LearningToAdapt with Word Embeddings: Domain Adaptation of Named Entity Recognition Systems

Final publisher's version (to cite): https://doi.org/10.1016/j.ipm.2021.102537

Debora Nozza[a,*], Pikakshi Manchanda[b], Elisabetta Fersini[c], Matteo Palmonari[c], Enza Messina[c]

[a] *Bocconi University, Via Sarfatti 25, 20136 Milan, Italy*
[b] *University of Exeter Business School, Rennes Drive, Exeter, EX4 4ST, UK*
[c] *DISCo, University of Milano-Bicocca, Viale Sarca 336, 20126 Milan, Italy*

Abstract

The task of Named Entity Recognition (NER) is aimed at identifying named entities in a given text and classifying them into pre-defined domain entity types such as persons, organizations, locations. Most of the existing NER systems make use of generic entity type classification schemas, however, the comparison and integration of (more or less) different entity types among different NER systems is a complex problem even for human experts. In this paper, we propose a supervised approach called L2AWE (Learning To Adapt with Word Embeddings) which aims at adapting a NER system trained on a source classification schema to a given target one. In particular, we validate the hypothesis that the embedding representation of named entities can improve the semantic meaning of the feature space used to perform the adaptation from a source to a target domain. The results obtained on benchmark datasets of informal text show that L2AWE not only outperforms several state of the art models, but it is also able to tackle errors and uncertainties given by NER systems.

Keywords: Named Entity Recognition, Domain Adaptation, Word Embeddings

*This is to indicate the corresponding author.
Email addresses: debora.nozza@unibocconi.it (Debora Nozza), p.manchanda@exeter.ac.uk (Pikakshi Manchanda), elisabetta.fersini@unimib.it (Elisabetta Fersini), matteo.palmonari@unimib.it (Matteo Palmonari), enza.messina@unimib.it (Enza Messina)

2010 MSC: 68T50

1. Introduction

With the continuous and fast evolution of the Internet and the advent of Social Media, the amount of unstructured textual data produced by the social interactions among people has become a huge hidden treasure of knowledge. In order to exploit such valuable insights for decision-making purposes, textual data needs to be processed to extract actionable insights in a machine-readable form by means of Natural Language Processing (NLP) techniques [23].

Named Entity Recognition and Classification is one of the key Information Extraction (IE) tasks, which is concerned with identifying **entity mentions**, which are text fragment(s) denoting real-world objects, from unstructured text and classifying them into entity types according to a given **classification schema**. Extracting valuable information from user-generated content in the form of entity mentions, events and relations is of utmost significance for knowledge discovery from natural language text.

For example, given the sentence,

> "*@EmmaWatson no1 can play hermione better than u*",

the process of named entity recognition will identify the entity mentions as:

> "[***@EmmaWatson***] *no1 can play* [***hermione***] *better than u*".

Consequently, the named entity classification process will assign an entity type to the entity mentions as indicated below:

> "[***@EmmaWatson***]$_{Person}$ *no1 can play* [***hermione***]$_{Character}$ *better than u*".

Over the past few years, several research studies towards Information Extraction have been proposed, giving leeway to the emergence of numerous academic and commercial NER systems usually characterized by *generic classification*

schemas i.e., schemas aimed at capturing general knowledge about the world by providing basic notions and concepts for things [21].

The possibility to easily access and exploit these sophisticated NER systems through APIs or pre-trained models became fundamental for addressing tasks such as Data Integration [34, 58], Question Answering [55, 5, 73], Privacy Protection [20, 46] , and Knowledge Base Construction [56].

As introduced in [63], NER models can be broadly distinguished into two main categories: (i) supervised machine learning models trained on large manually-labeled corpora [24]; and (ii) knowledge-based methods [67, 52] relying on lexicons and domain-specific knowledge. Further distinctions in NER models have also been studied in the state of the art in terms of:

(a) level of automation, i.e., pre-defined rules (declarative rule language) [8], automated processes (machine learning) [74, 25] and hybrid approaches [31];

(b) type of text, i.e., formal text such as news archives [51, 81, 32] and informal text such as blog posts, twitter feed, emails etc. [65, 43, 54, 27, 47];

(c) recognition and classification of named entities based on the use of domain-dependent [71, 17, 33] or domain-independent classification schemas [1, 67, 44].

However, both supervised and knowledge-based NER models suffer from two main limitations:

- The amount of data available to accurately train a NER system for a different domain classification schema can be limited, due to time, quality and cost constraints on the labeling activity. However, NER systems based on machine learning models (e.g. Conditional Random Fields [39], Hidden Markov Models [83] or Labeled LDA [64]) commonly assume that the training and test data must be represented in the same feature space and have the same underlying distribution. When a NER system needs to be adapted to a new target classification schema, this assumption may not hold. Since new incoming data may be characterized by a different

representation space and follow a different data distribution with respect to the source data, the lack of a training set for different data distributions can pose problems in terms of human effort, resource and time expenses. This scenario is perfectly depicted in the #Microposts2015 Challenge [66], where the number of training instances ($\sim$ 3500) is not sufficient for inducing a pre-trained NER system to identify and classify entity mentions according to a different domain schema.

- Beyond the machine learning approaches, the task of Named Entity Recognition and Classification can be also performed by exploiting Knowledge Bases (KBs). Knowledge Bases, also referred to as Knowledge Graphs, are defined by large networks of entities (representing real-world objects), their semantic types, properties, and relationships between entities [37]. In this case, identification of entity mentions in a given text is performed by looking for corresponding KB entities, i.e., extracting a list of all possible KB entities for a given entity mention. However, different NER systems can refer to different KBs (e.g. Wikipedia, DBpedia, Freebase, etc.) that are not guaranteed to be available and accessible at any time. For example, the system proposed in [65] trains a LabeledLDA model using Freebase as underlying ontology, which was shut down in 2014. Moreover, the dimensions of KBs increase rapidly due to newly evolving entities that users identify every day. In this case, it could be also expensive to frequently update the exploited knowledge.

An additional limitation common to both approaches relates to the difficulty in comparing and integrating different NER systems when different domain classification schemas are under consideration. A semantic alignment of different schemas with few instances of entity mentions from the training dataset can be very complex.

Although most of the NER systems make use of *generic entity types*, it is evident from Table 1 that there are considerable differences among them. This is motivated by the fact that, because of varying application scenarios and/or

Table 1: Popular commercial and academic Named Entity Recognition systems with their corresponding Generic Classification Schemas.

NER System	**Website**	**Generic Entity Types**
OSU Twitter NLP Tools [65]	https://github.com/aritter/twitter_nlp/	Band, Company, Facility, Geo-Location, Movie, Other, Person, Product, Sportsteam, TVshow
NERD [67]	http://nerd.eurecom.fr/	Thing, Amount, Animal, Event, Function, Location, Organization, Person, Product, Time
Standford NER [24]	https://nlp.stanford.edu/software/CRF-NER.shtml	Person, Organization, Money, Percent, Location, Date, Time
Dandelion API	https://dandelion.eu/	Person, Works, Organisations, Places, Events, Concepts
Google Cloud Natural Language API	https://cloud.google.com/natural-language/	Person, Consumer Good, Organization, Event, Location, Other
IBM Natural Language Understanding	https://www.ibm.com/watson/services/natural-language-understanding/	Anatomy, Award, Broadcaster, Company, Crime, Drug, EmailAddress, Facility, GeographicFeature, HealthCondition, Hashtag, IPAddress, JobTitle, Location, Movie, MusicGroup, NaturalEvent, Organization, Person, PrintMedia, Quantity, Sport, SportingEvent, TelevisionShow, TwitterHandle, Vehicle
Ambiverse	https://www.ambiverse.com/natural-language-understanding-api/	Person, Location, Organization, Event, Artifact, Other, Unknown
Bitext	https://www.bitext.com/	Person name, Car license plate, Place, Phone number, Email address, Company/Brand, Organization, URL, IP address, Date, Hour, Money, Address, Twitter hashtag, Twitter user, Other alphanumeric, Generic
MeaningCloud	https://www.meaningcloud.com/	Event, ID, Living Thing, Location, Numex, Organization, Person, Process, Product, Timex, Unit, Other
Ingen.io	https://ingen.io/	Person, Organization, Location, Geo Political Entity, Misc, Event, Structure, Category, Lang, Artwork
Rosette	https://www.rosette.com/	Location, Organization, Person, Product, Title, Nationality, Religion, Identifier, Temporal
Thomson Reuters Open Calais	http://www.opencalais.com/	Company, Person, Geography, Industry Classifications, Topics, Social Tags, Facts, Events
Alias-i Lingpipe	http://alias-i.com/lingpipe/	Person, Locations, Organizations
AYLIEN	https://aylien.com/text-api/	Person, Location, Organization, Product, Keyword, URLs, emails, telephone numbers, currency amounts, Percentage, Date
ANNIE	http://services.gate.ac.uk/annie/	Person, Location, Organization, Money, Percent, Date, Address, Identier, Unknown

requirements, different NER systems use different entity classification schemas to classify the discovered entity mentions into entity types.

From the generic classification schemas listed in Table 1, it is possible to derive several considerations:

- Only the type *Person* is equivalently reported in all the generic schemas;
- There are few types (e.g. *Location*) that are present in all the schemas but with different associated labels (e.g. *Location* / *Geo-Location* / *Place* / *Geography*);
- Several types (e.g. *Product*) are equivalently used by few NER systems, while in others they are either not present, attributable to other corresponding types (e.g. *Thing* / *Artifact* / *Artwork* / etc.) or partitioned in multiple types (e.g. *Movie*, *TelevisionShow*, *Vehicle*, etc.);
- Since each generic schema is composed of different types, the *Other* (or *Unkown*) type can assume different meanings depending on the other types involved.
- It is also notable the use of types that are particularly related to a language register, such as *Hashtag*, *Twitter user* or *IP address*.

This paper tries to overcome the limitations of adapting NER systems while taking into account the remarks about generic classification schemas available in the current panorama of NER tools. In the following subsection, we summarize the main contributions of the proposed approach with respect to the state of the art, highlighting the main corresponding findings.

1.1. Contributions & Findings

In this paper, we present a novel supervised approach called **Learning To Adapt with Word Embeddings (L2AWE)**[1] which exploits the distributional representation of named entities for adapting named entity types predicted by a NER system trained on a source generic classification schema to a

[1] We release the source code of our approach for reproducibility purposes at `https://github.com/dnozza/L2AWE`.

given target one. The main contributions of the paper with respect to the state of the art are summarized below highlighting the corresponding key findings:

- Unlike traditional NER systems which are based on machine learning models [39, 83, 64], L2AWE does not need to re-train the underlying NER model to be adapted to a new target generic classification schema. This allows for easy adaptation of existing NER systems and overcome the problem of lack of additional training time and training data labeled with the target entity types. By analyzing the learning curves associated to the proposed model, we derive as main finding that with a small percentage of training data (data annotated with the target entity types), our model is able to achieve good performance of adaptation.

- The proposed approach adopts a well-known distributional representation for representing entity mentions, i.e., word embeddings. The hypothesis is that, among all the implicit aspects of a word, this representation will be able to reflect the entity type as well, following the distributional hypothesis that words that occur in the same contexts tend to have similar meanings [28]. In particular, two different state of the art pre-trained word embedding models have been evaluated, and several aggregation functions for dealing with multi-word entity mentions have been compared. As main finding, we point out *GoogleNews(W2V)* Word Embeddings with the *mean* aggregation function as a suitable distributional representation model for enabling the adaptation a NER system.

