-Lingual Transfer in Fine-Tuned LLMs". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Wanxiang Che et al. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 15979–15996. ISBN: 979-8-89176-251-0. DOI: 10.18653/v1/2025.acl-long.778. URL: https://aclanthology.org/2025.acl-long.778/.

[205] Linlin Liu et al. "Towards Multi-Sense Cross-Lingual Alignment of Contextual Embeddings". In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022, pp. 4381–4396.

[206] Nelson F Liu et al. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173.

[207] Qianchu Liu et al. "Investigating Cross-Lingual Alignment Methods for Contextualized Embeddings with Token-Level Evaluation". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Ed. by . 2019, pp. 33–43.

[208] Ruijun Liu et al. "A survey of sentiment analysis based on transfer learning". In: *IEEE access* 7 (2019), pp. 85401–85412.

[209] Shangqing LIU et al. "Retrieval-augmented generation for code summarization via hybrid GNN.(2021)". In: *Proceedings of the Ninth International Conference on Learning Representations: ICLR*. 2021, pp. 4–8.

[210] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv e-prints* (2019). Ed. by , arXiv–1907.

[211] Ang Lu et al. "Deep multilingual correlation for improved word embeddings". In: *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*. 2015, pp. 250–256.

[212] Yu Ma et al. "Multi-source aggregated classification for stock price movement prediction". In: *Information Fusion* 91 (2023), pp. 515–528.

[213] Yu Ma et al. "Quantitative stock portfolio optimization by multi-task learning risk and return". In: *Information Fusion* 104 (2024), p. 102165.

[214] Fiona Macpherson and Dimitris Platchias. *Hallucination: Philosophy and psychology*. MIT Press, 2013.

[215] Aman Madaan et al. "SELF-REFINE: iterative refinement with self-feedback". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2024.

[216] Macedo Maia et al. "Www'18 open challenge: financial opinion mining and question answering". In: *Companion proceedings of the the web conference 2018*. 2018, pp. 1941–1942.

[217] Lorenzo Malandri et al. "MEET-LM: A method for embeddings evaluation for taxonomic data in the labour market". In: *Computers in Industry* 124 (2021), p. 103341.

[218] Lorenzo Malandri et al. "Model-contrastive explanations through symbolic reasoning". In: *Decision Support Systems* 176 (2024), p. 114040.

[219] Lorenzo Malandri et al. "Public mood–driven asset allocation: The importance of financial sentiment in portfolio management". In: *Cognitive Computation* 10.6 (2018), pp. 1167–1176.

[220] Lorenzo Malandri et al. "RE-FIN: Retrieval-based Enrichment for Financial data". In: *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*. 2025, pp. 751–759.

[221] Lorenzo Malandri et al. "SeNSe: embedding alignment via semantic anchors selection". In: *International Journal of Data Science and Analytics* (2024), pp. 1–15.

[222] Lorenzo Malandri et al. "Taxoref: Embeddings evaluation for ai-driven taxonomy refinement". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2021, pp. 612–627.

[223] Lorenzo Malandri et al. "The Good, the Bad, and the Explainer: A Tool for Contrastive Explanations of Text Classifiers." In: *IJCAI*. 2022, pp. 5936–5939.

[224] Alex Troy Mallen et al. "When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories". In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[225] Pekka Malo et al. "Good debt or bad debt: Detecting semantic orientations in economic texts". In: *Journal of the Association for Information Science and Technology* 65.4 (2014), pp. 782–796.

[226] Potsawee Manakul, Adian Liusie, and Mark Gales. "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023. URL: https://openreview.net/forum?id=RwzFNbJ3Ez.

[227] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[228] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to Information Retrieval". In: (2008).

[229] Rohan Manro et al. "A Cognitive Analysis of CEO Speeches and Their Effects on Stock Markets". In: *Proceedings of the 5th International Conference on Financial Technology (ICFT)*. Singapore, 2024.

[230] Rui Mao et al. "A survey on pragmatic processing techniques". In: *Information Fusion* 114 (2025), p. 102712. ISSN: 1566-2535.