- Finally, L2AWE contributes to tackle errors and uncertainties given by NER system and to correctly classify the most difficult cases when a source entity type could be associated to two (or more) target entity types. By analyzing the results obtained on benchmark datasets of informal text, we derive as main finding that the proposed approach outperforms not only traditional baselines, but also recent state of the art models specifically designed for adaptation purposes.

The subsequent sections are organized as follows. Section 2 introduces the proposed model for addressing the adaptation of NER systems based on generic classification schemas. An overview of the evaluation datasets and the results of the performed analysis are presented in Section 3 and Section 4 respectively. Results have revealed three main findings: (1) L2AWE is not only able to adapt an existing NER to a new target classification schema, but also to correct some of the NER misclassifications; (2) word embeddings provide a remarkable improvement on the adaptation performance; (3) L2AWE outperforms not only a manual mapping approach but also several baselines and state of the art models. Section 5 presents some related works, while Section 6 reports some conclusions and future work.

2. Learning to Adapt with Word Embeddings

2.1. Open problems

A first investigation aimed at dealing with the issue of comparing and integrating NER systems with different entity types has been presented in [67], where a manual mapping between generic schemas has been defined. As represented in Figure 1, a manual mapping is a deterministic mapping from the source entity types to the target entity types manually defined by a human expert.

Although this study represents a first step towards the definition of cross-ontology NER systems, there is still a need for automatic mapping methods that can be used for establishing mappings between cross-domain entity types without the need for human intervention. One of the main underlying motivations is that manual mapping is a fairly subjective task that is strictly related to the interpretation of manual annotators. Additionally, it is quite difficult to adapt manual mappings uniformly across diverse domains, in cases where domain changes might incur. In order to enable automatic mappings, some open problems need to be accurately addressed:

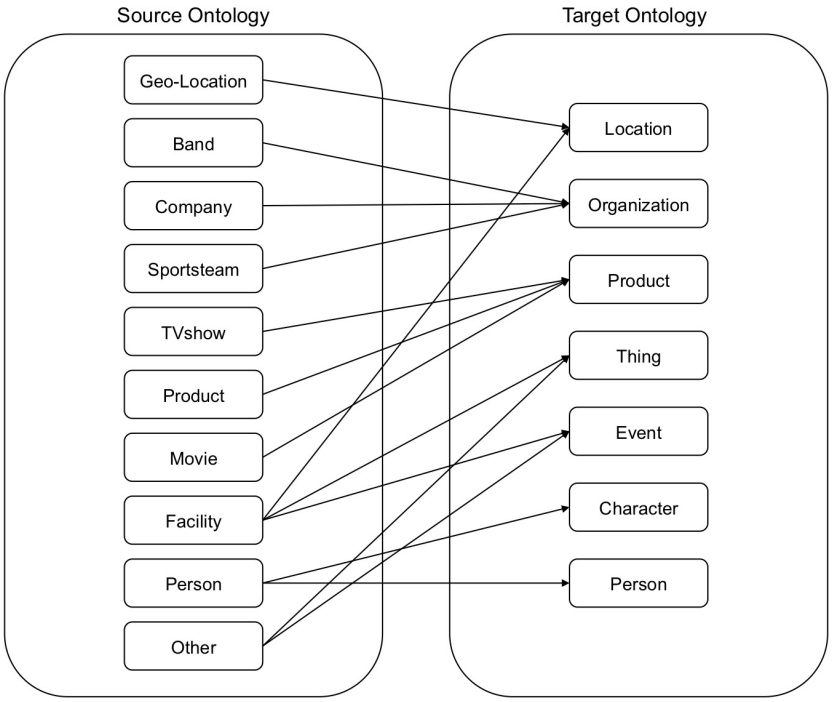


Figure 1: Manual mappings between two generic classification schemas.

1. **Mention Misclassification:** Entity mentions are often misclassified by supervised NER systems mainly because of two different reasons: (i) the training set is composed of very few instances, and (ii) the training set is characterized by an imbalanced distribution over the entity types. Consider, for instance, an entity mention *Black Sea* (i.e., location) that could be erroneously recognized as *Band* as per the source entity type. A deterministic manual mapping - as the one reported in Figure 1 - would map it as (*Organization*), thus, propagating the error.

2. **Type Uncertainty:** There are also cases in which the type of an identified entity mention may be particularly uncertain, since a mention may have subtly different meanings in the real-world. In this case, the decision of determining its type becomes difficult. While well-structured texts provide meaningful insights into the contextual usage of a mention, there

can still be cases where it is difficult for an entity recognition system to correctly classify a mention. Consider, for instance, the entity mention *Starbucks* in a well-structured document snippet:

> "*It now takes 300 stars to reach gold level status at Starbucks, which means spending approximately ...*"

A NER system could be uncertain about the type to be associated with *Starbucks*, because it could be equally probable to classify the entity mention as a *Geo-Location* (a particular Starbucks shop), or a *Company* (the Starbucks company) as per the source entity types. This ambiguity needs to be solved for determining the correct type in the target classification schema.

3. **Fork Mappings:** There are cases where mentions classified as one type according to the source classification schema could be cast into one among two (or more) different types in the target schema. Currently, focusing on Figure 1, three cases of fork mappings can be observed:
   - ◇ mapping of type *Person* in the source schema to the types *Person* or *Character* in the target schema,
   - ◇ mapping of type *Other* in the source schema to the types *Thing* or *Event* in the target schema,
   - ◇ mapping of type *Facility* in the source schema to the types *Thing*, *Event* or *Location* in the target schema,

2.2. The proposed solution

In order to tackle the above-mentioned issues arising when it is intended to adapt a NER system trained on a source classification schema to a given target one, this paper presents an approach called **Learning To Adapt with Word Embeddings (L2AWE)**, that extends the works discussed in [50, 22] by exploiting a distributional representation for obtaining a richer semantic input space. The adoption of such high-level input representation is motivated by the intuition that word embeddings will be able to capture all the implicit aspect

of a word including its entity type since entities of the same type are expected to appear in a similar context. Although the proposed approach for mapping entity types from a source to a target classification schema has been investigated on microblog posts, it can be applied to a variety of different textual formats. We consider the problem of adapting the types of entity mentions from a source to a target classification schema as a machine learning problem. In particular, given a set of entity mentions identified by a NER model originally trained according to a source schema, our main goal is to learn how to map the source type probability distribution to the target one.

More precisely, let R_S be a NER model trained on a set $\Omega_S = \{s_1, s_2, ..., s_{n_s}\}$ of entity mentions annotated according to a source classification schema O_S. Let $\Omega_T = \{t_1, t_2, ..., t_{n_t}\}$ be the set of entity mentions that needs to be automatically labeled according to a target schema O_T. The problem of labelling Ω_T according to O_T by using R_S can be viewed as a transfer learning problem [59]. In particular, the main goal is to learn a target predictive function $f(\cdot)$ in Ω_T using some knowledge both in the source schema O_S and the target schema O_T. More formally, let $P(\Omega_T, O_S)$ be the distribution in the source schema used to label an entity mention $t_i \in \Omega_T$ with the most probable type $y_S^* \in O_S$ according to R_S and let $E : \Omega_T \rightarrow \mathbb{R}^m$ be the function that maps the entity mention $t_i \in \Omega_T$ to a m-dimensional embedding representation. The input space of the investigated adaptation problem is defined as $X_{P^\frown E} = P(\Omega_T, O_S)^\frown E(\Omega_T)$, where $^\frown$ is the concatenation symbol. Thus, the input space is the concatenation of the probability distribution in the source schema and the embedded representation related to the entity mention $t_i \in \Omega_T$. Let $y_T \in O_T$ be the type in the target schema that the adaptation model should discover. Now, the adaptation of a source type y_S (of a given entity mention) to a target type y_T can be modeled as a learning problem aimed at seeking a function $\phi : X_{P^\frown E} \rightarrow y_T$ over the hypothesis space Φ. In our case, it is convenient to represent ϕ as a function $f : X_{P^\frown E} \times y_T \rightarrow \mathbb{R}$ such that:

$$g(P(t_i, y_S)^\frown E(t_i)) = \arg \max_{y_T \in O_T} f\Big((P(t_i, y_S)^\frown E(t_i)), y_T\Big) \quad (1)$$

In order to solve this problem, it is necessary to create an input space representing each entity mention t_i that can be used for learning to map the predicted source type $y_S \in O_S$ to the target type $y_T \in O_T$. As formally introduced, the input space $X_{P \frown E}$ for each entity mention t_i corresponds to the union of the explicit distribution given by R_S, $P(t_i, y_S)$, and its embedded representation $E(t_i)$. The output space denotes the most probable type $y_t \in O_T$. Using a model that is able to estimate a posterior distribution of y_T, we can therefore estimate the type distribution $P(\Omega_T, O_T)$ in the target schema. A graphical example of the proposed approach is reported in Figure 2.

The aim of L2AWE is then to learn the function f that is able to correctly label an entity mention $t_i \in \Omega_T$ according to the prediction $P(t_i, y_S)$ given by a NER model previously trained on Ω_S and its embedding representation $E(t_i)$. To accomplish this task, any machine learning algorithm can be adopted.

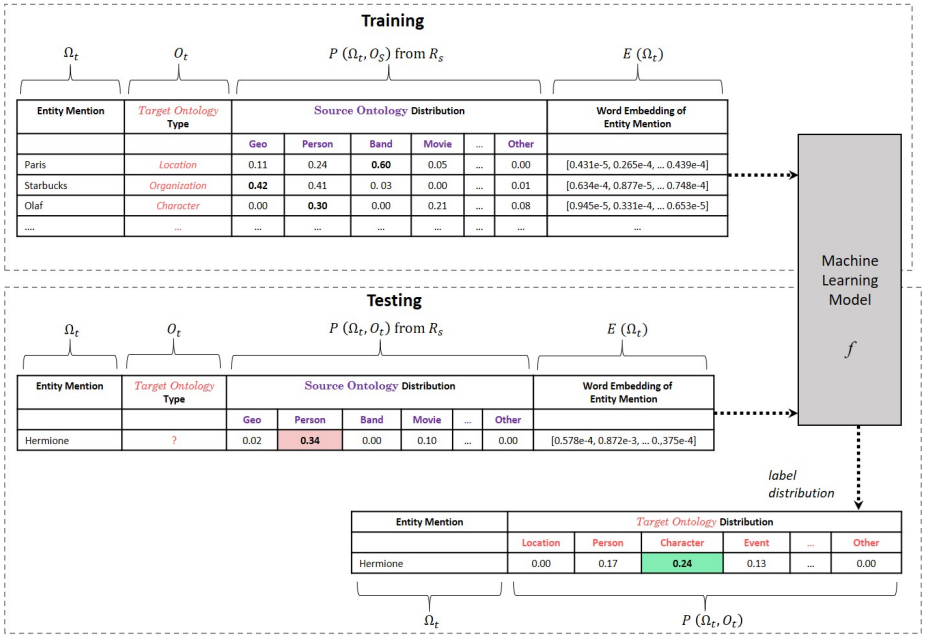


Figure 2: Graphical example of L2AWE.

2.3. Modeling the Embeddings

The additional input information provided by the embedding representation of the entity mention can strongly influence the performance of the adaptation model. The expected improvement is strictly related to the enhanced semantic meaning of the input representation. The core idea behind word embeddings is that words that appear in similar contexts should have similar vector representations. The proposed approach is motivated by the intuition that, among all the implicit aspects of the word, word embeddings will also reflect the *entity type* property. For instance, it is intuitive to think that words of type *Person* are used in the same context.