[231] Rui Mao et al. "Discovering the cognition behind language: Financial metaphor analysis with MetaPro". In: *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2023, pp. 1211–1216.

[232] Rui Mao et al. "GPTEval: A Survey on Assessments of ChatGPT and GPT-4". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL, 2024, pp. 7844–7866.

[233] Rui Mao et al. "MetaPro 2.0: Computational Metaphor Processing on the Effectiveness of Anomalous Language Modeling". In: *Findings of the Association for Computational Linguistics: ACL*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 9891–9908.

[234] Rui Mao et al. "MetaPro: A computational metaphor processing model for text pre-processing". In: *Information Fusion* 86-87 (2022), pp. 30–43. ISSN: 1566-2535.

[235] Yuning Mao et al. "Generation-Augmented Retrieval for Open-Domain Question Answering". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 4089–4100.

[236] Joshua Maynez et al. "On Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 1906–1919.

[237] Bryan McCann et al. "Learned in translation: Contextualized word vectors". In: *Advances in neural information processing systems* 30 (2017).

[238] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018).

[239] Clara Meister, Ryan Cotterell, and Tim Vieira. "If beam search is the answer, what was the question?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 2173–2185.

[240] Kevin Meng et al. "Locating and editing factual associations in gpt". In: *Advances in neural information processing systems* 35 (2022), pp. 17359–17372.

[241] Kevin Meng et al. "Mass-Editing Memory in a Transformer". In: *ICLR*. 2023.

[242] Fabio Mercorio et al. "SFAL: Semantic-Functional Alignment Scores for Distributional Evaluation of Auto-Interpretability in Sparse Autoencoders". In: *Industry Track of the Association for Computational Linguistics: EMNLP 2025*. 2025.

[243] Coşkun Mermer and Murat Saraçlar. "Bayesian word alignment for statistical machine translation". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 182–187.

[244] BRÉAL Michel. "Essai de sémantique: science des significations". In: *Paris: Hachette* 349 (1897).

[245] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation". In: *arXiv* (2013).

[246] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[247] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[248] David Mimno and Laure Thompson. "The strange geometry of skip-gram with negative sampling". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2873–2878.

[249] Sewon Min et al. "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 12076–12100.

[250] Shervin Minaee et al. "Large language models: A survey". In: *arXiv preprint arXiv:2402.06196* (2024).

[251] Kostadin Mishev et al. "Evaluation of sentiment analysis in finance: from lexicons to transformers". In: *IEEE access* 8 (2020), pp. 131662–131682.

[252] Eric Mitchell et al. "Fast Model Editing at Scale". In: *International Conference on Learning Representations*. 2022.

[253] Bhaskar Mitra and Nick Craswell. "Neural models for information retrieval". In: *arXiv preprint arXiv:1705.01509* (2017).

[254] John Morris et al. "Text Embeddings Reveal (Almost) As Much As Text". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 12448–12460.

[255] Niklas Muennighoff et al. "MTEB: Massive Text Embedding Benchmark". In: *arXiv preprint arXiv:2210.07316* (2022). DOI: 10.48550/ARXIV.2210.07316. URL: https://arxiv.org/abs/2210.07316.

[256] Niels Mündler et al. "Self-contradictory Hallucinations of Large Language Models: Evaluation, Detection and Mitigation". In: *ArXiv* abs/2305.15852 (2023). URL: https://api.semanticscholar.org/CorpusID:258887694.

[257] Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. "A Supervised Word Alignment Method based on Cross-Language Span Prediction using Multilingual BERT". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 555–565.

[258] Roberto Navigli. "Word sense disambiguation: A survey". In: *ACM computing surveys (CSUR)* 41.2 (2009), pp. 1–69.

[259] Navid Nobani, Fabio Mercorio, Mario Mezzanzanica, et al. "Towards an Explainer-agnostic Conversational XAI." In: *IJCAI*. 2021, pp. 4909–4910.