Since the amount of available training data is typically not enough for training a word embeddings model from scratch, we considered different pre-trained models for mapping entities to real-valued vectors:

- **Wiki2Vec**: these word embeddings have been obtained by training the Skip-gram model over a Wikipedia dump.
- **GoogleNews(W2V)**: Google News is the first corpus subjected to the learning of Word2vec (W2V) models [53]. This corpus is composed of 100 billion words. The model is available online[2] and it contains 300-dimensional vectors for 3 million words and phrases trained by CBOW model with negative sampling.
- **BERT** [13]: BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained contextual representation model which enabled researchers to obtain state of the art performance on numerous NLP tasks by fine-tuning the representations on their data set and task, without the need for developing and training highly-specific architectures [57]. The model has been pre-trained on the the BooksCorpus (800M words) [84] and English Wikipedia (2,500M words).

[2] `https://code.google.com/archive/p/word2vec/`

Word Embeddings. A word embeddings model is defined as a mapping $C : V \to \mathbb{R}^m$ that associates to each word in the vocabulary $w^* \in V$ a real vector $C(w^*) \in \mathbb{R}^m$. Indeed, given an entity mention t_i composed of several words $t_i = \{w_1^i, \ldots, w_n^i\}$, the function E can be written as the aggregation of the mapping C over all the words w_j^i:

$$E(t_i) = \bullet(w_1^i, \ldots, w_n^i), \tag{2}$$

where $\bullet$ is the aggregation function. This is a common approach addressed in several state of the art studies [16, 78, 11]. Beyond the commonly investigated aggregation functions *max*, *min* and *mean*, the *first* aggregation that corresponds to take the word embeddings of the first word only has also been evaluated (eventually considered as the one carrying the type information, e.g. *University of Milano-Bicocca*). Note that, when an entity mention is a single word, the function E will behave exactly like the original mapping C.

Contextual Embeddings. A contextual word embeddings model is defined as a mapping $C : D \to \mathbb{R}^m$ that associates to each document $d \in D$ a real vector $C(d) \in \mathbb{R}^m$. Note that each document d can be composed of any number of words, from single tokens to sentences.

In this paper, we investigate two different approaches for obtaining embedded representations of named entities. The first one, named $\mathbf{BERT}_s$, consider a named entity as a document[3]. In the second one, named $\mathbf{BERT}_t$, we further investigate the model by extracting the embeddings of single words within the context of the sentence. For example, given the sentence "Emma Watson is a great actress" where *Emma Watson* is the named entity, we give the sentence as input to the model and then we extract the layer representation of the single tokens *Emma* and *Watson*. This is different from all the previously presented approach that consider only the named entity and not its context. After we have obtained the representation for each token, we combine it exactly as we

[3] The representation has been obtained by exploiting *bert-as-a-service* available at `https://github.com/hanxiao/bert-as-service`.

previously described for word embeddings approaches[4]. Following the feature-based approach described in [13], the representation of single tokens is extracted by taking the sum of their last 4 hidden layers.

3. Experimental Settings

This section presents the investigated datasets and the comparative baselines used to evaluate the proposed approach. Moreover, different evaluation performance measures have been explored to analyze the performance of L2AWE for the adaptation problem.

3.1. Datasets

To perform an experimental analysis of the proposed approach, three benchmark datasets of microblog posts have been considered as **ground truth (GT)**. Two datasets were made available by the Named Entity Recognition and Linking Challenges for the #Microposts2015 [66] and #Microposts2016 [68] challenges. In particular, these ground truths are composed of 3,498 and 6,025 posts respectively, with a total of 4,016 and 8,664 entity mentions. The third dataset has been published in the context of the shared task "Novel and Emerging Entity Recognition at the 3rd Workshop on Noisy User-generated Text (W-NUT) [12]. The dataset consists of annotated texts taken from three different sources (Reddit, Twitter, YouTube, and StackExchange comments) that focus on emerging and rare entities. The ground truth of this dataset is composed of 5,691 posts respectively, with a total of 3,890 entity mentions.

Beyond word embeddings, the input space has been derived using the state of the art NER system named **T-NER** [65], specifically conceived for dealing with user-generated contents. In particular, T-NER makes use of Conditional Random Fields [39] and Labeled LDA [64] to derive $P(\Omega_T, O_S)$, one of the components of the L2AWE input space. T-NER is trained using an underlying

[4] Note that we use the term *token* instead of word because BERT uses the WordPiece tokenizer [79] where sentences can be splitted in words and subwords.

source schema O_S (which we refer to as the Ritter Schema) to finally derive a NER model R_S. In particular, the source schema (Ritter Schema, O_S) is composed of the types: *Band, Company, Facility, Geo-Location, Movie, Other, Person, Product, Sportsteam, TVshow*. Once the entity mentions are recognized by R_S and classified according to O_S, they need to be mapped to the entity types available in the target schema O_T). The target Microposts Schema is composed of the types: *Character, Event, Location, Person, Product, Organization, Thing*, while the target W-NUT17 Schema is composed of the types *Location, Corporation, Person, Product, Group, Creative work*.

T-NER identifies a total of 2,535, 4,394 and 3,090 entity mentions from the #Microposts2015, #Microposts2016 and W-NUT17 datasets respectively. In order to create the input space for the proposed L2AWE model, it is necessary to create a training set, where, for each entity mention identified by T-NER, the probability distribution $P(\Omega_T, O_S)$, the embedded representation $E(\Omega_T)$, the source type $y_S \in O_S$ and the target type $y_T \in O_T$ should be derived. While the probability distribution and the source type are explicitly provided by the T-NER system, the target type needs to be specified. However, when selecting a target type, it should be taken into account that an entity mention recognized by T-NER could be incorrectly segmented, and some tokens of multi-word entity can be classified as non-entity or a single-word entity can be coupled with some adjoining words and therefore incorrectly segmented as a multi-word entity. Two examples are reported below.

"*The* [***Empire State***]$_{Geo-Location}$ [***Building***]$_{Other}$ *is amazing!*".

"[***Paris Hilton*** *will*]$_{Person}$ *be in Venice next week!*".

To finally induce the L2AWE model, a *training set* (one for each dataset) has been **automatically** constructed by exploiting a string similarity measure (i.e., edit distance) which captures only the perfect matches between the mentions identified by T-NER and the mentions in the ground truth datasets. This means that, for each tweet in the datasets (gold standards), it has been associated with

each entity mention t_i*(T-NER)* given by T-NER with the most similar entity mention $t_j(GT)$ in the ground truth. A couple $< t_i, y_T >$ is added to the training set if and only if there is a perfect match between the entity mentions t_i*(T-NER)* and $t_j(GT)$, where y_T is the correct type for that mention in the target schema (made available from the ground truth). This automatic procedure for generating the training sets used by the L2AWE model is applicable to any labeled benchmark.

Table 2: Type Distribution (%) according to the Ritter Ontology (O_S).

	#Microposts2015	**#Microposts2016**	**W-NUT17**
Band	3.19	3.26	6.15
Company	8.86	6.99	7.51
Facility	1.99	2.53	9.03
Geo-Loc	28.86	33.17	23.17
Movie	1.87	1.86	4.82
Other	11.93	12.32	8.97
Person	35.24	33.97	27.28
Product	3.67	2.86	3.30
Sportsteam	3.07	2.16	2.46
TVshow	1.33	0.87	7.31

Table 3: Type Distribution (%) according to the Microposts Schema (O_T).

	#Microposts2015	**#Microposts2016**
Character	1.27	1.00
Event	1.75	3.83
Location	30.60	37.63
Organization	24.82	19.85
Person	31.69	29.57
Product	7.53	5.83
Thing	2.35	2.30

As a result, the training sets for #Microposts2015, #Microposts2016 and W-NUT17 are composed of 1,660, 3,003 and 2,108 training instances respectively. Tables 2, 3 and 4 show the distribution of entity types in the obtained training set, respectively referring to the Ritter Schema (O_S) and Microposts and W-NUT17 Schema (O_T). It is worth noticing that the distribution is strongly

Table 4: Type Distribution (%) according to the W-NUT17 Ontology (O_T).

	W-NUT17
Location	24.57
Corporation	9.64
Person	42.43
Product	6.57
Group	9.98
Creative work	6.81

imbalanced, much like real-world user-generated content. While *Person* and *Location* (or *Geo-Location*) are clearly the dominant entity types, other types are barely present (e.g. *Character*, *Event*, *TVshow*, *Movie*).

3.2. Baseline Models

In order to compare the proposed approach with a reference, seven **baseline** models have been considered:

- Baseline-Deterministic **(BL-D)**: it is based on the manual mapping between O_S and O_T shown in Figure 1.
- Baseline-Probabilistic **(BL-P1)**: it extends the previous baseline in order to deal with fork mappings in a non-deterministic way. In particular, for those mentions in O_S that can be classified in more than one type in O_T, the target type has been sampled according to the a priori distribution of mappings in the training set (e.g. 30% of *Person* entity mentions in O_S are classified as *Character* and 70% as *Person* in O_T).
- Baseline-Probabilistic **(BL-P2)**: A major downside of using the deterministic manual mapping (BL-D) is that since it directly depends on the output of the T-NER system, it will never be able to correct the target type of the mentions which have been incorrectly classified by T-NER. For this reason, an additional probabilistic baseline (BL-P2) has been introduced. For each mention, given the associated source type $y_S \in O_S$, the target type y_T has been sampled from the distribution $P(O_T|y_S \in O_S)$ estimated on the training set.

- Conditional Random Fields **(CRF)**: the most widely used named entity recognition model, named Conditional Random Field model [39], has been trained and tested directly over the target schema.

- Fine-tuned Bidirectional Encoder Representations from Transformers on Named Entity Recognition task (**BERT**$_{NER}$): we used a sentence encoder model pre-trained on large unlabeled text, like BERT, to perform downstream tasks through fine-tuning [13]. This permits to take advantage of the broad language knowledge learned during pre-training and to use it for a specific task (using less data and training fewer parameters). In our experiments, we fine-tuned BERT base model for the task of named entity recognition directly on the target schema.

- LearningToAdapt **(L2A)**: in order to understand the impact of the distributional representation, we compare L2AWE to our earlier proposed approach called L2A [50, 22]. L2A is aimed at adapting a NER system trained on a source schema to a given target domain by only exploiting the probability distribution space (X_P).

- Cross-Domain Model Adaptation for Named Entity Recognition **(CDMA-NER)**: a recently proposed and promising domain adaptation model for NER introduced in [44]. Lin et al. propose a transfer learning approach with neural networks, where a bidirectional LSTM model augmented with a CRF layer is exploited for cross-domain NER. Since CDMA-NER performs named entity recognition with its own model, we had to deal with a fair comparison with our L2AWE model. In particular, from the original set of data, we selected only the mentions that were identified both by CDMA-NER and T-NER (and therefore used by L2AWE). When comparing CDMA-NER to L2AWE, the used datasets are, therefore, composed of 955 and 1,918 instances for #Microposts2015 and #Microposts2016 respectively.

3.3. Performance Measures

In order to evaluate the different model configurations and to compare them with the afore-mentioned baselines, several state of the art performance measures have been considered. *Accuracy*, *Precision*, *Recall* and *F-measure* have been estimated. Since *Precision*, *Recall* and *F-measure* can be strongly influenced by the imbalanced distribution of the data over the entity types, the overall performance measures of a multi-class classification problem have been computed by two different types of average, *macro-average* and *micro-average* [80]. The *macro-averaged* measures give equal weight to each class, regardless of their frequencies. *Micro-averaged* weight each class with respect to its number of instances.

In addition to the well known metrics, we also estimated *Accuracy Contribution* which represents the number of correctly labeled entity mentions classified as a specific class y_T over the total number of instances:

$$Accuracy\_Contribution(y_T) = \frac{\#\text{ instances correctly classified as } y_T}{\#\text{ instances}} \quad (3)$$

In the following sections which report the computational results, given the highly imbalanced distribution of entity types, **Precision**, **Recall** and **F-measure** are shown in their *micro-averaged* version. It is important to note that the *micro-averaged* F-measure corresponds to the state of the art performance measure usually computed for evaluating NER systems, which is the Strong Typed Mention Match (**STMM**). Moreover, the *macro-averaged* F-measure has been also reported for the sake of completeness.

3.4. Validation Settings

Concerning the experimental evaluation, a *10-folds cross validation* has been performed. To compare L2AWE with the baseline models both on #Microposts2015 and #Microposts2016, $Precision_{micro}$, $Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$ have been used for comparing the types predicted by L2AWE with the real types available in the ground truth. In order to evaluate the

contribution of the different components of the input space, several configurations have been explored: the probability distribution in the source schema $X_P = P(\Omega_T, O_S)$, the embedded representation $X_E = E(\Omega_T)$ and the joint input space $X_{P \frown E} = P(\Omega_T, O_S) \frown E(\Omega_T)$. Concerning the models used to train L2AWE, i.e., Naïve Bayes (NB), Support Vector Machines (SVM), Decision Trees (DT), K-Nearest Neighbor (KNN), Bayesian Networks (BN), Multi-Layer Perceptron (MLP) and Multinomial Logistic Regression (MLR), no parameter optimization has been performed. The experiments have been conducted using default parameters of models implemented in Weka [5]. Since DT and NB learn on a discrete input space, a discretization policy is applied before the actual training phase. In particular, for J48 an entropy-based approach is adopted to find the optimal split point of a numeric attribute, while for NB a supervised discretization is used. In both cases, the discretization policies are directly implemented within the algorithm available in Weka.

4. Experimental Results

In this section, several computational experiments are presented in order to show the relevance of the proposed approach for the afore-mentioned datasets. In Section 4.1, the best configurations of L2AWE that consider the word embeddings feature space (X_E and $X_{P \frown E}$) have been studied. In Section 4.2, some selected results have been evaluated with respect to several baselines.

4.1. Word Embeddings Feature Space

In order to investigate the advantages of considering the word embeddings feature space, Table 5 reports the results in terms of Accuracy obtained on both datasets for all the chosen machine learning models, aggregation functions and pre-trained word embeddings models, highlighting the best results (**bold**) for each dataset. It is possible to highlight the following remarks based on Table 5:

[5] www.cs.waikato.ac.nz/ml/weka/

Table 5: Accuracy performance of L2AWE model considering word embeddings feature space.

		#Microposts2015				**#Microposts2016**			
		X_E		$X_{P^\frown E}$		X_E		$X_{P^\frown E}$	
		Wiki2Vec	GoogleNews (W2V)	Wiki2Vec	GoogleNews (W2V)	Wiki2Vec	GoogleNews (W2V)	Wiki2Vec	GoogleNews (W2V)
BN	mean	0.61	0.60	0.76	0.76	0.70	0.73	0.71	0.74
	max	0.58	0.58	0.73	0.74	0.71	0.74	0.72	0.75
	min	0.58	0.59	0.73	0.75	0.71	0.75	0.72	0.76
	first	0.60	0.59	0.74	0.73	0.69	0.72	0.70	0.73
DT	mean	0.51	0.54	0.75	0.74	0.70	0.71	0.77	0.78
	max	0.54	0.55	0.72	0.73	0.71	0.71	0.78	0.78
	min	0.52	0.55	0.75	0.74	0.68	0.69	0.78	0.77
	first	0.51	0.55	0.72	0.73	0.68	0.71	0.77	0.78
KNN	mean	0.58	0.60	0.75	0.79	0.80	0.82	0.81	0.83
	max	0.59	0.60	0.75	0.78	0.79	0.81	0.80	0.82
	min	0.59	0.61	0.75	0.79	0.78	0.81	0.80	0.83
	first	0.57	0.59	0.72	0.76	0.73	0.77	0.79	0.81
MLR	mean	0.58	0.54	0.75	0.75	0.83	0.78	0.77	0.78
	max	0.59	0.54	0.74	0.73	0.77	0.78	0.78	0.77
	min	0.59	0.53	0.76	0.74	0.81	0.78	0.77	0.77
	first	0.57	0.55	0.72	0.71	0.80	0.76	0.78	0.76
MLP	mean	0.63	0.64	0.82	**0.84**	0.85	0.85	**0.86**	0.85
	max	0.63	0.64	0.82	0.83	0.85	0.85	0.85	**0.86**
	min	0.63	0.64	0.82	0.83	0.85	0.85	**0.86**	0.85
	first	0.61	0.62	0.81	0.80	0.81	0.81	0.84	0.83
NB	mean	0.61	0.60	0.76	0.76	0.72	0.72	0.74	0.75
	max	0.58	0.58	0.73	0.74	0.73	0.74	0.75	0.77
	min	0.58	0.60	0.73	0.75	0.73	0.74	0.74	0.77
	first	0.60	0.59	0.74	0.74	0.72	0.73	0.74	0.75
SVM	mean	0.63	0.65	**0.84**	**0.85**	0.84	0.84	**0.86**	**0.86**
	max	0.63	0.65	**0.84**	**0.84**	0.85	0.85	**0.86**	**0.86**
	min	0.62	0.63	**0.84**	**0.84**	0.85	0.85	**0.86**	**0.86**
	first	0.61	0.64	0.82	0.81	0.81	0.81	0.83	0.83

- The joint input space $X_{P^\frown E}$ leads to better results as opposed to considering only the embedded representation (X_E) of the entity words. Although the probability distribution vector covers only 2% of the complete feature space, when combined with the embedded representation, it results in a great improvement in the classification. This means that considering only the word embeddings representation does not provide sufficient information for correctly mapping entity mentions.

- SVM and MLP proved to be the best models for dealing with the real-

valued vectors of word embeddings, while Decision Tree has been observed to exhibit the worst performance. This is likely due to the nature of the feature space, since the former models are best known for treating large real-valued vectors.

- While it is difficult to decide the best performing aggregation method among *mean, max* and *min*, it is clear that considering the representation of the *first* word of an entity mention (when dealing with multi-word mentions), a limited view of the underlying meaning of the entity mention has been observed, consequently leading to lower performance in terms of Accuracy.

- Finally, *GoogleNews(W2V)* pre-trained model performs better than the Wiki2Vec one. This can be due to the different nature and size of training data for these models.

Table 6: Class-Wise Accuracy Contribution on #Microposts2015 of L2AWE considering word embeddings feature space (X_E and $X_{P \frown E}$).

	MLP						**SVM**					
	mean		**max**		**min**		**mean**		**max**		**min**	
Entity Type	X_E	$X_{P \frown E}$	X_E	$X_{P \frown E}$	X_E	$X_{P \frown E}$	X_E	$X_{P \frown E}$	X_E	$X_{P \frown E}$	X_E	$X_{P \frown E}$
Character	0.48	0.54	0.48	0.48	0.42	0.54	0.54	0.48	**0.60**	0.48	0.48	0.48
Event	0.60	**1.14**	0.48	0.96	0.42	0.72	0.66	**1.14**	0.36	0.90	0.36	0.78
Location	21.81	27.23	21.93	27.11	22.17	27.23	22.11	**27.59**	22.41	27.47	22.05	27.59
Organization	13.86	18.73	14.04	18.92	13.98	19.22	13.61	**19.58**	14.28	19.34	12.83	19.22
Person	22.83	**29.94**	22.35	29.58	22.35	29.22	23.86	29.88	22.83	29.4	23.49	29.58
Product	3.19	4.76	3.25	4.52	3.25	4.28	3.43	**4.82**	3.19	**4.82**	3.19	4.58
Thing	0.90	1.63	1.08	**1.75**	1.08	1.63	1.02	1.63	0.96	1.63	1.02	1.57
Overall	63.67	83.98	63.61	83.31	63.67	82.83	65.24	**85.12**	64.64	84.04	63.43	83.8

In order to report a more compact representation of all the experimental results, in Tables 6 and 7 only the best performing configurations for L2AWE are shown. In particular, SVM and MLP have been selected as machine learning models, *mean*, *max* and *min* as aggregation methods and *GoogleNews(W2V)* pre-trained model as word embeddings representation. For each type, the highest result has been highlighted in bold.

Table 7: Class-Wise Accuracy Contribution on #Microposts2016 of L2AWE considering word embeddings feature space (X_E and $X_{P^\frown E}$).

	MLP						**SVM**					
	mean		**max**		**min**		**mean**		**max**		**min**	
Entity Type	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$
Character	**0.37**	**0.37**	0.30	0.33	0.30	**0.37**	0.33	0.30	0.27	0.33	0.33	0.30
Event	3.16	3.06	3.10	3.03	3.06	3.03	3.20	**3.26**	3.13	3.13	3.10	3.16
Location	34.13	34.37	34.53	34.67	34.43	34.63	34.47	34.87	**35.06**	**35.06**	35.03	35.00
Organization	14.85	14.65	14.99	**15.02**	14.62	14.59	13.79	14.82	13.89	14.72	14.19	14.69
Person	27.57	27.74	27.47	27.51	27.54	27.67	27.41	**27.91**	27.47	27.64	27.27	27.84
Product	2.93	3.20	3.06	3.13	2.86	3.10	2.83	2.96	3.20	**3.33**	2.80	2.90
Thing	1.70	1.70	1.83	1.83	**1.86**	1.70	1.73	1.83	1.80	1.76	1.80	1.80
Overall	84.72	85.08	85.28	85.51	84.68	85.08	83.75	85.95	84.82	**85.98**	84.52	85.68

As it is possible to perceive from both tables that Support Vector Machines generate better results by achieving the highest Overall Accuracy. While for #Microposts2015 the prevalence of the *mean* aggregation function is evident, the results on #Microposts2016 are less clear. However, by jointly considering the two datasets, the choice of using SVM as machine learning model and *mean* as aggregation function results as the best performing adaptation model in terms of Accuracy.

Table 8: Precision, Recall and F-Measure on #Microposts2015 and #Microposts2016 of L2AWE considering word embeddings feature space (X_E and $X_{P^\frown E}$).

		MLP						**SVM**					
		mean		**max**		**min**		**mean**		**max**		**min**	
		X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$
2015	**Precision**$_{micro}$	0.63	0.84	0.63	0.83	0.64	0.83	0.65	**0.85**	0.64	0.84	0.63	0.84
	Recall$_{micro}$	0.64	0.84	0.64	0.83	0.64	0.83	0.65	**0.85**	0.65	0.84	0.63	0.84
	F-measure$_{micro}$	0.63	0.84	0.63	0.83	0.63	0.83	0.65	**0.85**	0.64	0.84	0.63	0.84
	F-measure$_{macro}$	0.54	0.74	0.54	0.73	0.54	0.70	0.57	**0.75**	0.54	0.73	0.52	0.71
2016	**Precision**$_{micro}$	0.84	0.85	0.85	0.85	0.84	0.85	0.83	0.85	0.84	**0.86**	0.84	0.85
	Recall$_{micro}$	0.85	0.85	0.85	**0.86**	0.85	0.85	0.84	**0.86**	0.85	**0.86**	0.85	0.86
	F-measure$_{micro}$	0.84	0.85	0.85	0.85	0.84	0.85	0.83	**0.86**	0.84	**0.86**	0.84	0.85
	F-measure$_{macro}$	**0.75**	0.74	**0.75**	0.74	0.73	0.74	0.73	**0.75**	0.74	**0.75**	0.74	0.74

Since Accuracy has some limitations on evaluating the performance of a machine learning classifier, it is important to consider other measures for a wide and complete overview. As presented in the previous section, $Precision_{micro}$,

Table 9: Class-Wise Accuracy contribution on #Microposts2015 of L2AWE and baselines.