[260] Franz Josef Och and Hermann Ney. "A systematic comparison of various statistical alignment models". In: *Computational linguistics* 29.1 (2003), pp. 19–51.

[261] Chris Olah et al. "Zoom in: An introduction to circuits". In: *Distill* 5.3 (2020), e00024–001.

[262] Reham Omar et al. "Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots". In: *arXiv preprint arXiv:2302.06466* (2023).

[263] Aitor Ormazabal et al. "Analyzing the Limitations of Cross-lingual Word Embedding Mappings". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4990–4995.

[264] Charles E Osgood. "The measurement of meaning". In: *University of Illinois* (1957).

[265] Robert Östling and Jörg Tiedemann. "Efficient word alignment with markov chain monte carlo". In: *The Prague Bulletin of Mathematical Linguistics* (2016).

[266] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[267] Filippo Pallucchini. "Anchors Selection for Cross-lingual Embedding Alignment through Time." In: *IJCAI*. 2022, pp. 5867–5868.

[268] Filippo Pallucchini et al. "Lost in Alignment: A Survey on Cross-Lingual Alignment Methods for Contextualized Representation". In: *ACM Computing Surveys* (2025).

[269] Filippo Pallucchini et al. "Self-explanatory and Retrieval-augmented LLMs for Financial Sentiment Analysis". In: *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*. 2025, pp. 131–137.

[270] Lin Pan et al. "Multilingual BERT Post-Pretraining Alignment". In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2021.

[271] Nina Panickssery. *Red-teaming language models via activation engineering*. Alignment Forum. Accessed: 2023 or later (date of access can be added here). 2023. URL: https://www.alignmentforum.org/posts/iHmsJdxgMEWmAfNne.

[272] Maria Papoutsoglou et al. "Extracting knowledge from on-line sources for software engineering labor market: A mapping study". In: *IEEE Access* 7 (2019), pp. 157595–157613.

[273] Md Rizwan Parvez et al. "Retrieval Augmented Code Generation and Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021, pp. 2719–2734.

[274] Gonçalo Paulo et al. "Automatically interpreting millions of features in large language models". In: *arXiv preprint arXiv:2410.13928* (2024).

[275] Baolin Peng et al. *Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback*. 2023. arXiv: 2302.12813 [cs.CL]. URL: https://arxiv.org/abs/2302.12813.

[276] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Ed. by . 2014, pp. 1532–1543.

[277] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227–2237. URL: https://aclanthology.org/N18-1202.

[278] Matthew E. Peters et al. "Deep contextualized word representations". In: *Proceedings of NAACL-HLT*. 2018, pp. 2227–2237.

[279] Fabio Petroni et al. "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 2463–2473.

[280] Gabriel Peyré, Marco Cuturi, and Justin Solomon. "Gromov-wasserstein averaging of kernel and distance matrices". In: *International conference on machine learning*. PMLR. 2016, pp. 2664–2672.

[281] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4996–5001.

[282] Daniele Potertì, Andrea Seveso, and Fabio Mercorio. "Can Role Vectors Affect LLM Behaviour?" In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. 2025.

[283] Giovanni Puccetti et al. "Outliers Dimensions that Disrupt Transformers Are Driven by Frequency". In: *Findings of EMNLP 2022*. Association for Computational Linguistics, 2022.

[284] Guanghui Qin and Benjamin Van Durme. "Nugget: Neural agglomerative embeddings of text". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 28337–28350.

[285] Qiang Qu et al. "Space-Time Aware Behavioral Topic Modeling for Microblog Posts". In: *IEEE Data Engineering Bulletin* 38.2 (2015), pp. 58–67.

[286] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[287] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[288] Bhaktipriya Radharapu et al. "AART: AI-Assisted Red-Teaming with Diverse Data Generation for New LLM-powered Applications". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2023, pp. 380–395.

[289] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. "Hubs in space: Popular nearest neighbors in high-dimensional data". In: *Journal of Machine Learning Research* 11.sept (2010), pp. 2487–2531.

[290] Rafael Rafailov et al. "Direct preference optimization: Your language model is secretly a reward model". In: *Advances in neural information processing systems* 36 (2023), pp. 53728–53741.

[291] Sara Rajaee and Mohammad Taher Pilehvar. "An Isotropy Analysis in the Multilingual BERT Embedding Space". In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022, pp. 1309–1316.

[292] Ori Ram et al. “In-context retrieval-augmented language models”. In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 1316–1331.

[293] Reinhard Rapp. “Identifying word translations in non-parallel texts”. In: *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. 1995, pp. 320–322.

[294] Tilman Räuker et al. “Toward transparent ai: A survey on interpreting the inner structures of deep neural networks”. In: *2023 ieee conference on secure and trustworthy machine learning (satml)*. IEEE. 2023, pp. 464–483.

[295] Nils Reimers and Iryna Gurevych. “Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. 2020.

[296] Samuel Rönnqvist et al. “Is Multilingual BERT Fluent in Language Generation?” In: *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*. 2019, pp. 29–36.

[297] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[298] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. “A survey of cross-lingual word embedding models”. In: *JAIR* 65 (2019), pp. 569–631.

[299] Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. “Skip-prop: Representing sentences with one vector per proposition”. In: *Proceedings of the 12th International Conference on Computational Semantics (IWCS)—Short papers*. 2017.

[300] Masoud Jalili Sabet et al. “SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings”. In: *EMNLP 2020*. 2020, pp. 1627–1643.

[301] Vinu Sankar Sadasivan et al. “Can AI-generated text be reliably detected?” In: *arXiv preprint arXiv:2303.11156* (2023).

[302] Alireza Salemi and Hamed Zamani. “Evaluating retrieval quality in retrieval-augmented generation”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024, pp. 2395–2400.

[303] Dominik Schlechtweg et al. “A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 732–746.

[304] Dominik Schlechtweg et al. “German in Flux: Detecting Metaphoric Change via Word Entropy”. In: *Proceedings of CoNLL*. Vol. 354. 2017, p. 367.

[305] Peter H Schönemann. “A generalized solution of the orthogonal procrustes problem”. In: *Psychometrika* 31.1 (1966), pp. 1–10.

[306] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[307] Tal Schuster et al. “Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 1599–1613.

[308] S Selva Birunda and R Kanniga Devi. "A review on word embedding techniques for text classification". In: *Innovative Data Communication Technologies and Application* (2021), pp. 267–281.

[309] Minjoon Seo et al. "Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4430–4441.

[310] Claude E Shannon. "Prediction and entropy of printed English". In: *Bell system technical journal* 30.1 (1951), pp. 50–64.

[311] Lee Sharkey et al. "Open Problems in Mechanistic Interpretability". In: *arXiv preprint arXiv:2501.16496* (2025).

[312] Mrinank Sharma et al. "Towards Understanding Sycophancy in Language Models". In: *The Twelfth International Conference on Learning Representations*. 2023.

[313] Freda Shi et al. "Large language models can be easily distracted by irrelevant context". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 31210–31227.

[314] Haoyue Shi, Luke Zettlemoyer, and Sida I Wang. "Bilingual Lexicon Induction via Unsupervised Bitext Construction and Word Alignment". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by . 2021, pp. 813–826.

[315] Weijia Shi et al. "Trusting your evidence: Hallucinate less with context-aware decoding". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. 2024, pp. 783–791.

[316] Yutaro Shigeto et al. "Ridge regression, hubness, and zero-shot learning". In: *Joint European conference on machine learning and knowledge discovery in databases*. Ed. by . Springer. 2015, pp. 135–151.

[317] Aditya Siddhant et al. "Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 8854–8861.

[318] Ankur Sinha et al. "SEntFiN 1.0: Entity-aware sentiment analysis for financial news". In: *Journal of the Association for Information Science and Technology* 73.9 (2022), pp. 1314–1335.

[319] Samuel L Smith et al. "Offline bilingual word vectors, orthogonal transformations and the inverted softmax". In: *International Conference on Learning Representations*. 2017.