	Baselines				L2A							L2AWE	
Entity Type	**BL-D**	**BL-P1**	**BL-P2**	**CRF**	**BN** (X_P)	**NB** (X_P)	**MLR** (X_P)	**MLP** (X_P)	**SVM** (X_P)	**DT** (X_P)	**KNN** (X_P)	**SVM**$_{mean}$ (X_E)	**SVM**$_{mean}$ ($X_{P\frown E}$)
Character	0.00	**0.96**	0.48	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.54	0.48
Event	0.00	1.14	0.30	1.04	**1.20**	**1.20**	0.00	0.00	0.00	0.06	0.54	0.66	1.14
Location	24.76	26.69	20.72	5.41	26.45	26.45	26.20	**27.71**	26.27	27.59	27.41	22.11	27.59
Organization	11.63	11.63	11.02	2.89	15.24	15.30	17.71	17.59	17.65	17.47	17.11	13.61	**19.58**
Person	27.29	27.29	22.11	19.16	25.30	25.30	27.47	27.05	26.99	26.99	26.75	23.86	**29.88**
Product	2.35	2.35	1.57	1.26	1.02	1.02	2.05	2.71	1.99	2.11	2.35	3.43	**4.82**
Thing	0.66	0.66	1.57	0.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.63**	1.02
Overall	66.69	70.72	57.77	30.51	69.22	69.28	73.43	75.06	72.89	74.22	74.16	65.24	**85.12**

Table 10: Class-Wise Accuracy contribution on #Microposts2016 of L2AWE and baselines.

	Baselines				L2A							L2AWE	
Entity Type	**BL-D**	**BL-P1**	**BL-P2**	**CRF**	**BN** (X_P)	**NB** (X_P)	**MLR** (X_P)	**MLP** (X_P)	**SVM** (X_P)	**DT** (X_P)	**KNN** (X_P)	**SVM**$_{mean}$ (X_E)	**SVM**$_{mean}$ ($X_{P\frown E}$)
Character	0.00	**0.63**	0.27	0.16	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.33	0.30
Event	0.00	2.70	1.76	0.24	2.26	0.20	0.67	1.53	0.63	2.26	2.30	3.20	**3.26**
Location	29.77	32.77	27.34	1.62	32.93	34.43	32.13	33.97	31.90	33.63	33.83	34.47	**34.87**
Organization	8.72	8.72	8.23	2.36	11.92	11.29	13.22	11.92	13.55	13.32	13.39	13.79	**14.82**
Person	25.31	25.31	20.38	9.10	23.34	25.94	25.34	26.04	24.98	25.37	24.94	27.41	**27.91**
Product	2.00	2.00	1.23	**8.85**	1.47	0.13	1.80	1.67	1.96	1.90	1.96	2.83	2.96
Thing	0.50	0.50	**1.76**	0.89	0.10	0.13	0.00	0.03	0.00	0.13	0.30	1.73	**1.83**
Overall	66.30	72.63	60.97	23.23	72.03	72.13	73.16	75.16	73.03	76.66	76.72	83.75	**85.95**

$Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$ (Table 8) provide more details about the issues of multi-class classification and imbalanced class distribution as well. By looking at these measures, the superiority of SVM model is even more unequivocal. Moreover, using the *mean* aggregation function leads to the best results for both datasets, except for the $Precision_{micro}$ of #Microposts2016. These results have further motivated the choice of SVM-*mean* as the best performing model considering word embeddings representation.

Contextual Embeddings. With the aim of understanding the potential of more recent and outstanding contextual representation models, we compare the word embeddings model that showed the best performances, i.e., *GoogleNews(W2V)*, with the contextual embeddings model BERT (following the two approaches presented in Sec. 2.3). Given the input representation, we employ the best configuration framework, i.e. SVM as learning model and *mean* as aggregation method.

Table 11: Precision, Recall and F-Measure on #Microposts2015, #Microposts2016 and WNUT-17 datasets of L2AWE considering word and contextual embeddings feature space (X_E and $X_{P^\frown E}$).

		SVM$_{mean}$						**BERT**$_{NER}$
		GoogleNews(W2V)		**BERT**$_s$		**BERT**$_t$		
		X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	
2015	**Precision**$_{micro}$	0.65	**0.85**	0.66	0.73	0.73	0.78	0.30
	Recall$_{micro}$	0.65	**0.85**	0.67	0.73	0.74	0.79	0.33
	F-measure$_{micro}$	0.65	**0.85**	0.66	0.73	0.72	0.78	0.31
	F-measure$_{macro}$	0.57	**0.75**	0.50	0.57	0.50	0.56	0.31
2016	**Precision**$_{micro}$	0.83	**0.85**	0.72	0.78	0.76	0.81	0.43
	Recall$_{micro}$	0.84	**0.86**	0.72	0.78	0.78	0.82	0.47
	F-measure$_{micro}$	0.83	**0.86**	0.72	0.78	0.75	0.81	0.45
	F-measure$_{macro}$	0.73	**0.75**	0.59	0.63	0.50	0.60	0.45
WNUT-17	**Precision**$_{micro}$	0.75	**0.80**	0.62	0.70	0.66	0.73	0.41
	Recall$_{micro}$	0.76	**0.80**	0.63	0.71	0.68	0.74	0.34
	F-measure$_{micro}$	0.76	**0.80**	0.63	0.71	0.66	0.73	0.37
	F-measure$_{macro}$	0.65	**0.70**	0.53	0.60	0.53	0.62	0.37

Note that the aggregation method is irrelevant for BERT$_s$. Finally, we also compare the results with the fine-tuned BERT model (BERT$_{NER}$) on the considered dataset directly on the target ontology. We report in Table 11 the results in terms of $Precision_{micro}$, $Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$.

From the table, it can be easily noted that the L2AWE approach (SVM over the $X_{P^\frown E}$ joint input space using GoogleNews Word Embeddings) outperforms the BERT-based approaches on all the datasets (the results showed significant differences at 99.5% confidence level by paired t-test). It is also possible to notice the lower performance achieved by BERT$_{NER}$ with respect to the proposed model. These unfulfilling results can be attributed to the very small size of the training data. The impact of the reduced training instances size on the Deep Learning models training capabilities is a known issue, which has been also demonstrated for BERT fine-tuning [15]. Moreover, the recognition of named entities on informal textual contents (i.e., Twitter posts) is a very difficult task[6] and, when few training data are available, dedicated methods should be

[6] `https://github.com/huggingface/transformers/tree/master/examples/token-classification`

appointed for its resolution (such as T-NER).

Considering the representation models, as a general remark, we show that the joint input space $X_{P \frown E}$ permits to obtain higher performance for all the considered models with respect to the only embedded representation (X_E).

More importantly, the traditional word embeddings model *GoogleNews(W2V)* obtains higher performance in all the considered metrics with respect to the two variants of contextual representation model BERT. This is due to the fact that BERT is designed as a sentence-level representation model, and consequently, the extracted entity mentions representation is not static, and it is strictly dependent from the context. In the BERT_s representation model, we consider an entity as a document, obtaining static representations but without context: entity mentions are usually characterized by just one or two words (e.g., on #Microposts2016 there are 62% one-word named entities and 34% two-words named entities) and do not form an appropriate sentence by themselves. A different case is the BERT_t representation model, which extracts the word-level representation of the named entity considering the whole sentence as context. Thus, this representation model is able to include context, but the extracted representation is not static: entity mention representations vary widely depending on the context. This means that two sentences mentioning the same named entity will create two different output representations. For example, the representation of "Paris Hilton" will vary widely considering the following sentences "I love Paris Hilton" and "I've seen Paris Hilton on TV". This variance in representations will induce more noise in the subsequent machine learning models resulting in a lower performance with respect to classification based on word embedding representation.

A similar consideration has been drawn by Bommasani et al. [4], where the authors defined a single word as an unnatural input to the pretrained encoder. They have shown that the best performing word embeddings can be extracted by aggregating representations of the single word across multiple contexts. However, following this procedure is a probing operation as it would imply to collect 1M general-purpose sentences for each word for sampling a sufficient number of

contexts.

4.2. Baselines vs *L2AWE*

Following the above considerations, the next evaluation step refers to the comparison of L2AWE, considering the best model configuration (i.e., SVM as machine learning model, *mean* as aggregator method and *GoogleNews(W2V)* pre-trained model as word embeddings model), with the baselines and all the considered machine learning models on the probability distribution in the source schema (Tables 9 and 10).

It can be easily noticed that all the L2AWE configurations are able to achieve good adaptation performance in terms of global Accuracy. Lower Accuracy contributions by L2AWE can be observed for the entity types *Character* and *Thing*. This could be attributed to the low number of training instances available for *Character* (1.27% in #Microposts2015 and 0.99% in #Microposts2016 dataset) and for *Thing* (2.35% in #Microposts2015 and 2.30% in #Microposts2016 dataset) that does not allow any algorithm to provide remarkable contributions to the total Accuracy.

Except for few cases in #Microposts2015, the consideration of a joint input space $X_{P \frown E}$ leads to the best Accuracy results, further demonstrating that taking into account the probability distribution and the word embeddings representation is the winning strategy for the investigated adaptation problem. This behavior can be motivated by the fact that, while the embedded representation is capable of extracting underlying factors of the entity mentions, these are not sufficient on their own. They do, however, have a great advantage of enhancing the mere probability distribution vector, thereby, resulting in significantly better performance.

Analyzing the adaptation results of L2AWE from a qualitative point of view, it is interesting to highlight that the model is able to correctly re-classify the target types of entity mentions that have been misclassified, i.e., mentions which would have been cast to incorrect target types due to wrong predictions given by the T-NER system. For example, the entity mention “iPhone” was classified

Table 12: Precision, Recall and F-Measure on #Microposts2015 and #Microposts2016 of L2AWE and baselines.

		Baselines				L2A							L2AWE	
		BL-D	**BL-P1**	**BL-P2**	**CRF**	**BN** (X_P)	**NB** (X_P)	**MLR** (X_P)	**MLP** (X_P)	**SVM** (X_P)	**DT** (X_P)	**KNN** (X_P)	**SVM**$_{mean}$ (X_E)	**SVM**$_{mean}$ ($X_{P^\frown E}$)
2015	**Precision**$_{micro}$	0.73	0.77	0.78	0.48	0.75	0.75	0.69	0.70	0.69	0.71	0.71	0.65	**0.85**
	Recall$_{micro}$	0.67	0.71	0.58	0.31	0.69	0.69	0.73	0.75	0.73	0.74	0.74	0.65	**0.85**
	F-measure$_{micro}$	0.68	0.72	0.65	0.27	0.70	0.70	0.71	0.73	0.70	0.72	0.72	0.65	**0.85**
	F-measure$_{macro}$	0.38	0.62	0.46	0.34	0.38	0.39	0.38	0.40	0.38	0.42	0.43	0.57	**0.75**
2016	**Precision**$_{micro}$	0.72	0.78	0.79	0.05	0.73	0.72	0.71	0.72	0.71	0.75	0.75	0.84	**0.86**
	Recall$_{micro}$	0.66	0.73	0.61	0.23	0.72	0.72	0.73	0.75	0.73	0.77	0.77	0.85	**0.86**
	F-measure$_{micro}$	0.68	0.74	0.68	0.08	0.72	0.72	0.71	0.73	0.71	0.75	0.75	0.84	**0.86**
	F-measure$_{macro}$	0.37	0.61	0.49	0.09	0.44	0.44	0.42	0.45	0.42	0.50	0.51	0.74	**0.75**

as a *Company* by T-NER (which would lead to the target type *Organization* using manual mappings), while L2AWE correctly re-classifies it as a *Product*. As another example, "Ron Weasley" (a character in Harry Potter movies/books) was misclassified as *Band* by T-NER, while L2AWE correctly re-classifies it as a *Character*. In the latter case, L2AWE was able to assign the correct type among the two possible types defined according to fork mappings. Although there are very few instances in the training sets for the target types *Character* and *Event* and the performance of L2AWE is not very high in terms of Accuracy contribution, the proposed approach seems to be promising.