[320] Anders Søgaard, Sebastian Ruder, and Ivan Vulić. "On the Limitations of Unsupervised Bilingual Dictionary Induction". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 778–788.

[321] Sahar Sohangir et al. "Big Data: Deep Learning for financial sentiment analysis". In: *Journal of Big Data* 5.1 (2018), pp. 1–25.

[322] Kacper Sokol and Peter Flach. "Explainability fact sheets: A framework for systematic assessment of explainable approaches". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 56–67.

[323] Juntong Song et al. "RAG-HAT: A Hallucination-Aware Tuning Pipeline for LLM in Retrieval-Augmented Generation". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Ed. by Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina. Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 1548–1558. DOI: 10.18653/v1/2024.emnlp-industry.113. URL: https://aclanthology.org/2024.emnlp-industry.113/.

[324] Zhaochen Su et al. "Improving Temporal Generalization of Pre-trained Language Models with Lexical Semantic Change". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by . 2022, pp. 6380–6393.

[325] ZhongXiang Sun et al. "LargePiG for Hallucination-Free Query Generation: Your Large Language Model is Secretly a Pointer Generator". In: *THE WEB CONFERENCE 2025*. 2025. URL: https://openreview.net/forum?id=MyywdOeyn0.

[326] Anirudh Sundar et al. "Steering into New Embedding Spaces: Analyzing Cross-Lingual Alignment Induced by Model Interventions in Multilingual Language Models". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Wanxiang Che et al. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 2375–2401. ISBN: 979-8-89176-251-0. DOI: 10.18653/v1/2025.acl-long.118. URL: https://aclanthology.org/2025.acl-long.118/.

[327] Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. "Cross-lingual word clusters for direct transfer of linguistic structure". In: *The 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt 2012)*. Ed. by . 2012, pp. –.

[328] Chenmien Tan, Ge Zhang, and Jie Fu. "Massive Editing for Large Language Models via Meta Learning". In: *ICLR*. 2024.

[329] Weiting Tan et al. "Multilingual Representation Distillation with Contrastive Learning". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2023, pp. 1477–1490.

[330] Xiohang Tang, Yi Zhou, and Danushka Bollegala. "Learning Dynamic Contextualised Word Embeddings via Template-based Temporal Adaptation". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023),* ed. by . 2023.

[331] Gemma Team et al. "Gemma 2: Improving open language models at a practical size". In: *arXiv preprint arXiv:2408.00118* (2024).

[332] Adly Templeton et al. "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[333] Nandan Thakur et al. "Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks". In: *arXiv preprint arXiv:2010.08240* (2020).

[334] Romal Thoppilan et al. "Lamda: Language models for dialog applications". In: *arXiv preprint arXiv:2201.08239* (2022).

[335] Jörg Tiedemann. "Parallel data, tools and interfaces in OPUS." In: *Lrec*. Ed. by . Vol. 2012. Citeseer. 2012, pp. 2214–2218.

[336] Chih-chan Tien and Shane Steinert-Threlkeld. "Bilingual alignment transfers to multilingual alignment for unsupervised parallel text mining". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 8696–8706.

[337] Carlo Tomasi and Takeo Kanade. "Shape and motion from image streams under orthography: a factorization method". In: *International journal of computer vision* 9.2 (1992), pp. 137–154.

[338] S. M Towhidul Islam Tonmoy et al. *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. 2024. arXiv: 2401.01313 [cs.CL]. URL: https://arxiv.org/abs/2401.01313.

[339] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[340] Bricken Trenton et al. "Using Dictionary Learning Features as Classifiers". In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/features-as-classifiers/index.html.

[341] Harsh Trivedi et al. "Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions". In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[342] George Tsatsaronis et al. "An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition". In: *BMC Bioinformatics* 16 (Apr. 2015), p. 138. DOI: 10.1186/s12859-015-0564-6.

[343] Arthur Turrell et al. "Using job vacancies to understand the effects of labour market mismatch on UK output and productivity". In: (2018).

[344] Matej Ulčar and Marko Robnik-Šikonja. "Cross-lingual alignments of ELMo contextual embeddings". In: *Neural Computing and Applications* 34.15 (2022), pp. 13043–13061.

[345] Marjan Van de Kauter, Diane Breesch, and Véronique Hoste. "Fine-grained analysis of explicit and implicit sentiment in financial news articles". In: *Expert Systems with applications* 42.11 (2015), pp. 4999–5010.

[346] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[347] Oleg Vasilyev, Fumika Isono, and John Bohannon. "Linear Cross-Lingual Mapping of Sentence Embeddings". In: *Findings of the Association for Computational Linguistics: ACL 2024*. 2024, pp. 8163–8171.

[348] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[349] Tu Vu et al. "FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13697–13720. DOI: 10.18653/v1/2024.findings-acl.813. URL: https://aclanthology.org/2024.findings-acl.813/.

[350] Chang Wang and Sridhar Mahadevan. "Manifold alignment using procrustes analysis". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1120–1127.

[351] Chao Wang et al. "Personalized and explainable employee training course recommendations: A bayesian variational approach". In: *ACM Transactions on Information Systems (TOIS)* 40.4 (2021), pp. 1–32.

[352] Hongwei Wang and Dong Yu. "Going beyond sentence embeddings: A token-level matching algorithm for calculating semantic textual similarity". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2023, pp. 563–570.

[353] Yufei Wang et al. "Aligning large language models with human: A survey". In: *arXiv preprint arXiv:2307.12966* (2023).

[354] Yuxuan Wang et al. "Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 5721–5727.

[355] Zirui Wang et al. "Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework". In: *International Conference on Learning Representations*. 2020.

[356] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[357] Jerry Wei et al. "Simple synthetic data reduces sycophancy in large language models". In: *arXiv preprint arXiv:2308.03958* (2023).

[358] Jerry Wei et al. *Simple synthetic data reduces sycophancy in large language models*. 2024. arXiv: 2308.03958 [cs.CL]. URL: https://arxiv.org/abs/2308.03958.

[359] Paul J Werbos. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10 (2002), pp. 1550–1560.

[360] John Wieting et al. "Simple and Effective Paraphrastic Similarity from Parallel Translations". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4602–4608.

[361] Shangyu Wu et al. "Retrieval-augmented generation for natural language processing: A survey". In: *arXiv preprint arXiv:2407.13193* (2024).

[362] Shijie Wu and Mark Dredze. "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 833–844.

[363] Shijie Wu and Mark Dredze. "Do Explicit Alignments Robustly Improve Multilingual Encoders?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 4471–4482.

[364] Shijie Wu et al. "Bloomberggpt: A large language model for finance". In: *arXiv preprint arXiv:2303.17564* (2023).

[365] Shijie Wu et al. "Emerging cross-lingual structure in pretrained language models". In: *arXiv preprint arXiv:1911.01464* (2019).

[366] Chao Xing et al. "Normalized word embedding and orthogonal transform for bilingual word translation". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by . 2015, pp. 1006–1011.

[367] Frank Xing. "Designing Heterogeneous LLM Agents for Financial Sentiment Analysis". In: *arXiv preprint arXiv:2401.05799* (2024).

[368] Frank Xing et al. "Financial sentiment analysis: an investigation into common mistakes and silver bullets". In: *Proceedings of the 28th international conference on computational linguistics*. 2020, pp. 978–987.

[369] Frank Z Xing, Erik Cambria, and Roy E Welsch. "Natural language based financial forecasting: a survey". In: *Artificial Intelligence Review* 50.1 (2018), pp. 49–73.

[370] Frank Z Xing, Filippo Pallucchini, and Erik Cambria. "Cognitive-inspired domain adaptation of sentiment lexicons". In: *Information Processing & Management* 56.3 (2019), pp. 554–564.

[371] Haoran Xu and Philipp Koehn. "Cross-lingual bert contextual embedding space mapping with isotropic and isometric conditions". In: *arXiv preprint arXiv:2107.09186* (2021).