In Table 12, the performance of the proposed approach with respect to different input space configurations are compared with the baselines in terms of $Precision_{micro}$, $Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$. As expected, the deterministic baseline (BL-D) achieves good performance in terms of $Precision_{micro}$, but low results of $Recall_{micro}$. In fact, BL-D is accurate when labeling mentions thanks to the deterministic mapping, at the expense of $Recall_{micro}$. Also in this case, it can be easily noted that using SVM over the joint input space $X_{P^\frown E}$ significantly outperforms the baselines and the other L2AWE configurations both for the #Microposts2015 and #Microposts2016 datasets (the results showed significant differences at 99.5% confidence level by paired t-test).

These experiments show that the proposed approach provides significant

results with respect to all the considered performance measures and obtains a balanced contribution of $Precision_{micro}$ and $Recall_{micro}$. Moreover, L2AWE drastically improves the $Recall_{micro}$ measure. This is likely due to its ability to learn how to map the initial hypothesis given by T-NER to a new target type, adapting type mentions that were previously misclassified.

For a comparison with the most recent approach of the state art, we report the class-wise Accuracy, $Precision_{micro}$, $Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$ achieved by **CMDA-NER** and L2AWE in Tables 13 and 14. Similar to the previous findings, a low accuracy contributions can be observed for the entity types *Character* and *Thing* due to their scarce presence in the training set. It can be also noticed that the results on the #Microposts2015 dataset are on average lower than the #Microposts206 one. This is likely due to the lower number of instances and consequently the reduced size of the training set. As a final consideration, L2AWE is able to obtain significantly higher results in terms of all the considered performance measure with respect to the state of the art CDMA-NER model (the results showed significant differences at 99.5% confidence level by paired t-test).

Similar results have been obtained on the W-NUT17 dataset, where L2AWE computed on the joint input space $X_{P \frown E}$ ($F\text{-}measure_{micro}$ equals to 0.79) outperforms the L2AWE model computed on the embedding input space X_E ($F\text{-}measure_{micro}$ equals to 0.75) and the L2A model exploiting SVM as learning model ($F\text{-}measure_{micro}$ equals to 0.69). The results showed significant differences at 99.5% confidence level by paired t-test.

Beyond the classic performance evaluation measures, several *capabilities* have been measured on #Microposts2015 and #Microposts2016 datasets with respect to the three issues stated in Section 2, i.e., mention misclassification, type uncertainty and fork mapping.

These capability measures are described as follows:

1. **Mention Misclassifications Correctly Mapped (MMCM)**: this mea-

Table 13: Class-Wise Accuracy contribution on #Microposts2015 of L2AWE and CDMA-NER models.

	2015		2016	
Entity Type	**CDMA-NER**	**L2AWE**	**CDMA-NER**	**L2AWE**
Character	0.32	0.32	0.00	0.31
Event	0.11	0.84	3.09	3.87
Location	26.05	30.91	37.54	41.05
Organization	10.44	17.72	9.69	13.46
Person	24.58	31.33	21.52	25.65
Product	1.90	4.85	0.99	2.46
Thing	0.00	1.79	0.21	1.83
Overall	63.40	87.76	73.04	88.64

Table 14: Precision, Recall and F-Measure on #Microposts2015 and #Microposts2016 of L2AWE and CDMA-NER models.

		CMDA-NER	**L2AWE**
2015	**Precision**$_{micro}$	0.61	0.88
	Recall$_{micro}$	0.38	0.74
	F-measure$_{micro}$	0.62	0.88
	F-measure$_{macro}$	0.4	0.76
2016	**Precision**$_{micro}$	0.72	0.88
	Recall$_{micro}$	0.46	0.77
	F-measure$_{micro}$	0.72	0.88
	F-measure$_{macro}$	0.48	0.78

sure indicates the percentage of entity mentions that T-NER has incorrectly classified and a model is able to correctly map according to the target entity types. For the considered experimental set, in the training sets for #Microposts2015 and #Microposts2016, T-NER has incorrectly classified 524 and 921 entity mentions respectively.

2. **Type Uncertainty Correctly Mapped (TUCM)**: this measure denotes the percentage of uncertain entity mentions that a model correctly maps to entity types in the target schema. To compute this measure, a mention t_i has been defined as an *uncertain mention* when it has a low gap between probability distribution over different types. More formally, t_i is considered as *uncertain* if:

$$P(t_i, y_{T_j}) - P(t_i, y_{T_k}) \leq \alpha_U \quad \forall j \neq k \tag{4}$$

where α_U is a parameter that has been experimentally determined as equal to 0.2. The number of mentions that have been recognized as *uncertain* in the training sets are 59 for #Microposts2015 and 109 for #Microposts2016.

3. **Fork Mappings Correctly Resolved (FMCR)**: this measure represents the percentage of mentions of a type defined as fork mappings (i.e., *Event*, *Location*, and *Character*) that have been correctly classified by a model. According to the training sets, the number of mentions that fall under this category is 50 for #Microposts2015 and 145 for #Microposts2016.

The results are shown in Table 15, where the most successful models have been considered. Since the capabilities depends on the performance of the named entity recognition system adopted (two NER systems could generate different mention misclassification or entity types that in one case are uncertain and in another are considered certain) for a fair and significant comparison we reported only those results obtained by the adaptation systems that share the same NER system. Results are not reported for the CRF model, however, since

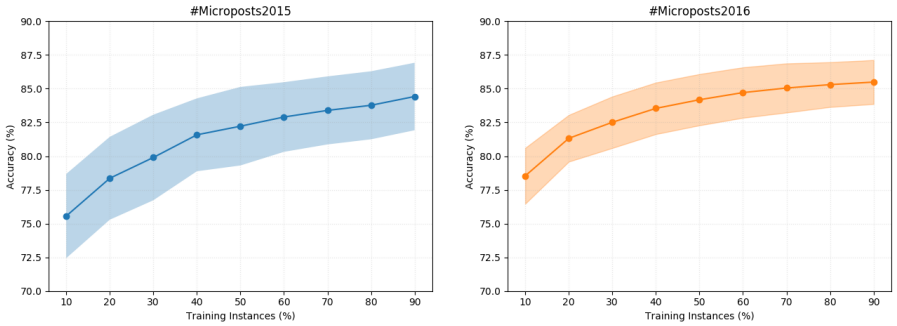
they do not perform any adaptation and are trained and tested directly into the target domain.

Table 15: Capabilities performance on #Microposts2015 of L2AWE and baselines.

		Baselines		**L2A**				**L2AWE**
		BL-P1	**BL-P2**	**MLP** (X_P)	**SVM** (X_P)	**DT** (X_P)	**KNN** (X_P)	**SVM**$_{mean}$ ($X_{P^\frown E}$)
2015	**MMCM**	2.67	19.00	35.88	25.57	36.07	34.92	**64.50**
	TUCM	15.25	27.29	57.63	42.37	45.76	45.76	**76.27**
	FMCR	25.96	22.10	26.19	25.00	28.57	40.48	**63.10**
2016	**MMCM**	4.67	18.26	32.03	24.00	40.50	39.52	**64.71**
	TUCM	14.68	24.53	53.21	31.19	56.88	52.29	**76.15**
	FMCR	34.00	32.17	48.67	28.52	57.41	58.94	**78.33**

Moreover, among the baselines, the deterministic one (BL-D) has been discarded because its capability performance always amounts to zero scores since it mimics a fixed manual mapping a priori defined. This means that if an entity mention is incorrectly classified in the source schema, it will always be mapped to the corresponding (incorrect) entity type in the target schema. For instance, if the mention "Paris" is incorrectly classified by T-NER as *Movie* (whereas its correct type is *Location*), BL-D will map Paris to *Product*, providing no improvement for the MMCM capability. The same reasoning is applicable also for TUCM and FMCR capabilities.

The first consideration that can be derived from the capabilities performance results is that, once again, SVM performs considerably better for all the considered measures over the joint space. Secondly, it can also be observed that, in most cases, the results on #Microposts2015 set are worse than the ones on #Microposts2016. This is due to the fact that the number of entity mentions available for training L2AWE in the #Microposts2016 is about twice as much as in #Microposts2015 (as stated in Section 3). In other words, the higher the number of mentions that L2AWE can use to learn the correct mappings, the better the capabilities will be. Furthermore, in order to better understand the poor results of FMCR, a detailed investigation has been conducted on the predictions of the machine learning models. For #Microposts2015, the number of



(a) (b)

Figure 3: Cross-validation Learning curves. The curves are plotted with the mean scores, while variability during cross-validation is shown with the shaded areas that represent a standard deviation above and below the mean for all cross-validations.

mentions involved in a fork mapping is 50 (21 for the entity type *Character* and 29 for the entity type *Event*). Given the low frequency of these entity types in the dataset (note that the entity types *Location, Person* and *Organization* are composed of more than 400 instances each), it is very difficult for a machine learning algorithm to learn how to recognize their presence. On the other hand, in #Microposts2016, there are 145 entities involved in a fork mapping: 30 entities are *Character* and 115 *Event*. The results in terms of FMCR are promising but, following the previous intuition, the performance increase is mainly due to correctly classified instances for the entity type *Event*, while only a few instances of the type *Character* have been correctly identified.

In order to demonstrate the ability of the model to obtain satisfying results when few training instances are available (i.e., data annotated with the target entity types), we estimated the validation learning curves on the considered datasets reporting the results in Figure 3. It is possible to notice that, from the 30% and 20% of the training instances for the #Microposts2015 and #Microposts2016 datasets respectively (which corresponds to approximately 500 and 600 instances), L2AWE is able to achieve an accuracy higher than 80% still

Table 16: Type Distribution (%) according to the Ritter Ontology (O_S).

	500-news	**Reuters-128**
Band	2.59	1.36
Company	6.48	15.49
Facility	3.11	1.9
Geo-Loc	24.35	42.93
Movie	0.52	2.45
Other	14.77	14.40
Person	39.12	17.66
Product	1.81	2.45
Sportsteam	5.18	1.36
TVshow	2.07	0.00

Table 17: Type Distribution (%) according to the News Ontology (O_T)

Entity Type	**500-news**	**Reuters-128**
Event	1.30	0.00
Organization	38.34	39.13
Person	38.08	19.84
Place	18.91	39.40
Species	0.52	0.00
Thing*	1.04	0.82
Work	1.81	0.82

maintaining a constant standard deviation of the performance measure.

4.3. Additional Experiments

In order to demonstrate the generalization abilities of the proposed approach to other domains, we performed additional experiments considering two benchmark corpora composed of news based documents. These corpora differ from the baseline datasets (as described in Section 3.1) primarily in their domain, i.e., news domain as opposed to the microblogging domain. Documents in the news corpora are relatively longer, more formal, often very factual, and have an enhanced descriptive textual format than microblogging posts, which are generally shorter, informal, and often opinionated. Given the dataset size, we performed a *5-folds cross validation*.

Dataset. The two news datasets that we additionally considered have been published under the N^3 datasets collection [69]. These datasets, namely, a.) *500-news* and b.) *Reuters-128* consist of 500 and 128 documents, respectively, where each document is annotated with entity mentions and their corresponding DBpedia URIs. The 500-news dataset contains generic English news snippets, while Reuters-128 contains economic news articles. A total of 1,000 entity mentions are present in 500-news dataset with a uniform distribution of 2 entity mentions per document, whereas the Reuters-128 dataset has a total of 880 entity mentions, with an average of 3 entity mentions per document.

Differently from the previous benchmark datasets specifically created for Named Entity Recognition and Classification task, the entity type was not given explicitly in these two news datasets[7]. For this reason, we extracted the entity type information from DBpedia by querying its public SPARQL endpoint with the named entities URIs by taking only a selected pool of types of interest. This process resulted in the extraction of the entity type for 523 entity mentions in the *500-news* dataset and for 648 entity mentions for the *Reuters-128* dataset. The retrieval of all the entity mentions in the datasets was not possible due to not available or inconsistent DBpedia URIs (some mentions refer to entities not in DBpedia, while some URIs are not existent anymore). The final target News Schema is, thus, composed of the pairwise disjoint types: *Person, Place, Organization, Work, Event, Species, Thing**. While the main classes are still present compared to the Microposts and Ritter Ontology (*Person, Place, Organization, Other*), we can observe few differences in the less populated classes, such as *Species* and *Work*. Also, the target News Schema represents a complete traversal of the DBpedia Ontology at a medium level of abstraction, with Thing* being used for all the entities that do not have any of the other more specific types. As a consequence, we observe that Thing* in the target News Schema has a different interpretation and extension than *Other* in the Ritter Ontology (which conceptually includes also species and works) and *Thing* in

[7]It is assumed that the entity types can be fetched directly from DBpedia.

the DBpedia Ontology (where it indicates the superclass of any other class and thus the type of any entity consistently with OWL2 semantics[8]).

In order to proceed with the classification through the L2AWE models, we replicated the same process described in Section 3.1 for obtaining the input data. We used the benchmark NER system T-NER for deriving $P(\Omega_T, O_S)$. Consequently, we retained only the named entities in the datasets which have also been extracted by T-NER.

As a result, the datasets for Reuters-128 and 500-news are composed of 368 and 386 instances respectively. Tables 16 and 17 reports the distribution of the entity types in the derived datasets, respectively referring to the Ritter Schema (O_S) and News Schema (O_T).

As with the previously investigated datasets, the distribution is strongly imbalanced. *Person* and *Geo-Location* (or *Place*) and *Organization* are the most common types in both ontologies. It is interesting to notice the high percentage of mention annotated as *Company* in the Ritter Ontology for the Reuters-128 dataset, which is an expected output given the economic news domain. It should also be mentioned that the Reuters-128 dataset does not contain instances of two classes (i.e., *Event*,*Species*), due to the limited number of entity mentions.

Results. In Table 18, we show the results of the best performing L2AWE model i.e., SVM as learning model, *mean* as aggregation method and *GoogleNews(W2V)* as representation method, compared to the approaches based on BERT (following the three approaches considered in Sec. 4.1). The results are reported in terms of $Precision_{micro}$, $Recall_{micro}$, $F\text{-}measure_{micro}$ and $F\text{-}measure_{macro}$.

Findings are very similar to the ones drawn for the Social Media datasets (Table 11). It is clear how the L2AWE model outperforms the BERT-based model (results are significantly different at 99.5% confidence level computed with paired t-test). This further demonstrates the positive impact of the proposed approach for the task of adapting named entity types to a given target schema.

[8] `https://www.w3.org/TR/owl2-overview/`

Table 18: Precision, Recall and F-Measure on news-500 and Reuters-128 datasets of L2AWE considering word and contextual embeddings feature space (X_E and $X_{P^\frown E}$).

		SVM$_{mean}$						**BERT**$_{NER}$
		GoogleNews(W2V)		**BERT**$_s$		**BERT**$_t$		
		X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	X_E	$X_{P^\frown E}$	
news-500	**Precision**$_{micro}$	0.83	**0.84**	0.73	0.78	0.78	0.78	0.26
	Recall$_{micro}$	0.84	**0.85**	0.73	0.78	0.77	0.78	0.29
	F-measure$_{micro}$	0.83	**0.84**	0.73	0.78	0.77	0.78	0.27
	F-measure$_{macro}$	0.53	**0.54**	0.50	0.52	0.53	0.53	0.27
Reuters-128	**Precision**$_{micro}$	**0.89**	**0.89**	0.85	0.85	0.81	0.83	0.13
	Recall$_{micro}$	0.89	**0.90**	0.85	0.85	0.80	0.83	0.14
	F-measure$_{micro}$	**0.89**	**0.89**	0.85	0.85	0.81	0.83	0.14
	F-measure$_{macro}$	**0.73**	**0.73**	0.71	0.71	0.68	0.69	0.14

Moreover, results show that L2AWE is not only suitable for domains other than Social Media, but that it is also an effective methodology. The main difference that can be noticed with the Social Media datasets is the importance of the probability distribution extracted by T-NER. Indeed, results are not significantly different at 99.5% confidence level by paired t-test. Thus, it could be inferred from these findings that the probability distribution obtained by T-NER is not helpful in improving performance on news datasets. However, this issue is due to T-NER source domain, i.e., Social Media. Indeed, this NER system is specifically conceived for dealing with user-generated contents. This means that the output probability distribution computed on news data will be less precise and useful than the ones obtained on Social Media domain. BERT-based models obtained lower performance than L2AWE, the reasons could be the same presented in Section 4.1: limited training instance size, the limitation of using a sentence-level representation model, and non-static and unstable entity-level representations.

5. Related Works

This section presents a detailed account of related works in the field of domain adaptation for named entity recognition. In particular, we focus on the research studies as well as the research gaps that have been found in the area of

NER domain adaptation for informal text formats such as Twitter microposts.

Primarily, in the context of Semantic Web as well as application areas such as large information systems, [14] have proposed the use of machine learning techniques (multi-stage learning) to semi-automatically create semantic mappings between entity types (concepts) of two classification schemas using some state of the art semantic similarity measures. It is also possible to find machine learning methods applied to Ontology Matching [19, 72, 18] in the literature. Textual annotation and re-classification statistics have been also proposed in [2] in order to semantically interpret class-to-class ontology mappings. However, it is important to note that these approaches have been based on collecting feedback on class-to-class mappings in order to improve ontology alignments. Moreover, in [6] the authors suggested that the performance of similarity measures (based on different textual features) for ontology matching is dependent on the type of ontologies being investigated. [29] provide a thorough review of ontology engineering with respect to mapping approaches seen in the state of the art, specifically, for the field of biomedicine.

During the last years, the use of embedding representation of word and sentences has been increasingly seen in the NLP community. Word Embeddings [75, 53, 61] are the standard component of most NLP approaches due to their ability to capture syntactic and semantic information of words from large scale unlabeled text. These approaches have been further generalized to a higher granularity level, such as sentence embeddings [35, 49, 41]. More recently, contextual embeddings trained on large corpora [30, 62, 13] took over advancing the state of the art for several major NLP tasks. In particular, BERT (Bidirectional Encoder Representations from Transformers) [13] demonstrated impressive results on eleven NLP tasks [76] paving the way to a large number of its extensions [48, 40, 70].

Concerning the considered problem, in [60], the authors have studied the use of neural word embeddings to obtain a high performing named entity recognition system, while in [82] the representation of words as vectors in the semantic space followed by the use of string similarity metrics for ontology mapping

has been investigated. More recently, in [36] an unsupervised approach named *DeepAlignment* has been presented. This approach makes use of pre-trained word vectors of entity types in order to compute semantic similarity across entity types of different ontologies when performing ontology matching. Although this paper focuses on the use of a flat classification schema, we drew inspiration and realised research gaps in the literature in terms of class-to-class mappings across diverse ontologies, particularly for the task of Named Entity Recognition, from the works as cited above.

When it comes to mapping entity types of classification schemas used specifically by NER systems, a manual (and subjective) approach has been used for establishing such mappings between entity types of two different NER schemas. In particular, [67] have used manual mappings as a way to bridge the gap between NER systems using different schemas for tasks such as comparison and/or integration of NER systems. When many-to-one mappings are used, i.e., when one source type is mapped to at most one target type, and when the source classification is reasonably accurate, manual mappings may achieve a good performance. However, in contexts such as microblogging platforms, when different generic schemas are used for classification (for a given domain) and pre-trained NER systems are affected by the dynamics of new upcoming entities, these mappings have several limitations, as have been discussed in the previous sections.

Furthermore, the problem of adapting NER models had been investigated in the context of formal texts in [9, 10] in the last decade. Arnold et al. [1] proposed an approach for domain adaptation to learn a domain-independent NER base model, which can be adapted to specific domains. [45] present a very recent investigation for cross-domain NER (pre-trained on a source domain such as *online news*) by use of an *instance transfer based approach with enhanced Recurrent Neural Network (RNN)* for a target domain of *politics* and study the application of such a model for Q&A systems. While these models aimed to adapt different domain-independent classification schemas, several state of the art approaches have been introduced to adapt NER models trained on specific schemas to adapt to new domains since then. A bit further from L2AWE, there

are approaches that work on mapping a domain-specific source schema to a completely different target one. In [8] the authors presented a NER rule-based language that is further used for building domain-specific rule-based NER systems. However, rule-based approaches require additional time for developing and tuning the defined rules according to each domain or type of dataset. A more recent study presented in [38] portrays the use of distributed word representations to adapt named entity recognition models learned in one domain (such as Sports) to other domains (such as Finance, and Medicine) by modeling domain-specific differences of language semantics. Moreover, investigations presented in [44] show a bidirectional LSTM model augmented with a CRF layer for cross-domain NER, while neural methods for transferring well-learned knowledge in the source domain to target domain have been studied in [7].

Specific domains, such as eHealth, have witnessed a growth of research contributions. Considering the scarcity of domain-specific training data as well as the necessity of adopting peculiar classification schemas, several named entity recognition systems based on neural architectures and transfer learning paradigms have been proposed. A recent transfer-learning based approach has been proposed in [63] for adapting a NER system from a source (medicine) domain to a target domain by using a linear chain CRF. In particular, the authors proposed the use of a transfer learning approach to adapt a CRF trained on a source domain to a target domain by learning the source and target correlations and fine-tuning the model to domain-specific patterns. Lee et al. [42] propose a transfer learning approach to perform named entity recognition in the form of de-identification of a given patient's protected health information. The proposed approach transfers a pre-trained LSTM-CRF model from a large labeled dataset to a smaller dataset. A similar approach has been explored by Giorgi et al. [26] where a bidirectional LSTM model trained on large and noisy biomedical datasets has been used to perform named entity recognition on small and domain-specific gold standard datasets based on four entity types. For the study of *cross-specialty medical NER*, [77] propose the use of a double transfer learning framework which leverages a Bi-LSTM network for learning text

representations and performing feature representation transfer. The Bi-LSTM network is coupled with the use of CRF models for sequence labeling in source and target domains separately to avoid annotation efforts. Additionally, Bhatia et al. presented a framework in [3] for performing named entity recognition for domains with low resources such as medicinal texts. They proposed a tunable transfer learning architecture to counter the data scarcity problem, coupled with a parameter sharing approach to transfer overlapped representation from the source to the target domain.

While the above-mentioned approaches are focused on formal text, few works [50, 22] have addressed the adaptation problem in an informal context such as when dealing with microblogging formats, where the language used by the users can vary significantly (use of abbreviations, slangs, punctuations and wrong use of capital letters/words) and new entities emerge frequently. In this work, we extended the former investigations (as seen in [50, 22]) by introducing word embeddings for adapting a pre-trained NER system to novel generic classification schemas. To the best of our knowledge, this work is one of the primary contributions for informal textual contents (i.e., Twitter microposts) where different distributional representations with several aggregation functions have been evaluated for adaptation purposes.