[372] Jundong Xu et al. "Faithful Logical Reasoning via Symbolic Chain-of-Thought". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 13326–13365.

[373] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. "FinGPT: Open-Source Financial Large Language Models". In: *FinLLM at IJCAI* (2023).

[374] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. "Finbert: A pretrained language model for financial communications". In: *arXiv preprint arXiv:2006.08097* (2020).

[375] Zhilin Yang et al. "Breaking the Softmax Bottleneck: A High-Rank RNN Language Model". In: *International Conference on Learning Representations*. 2018.

[376] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019). Ed. by .

[377] Jingwei Yi et al. "Effective and Efficient Query-aware Snippet Extraction for Web Search". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 3035–3046.

[378] Wenhao Yu et al. "Chain-of-note: Enhancing robustness in retrieval-augmented language models". In: *arXiv preprint arXiv:2311.09210* (2023).

[379] G. Udny Yule. "On the Methods of Measuring Association Between Two Attributes". In: *Journal of the Royal Statistical Society* 75.6 (1912), pp. 579–652. ISSN: 09528385. URL: http://www.jstor.org/stable/2340126 (visited on 05/14/2025).

[380] Hamed Zamani and W Bruce Croft. "Embedding-based query language models". In: *Proceedings of the 2016 ACM international conference on the theory of information retrieval*. Ed. by . 2016, pp. 147–156.

[381] Boyu Zhang et al. "Enhancing financial sentiment analysis via retrieval augmented large language models". In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. 2023, pp. 349–356.

[382] Jinpeng Zhang et al. “Combining Static Word Embeddings and Contextual Representations for Bilingual Lexicon Induction”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by . 2021, pp. 2943–2955.

[383] Meng Zhang et al. “Adversarial training for unsupervised bilingual lexicon induction”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by . 2017, pp. 1959–1970.

[384] Mike Zhang et al. “SkillSpan: Hard and Soft Skill Extraction from English Job Postings”. In: *Proceedings of the 2022 Conference of the North American Chapter of the ACL: Human Language Technologies*. 2022, pp. 4962–4984.

[385] Xulang Zhang, Rui Mao, and Erik Cambria. “A Survey on Syntactic Processing Techniques”. In: *Artificial Intelligence Review* 56 (2023), pp. 5645–5728.

[386] Xulang Zhang, Rui Mao, and Erik Cambria. “SenticVec: Toward Robust and Human-Centric Neurosymbolic Sentiment Analysis”. In: *Findings of the Association for Computational Linguistics: ACL*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 4851–4863. URL: https://aclanthology.org/2024.findings-acl.289.

[387] Yanzhao Zhang et al. “Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models”. In: *arXiv preprint arXiv:2506.05176* (2025).

[388] Yue Zhang, Leyang Cui, Shuming Shi, et al. “Alleviating Hallucinations of Large Language Models through Induced Hallucinations”. In: *Findings of the Association for Computational Linguistics: NAACL 2025*. 2025, pp. 8218–8232.

[389] Zheng Zhang et al. “Cross-lingual contextual word embeddings mapping with multi-sense words in mind”. In: *arXiv preprint arXiv:1909.08681* (2019).

[390] Ruochen Zhao et al. “Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework”. In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[391] Wei Zhao et al. “Inducing Language-Agnostic Multilingual Representations”. In: *Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*. 2021, pp. 229–240.

[392] Chunting Zhou et al. “Lima: Less is more for alignment”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[393] Huiwei Zhou et al. “Learning bilingual sentiment word embeddings for cross-language sentiment classification”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by . 2015, pp. 430–440.

[394] Luyao Zhu et al. “Neurosymbolic AI for Personalized Sentiment Analysis”. In: *Proceedings of International Conference on Human-Computer Interaction (HCII)*. Washington DC, USA, 2024.

[395] Yutao Zhu et al. “Large language models for information retrieval: A survey”. In: *arXiv preprint arXiv:2308.07107* (2023).