6. Conclusion and Future Works

This paper presents an approach named LearningToAdapt with Word Embeddings (L2AWE), which aims at adapting a NER system trained on a source generic classification schema to a given target one by exploiting a rich semantic input space. From the experimental evaluation, it is possible to conclude that the use of word embeddings can strongly improve the performance, both in terms of traditional measures and capabilities, on the task of adapting trained NER systems to new schemas. The best adaptation abilities have been obtained by jointly considering word embeddings of the entity mentions and the probability distribution over the source entity types as the input space, and by using Sup-

port Vector Machines as the machine learning classifier. Remarkably, we found word embeddings based on Word2vec more useful for this task than contextual word embeddings based on BERT, probably because word-dependent representations provide more consistent signals to the classifier than (highly specialized) sentence-dependent representations when limited training data are used.

As future work, we highlight the possibility to investigate some additional and promising models for training L2AWE, such as Random Forest. This step would allow us to understand the robustness of the model also adopting ensemble learning strategies. Another important issue that could be studied as future work regards the case of new or rare entity mentions that cannot be represented with word embeddings since they are not included in their vocabulary. This is especially crucial in the context of social media where users generate new words and acronyms everyday. One solution could be to address the problem by a combined approach of character-level and word-level embeddings. It would be also interesting to specifically train word embeddings models on corpora where the ontology type class is provided for the entity mentions, for example by exploiting the Wikipedia pages structure. Moreover, additional experiments over different and more domain-specific corpora (e.g. medical) are planned for future investigations.

Acknowledgements

This work has been partially supported by PON I&R 2014-20, with the grant for research project "SmartCal", CUP B48I15000180008.

References

[1] Arnold, A., Nallapati, R., and Cohen, W. W. (2008). Exploiting feature hierarchy for transfer learning in named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 245–253.

[2] Atencia, M., Borgida, A., Euzenat, J., Ghidini, C., and Serafini, L. (2012). A formal semantics for weighted ontology mappings. In *Proceedings of the 11th International Semantic Web Conference*, pages 17–33.

[3] Bhatia, P., Arumae, K., and Celikkaya, E. B. (2019). Dynamic transfer learning for named entity recognition. In *International Workshop on Health Intelligence*, pages 69–81.

[4] Bommasani, R., Davis, K., and Cardie, C. (2020). Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781.

[5] Bouarroudj, W., Boufaida, Z., and Bellatreche, L. (2019). Welink: A named entity disambiguation approach for a qas over knowledge bases. In *Proceedings of the International Conference on Flexible Query Answering Systems*, pages 85–97.

[6] Cheatham, M. and Hitzler, P. (2013). String similarity metrics for ontology alignment. In *Proceedings of the 12th International Semantic Web Conference*, pages 294–309.

[7] Chen, L. and Moschitti, A. (2019). Transfer learning for sequence labeling using source model and target data. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 6260–6267.

[8] Chiticariu, L., Krishnamurthy, R., Li, Y., Reiss, F., and Vaithyanathan, S. (2010). Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1012.

[9] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175.

[10] Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263.

[11] De Boom, C., Van Canneyt, S., Demeester, T., and Dhoedt, B. (2016). Learning representations for tweets through word embeddings. In *Proceedings of Benelearn*, page 3.

[12] Derczynski, L., Nichols, E., van Erp, M., and Limsopatham, N. (2017). Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147. Association for Computational Linguistics.

[13] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). 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, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

[14] Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2004). Ontology matching: A machine learning approach. In *Handbook on ontologies*, pages 385–403. Springer.

[15] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

[16] dos Santos, C. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 69–78.

[17] Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., and Wudali, R. (2010). Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer.

[18] Duan, S., Fokoue, A., and Srinivas, K. (2010). One size does not fit all: Customizing ontology alignment using user feedback. In *Proceedings of the 9th International Semantic Web Conference*, pages 177–192.

[19] Eckert, K., Meilicke, C., and Stuckenschmidt, H. (2009). Improving ontology matching using meta-level learning. In *Proceedings of the 6th European Semantic Web Conference*, pages 158–172.

[20] Fabregat, H., Duque, A., Martinez-Romo, J., and Araujo, L. (2019). De-identification through named entity recognition for medical document anonymization. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing*, pages 663–670.

[21] Fensel, D. (2001). Ontologies. In *Ontologies*, pages 11–18. Springer.

[22] Fersini, E., Manchanda, P., Messina, E., Nozza, D., and Palmonari, M. (2018). Adapting named entity types to new ontologies in a microblogging environment. In *Proceedings of the 31st International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, pages 783–795.

[23] Fersini, E., Messina, E., Felici, G., and Roth, D. (2014). Soft-constrained inference for named entity recognition. *Information Processing & Management*, 50(5):807–819.

[24] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

[25] Finkel, J. R. and Manning, C. D. (2009). Nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 141–150.

[26] Giorgi, J. M. and Bader, G. D. (2018). Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*, 34(23):4087–4094.

[27] Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., and Sheth, A. (2009). Context and domain knowledge enhanced entity spotting in informal text. In *Proceedings of the 8th International Semantic Web Conference*, pages 260–276.

[28] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[29] Harrow, I., Balakrishnan, R., Jimenez-Ruiz, E., Jupp, S., Lomax, J., Reed, J., Romacker, M., Senger, C., Splendiani, A., Wilson, J., et al. (2019). Ontology mapping for semantically enabled applications. *Drug discovery today*.

[30] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics.

[31] Jansche, M. and Abney, S. (2002). Information extraction from voicemail transcripts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 320–327.

[32] K, S. and Thilagam, P. S. (2019). Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. *Information Processing & Management*, 56(6):102059.

[33] Karimi, S., Zobel, J., and Scholer, F. (2012). Quantifying the impact of concept recognition on biomedical information retrieval. *Information Processing & Management*, 48(1):94 – 106.

[34] Kejriwal, M., Szekely, P. A., and Knoblock, C. A. (2018). Investigative knowledge discovery for combating illicit activities. *IEEE Intelligent Systems*, 33(1):53–63.

[35] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302. Curran Associates, Inc.

[36] Kolyvakis, P., Kalousis, A., and Kiritsis, D. (2018). Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 787–798.

[37] Krötzsch, M. and Weikum, G. (2016). Journal of web semantics: Special issue on knowledge graphs.

[38] Kulkarni, V., Mehdad, Y., and Chevalier, T. (2016). Domain adaptation for named entity recognition in online media with word embeddings. *CoRR*, abs/1612.00148.

[39] Lafferty, J. D., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

[40] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A Lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

[41] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1188–1196.

[42] Lee, J. Y., Dernoncourt, F., and Szolovits, P. (2018). Transfer learning for named-entity recognition with neural networks. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 4470–4473.

[43] Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., and Lee, B.-S. (2012). Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730.

[44] Lin, B. Y. and Lu, W. (2018). Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods on Natural Language Processing*, pages 2012–2022.

[45] Liu, C., Fan, C., Wang, Z., and Sun, Y. (2020). An instance transfer-based approach using enhanced recurrent neural network for domain named entity recognition. *IEEE Access*, 8:45263–45270.

[46] Liu, C., Li, J., Liu, Y., Du, J., Tang, B., and Xu, R. (2019a). Named entity recognition in clinical text based on Capsule-LSTM for privacy protection. In *Proceedings of the 8th International Artificial Intelligence and Mobile Services Conference*, pages 166–178.

[47] Liu, X. and Zhou, M. (2013). Two-stage NER for tweets with clustering. *Information Processing & Management*, 49(1):264 – 273.

[48] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

[49] Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

[50] Manchanda, P., Fersini, E., Palmonari, M., Nozza, D., and Messina, E. (2017). Towards adaptation of named entity classification. In *Proceedings of the Symposium on Applied Computing*, pages 155–157.

[51] McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

In *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191.

[52] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8.

[53] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Workshop Track Proceedings of the 1st International Conference on Learning Representations*.

[54] Minkov, E., Wang, R. C., and Cohen, W. W. (2005). Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 443–450.

[55] Mollá, D., van Zaanen, M., and Smith, D. (2006). Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58.

[56] Nguyen, D. B., Abujabal, A., Tran, N. K., Theobald, M., and Weikum, G. (2017). Query-driven on-the-fly knowledge base construction. *Proceedings of the VLDB Endowment*, 11(1):66–79.

[57] Nozza, D., Bianchi, F., and Hovy, D. (2020). What the [mask]? making sense of language-specific bert models.

[58] Nozza, D., Ristagno, F., Palmonari, M., Fersini, E., Manchanda, P., and Messina, E. (2017). TWINE: A real-time system for TWeet analysis via INformation extraction. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28.

[59] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[60] Passos, A., Kumar, V., and McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the 18th Conference on Computational Natural Language Learning*, pages 78–86.

[61] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, volume 14, pages 1532–1543.

[62] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). 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)*, pages 2227–2237. Association for Computational Linguistics.

[63] Qu, L., Ferraro, G., Zhou, L., Hou, W., and Baldwin, T. (2016). Named entity recognition for novel types by transfer learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 899–905.

[64] Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256.

[65] Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534.

[66] Rizzo, G., Basave, A. E. C., Pereira, B., and Varga, A. (2015). Making sense of microposts (#Microposts2015) named entity recognition and linking (NEEL) challenge. In *Proceedings of the 5th Workshop on Making Sense of Microposts co-located with the 24th International World Wide Web Conference*, volume 1395, pages 44–53.

[67] Rizzo, G. and Troncy, R. (2012). NERD: A framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–76.

[68] Rizzo, G., van Erp, M., Plu, J., and Troncy, R. (2016). Making sense of microposts (#Microposts2016) named entity recognition and linking (NEEL) challenge. In *Proceedings of the 6th Workshop on Making Sense of Microposts co-located with the 25th International World Wide Web Conference*, volume 1691, pages 50–59.

[69] Röder, M., Usbeck, R., Hellmann, S., Gerber, D., and Both, A. (2014). N^3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *LREC*, pages 3529–3533.

[70] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[71] Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 107–110.

[72] Shi, F., Li, J., Tang, J., Xie, G., and Li, H. (2009). Actively learning ontology matching via user interaction. In *Proceedings of the 8th International Semantic Web Conference*, pages 585–600.

[73] Shin, S., Jin, X., Jung, J., and Lee, K.-H. (2019). Predicate constraints based question answering over knowledge graph. *Information Processing & Management*, 56(3):445 – 462.

[74] Singh, S., Hillard, D., and Leggetter, C. (2010). Minimally-supervised extraction of entities from text advertisements. In *Proceedings of the 2010 Con-*

ference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 73–81.

[75] Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

[76] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

[77] Wang, Z., Qu, Y., Chen, L., Shen, J., Zhang, W., Zhang, S., Gao, Y., Gu, G., Chen, K., and Yu, Y. (2018b). Label-aware double transfer learning for cross-specialty medical named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–15.

[78] Weston, J., Chopra, S., and Adams, K. (2014). #TagSpace: Semantic embeddings from hashtags. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1822–1827.

[79] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[80] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49.

[81] Zhang, H., Boons, F., and Batista-Navarro, R. (2019). Whose story is it anyway? Automatic extraction of accounts from news articles. *Information Processing & Management*, 56(5):1837 – 1848.

[82] Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., and Lv, X. (2014). Ontology matching with word embeddings. In *Proceedings of the 13th Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 34–45. Springer.

[83] Zhou, G. and Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480.

[84] Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, pages 19–27. IEEE Computer Society.