Towards Encoding Time in Text-Based Entity Embeddings

Federico Bianchi, Matteo Palmonari and Debora Nozza

University of Milano - Bicocca, Viale Sarca 336, Milan, Italy
federico.bianchi,palmonari,debora.nozza@disco.unimib.it

Abstract. Knowledge Graphs (KG) are widely used abstractions to represent entity-centric knowledge. Approaches to embed entities, entity types and relations represented in the graph into vector spaces - often referred to as KG embeddings - have become increasingly popular for their ability to capture the similarity between entities and support other reasoning tasks. However, representation of time has received little attention in these approaches. In this work, we make a first step to encode time into vector-based entity representations using a text-based KG embedding model named Typed Entity Embeddings (TEEs). In TEEs, each entity is represented by a vector that represents the entity and its type, which is learned from entity mentions found in a text corpus. Inspired by evidence from cognitive sciences and application-oriented concerns, we propose an approach to encode representations of years into TEEs by aggregating the representations of the entities that occur in event-based descriptions of the years. These representations are used to define two time-aware similarity measures to control the implicit effect of time on entity similarity. Experimental results show that the linear order of years obtained using our model is highly correlated with natural time flow and the effectiveness of the time-aware similarity measure proposed to flatten the time effect on entity similarity.

1 Introduction

Knowledge Graphs (KGs) provide useful abstractions for representing knowledge, with nodes describing real-world entities and entity types, and labeled edges representing relations between entities, between types, and between entities and types. Traditional approaches to represent KGs use graph databases and semantic web technologies based on the RDF model[1]. More recently, complementary models to represent KGs have been proposed, which embed KG elements such as entities, types and relations into vector spaces of fixed dimensionality and learn such representations from large amounts of data [6, 14, 32, 1, 21, 30, 19]. We refer to these models as *KG embeddings*. In KG embeddings, entities are represented by vectors, and efficient geometric operations can support a variety of tasks such as the evaluation of similarity between arbitrary entity pairs.

[1] https://www.w3.org/RDF/

Some approaches generate KG embeddings using structured data as a source, e.g., relations occurring in the KG, and are mainly targeted at predictive reasoning tasks such as link prediction [6, 14, 32]. Other approaches generate the KG embeddings from text corpora using methods similar to the ones used to generate word embeddings [1], under the distributional hypothesis [11]. These models referred to as *text-based KG embeddings* in the following, are mainly targeted at similarity evaluation tasks.

In a previous work, we have presented Typed Entity Embeddings (TEEs) as one of the latter models [2, 4]. In TEEs, embeddings of entities and types are generated under the following entity-centric reinterpretation of the distributional hypothesis: *entities and types that appear in similar contexts are similar*. An entity linking algorithm [22] is used to find entity mentions in the corpus, while the KG is used to extract the most specific types of the mentioned entities. Then, based on the co-occurrence of entities and types in the text corpora two vector spaces are generated, one for entities and one for types. The direct sum of the two vector spaces leads to a *typed entities space*. In this space, each entity is represented by the concatenation of its vector in the entity space and the vector of its type. For example, the typed vector of the DBpedia entity `dbr:Rome` is the concatenation of the vectors generated for `dbr:Rome` and `dbo:City`.

Time is an important aspect in knowledge representation and has been extensively studied in the field of qualitative temporal representation and reasoning [31, 25, 23, 15]. In addition, time is essential to human cognition, as people "place events in time, deciding when they occurred, in which order and on what scale, whether that of a lifetime or of a few seconds" [8]. Finally, recent work has investigated temporal word embeddings to study language evolution along time using diachronic corpora [27]. Thus, we believe that encoding time into KG embeddings models is an important research objective.

Our work is inspired by evidence found in cognitive science studies as well as by application-oriented concerns. Time and time perception have been deeply investigated in the cognitive psychology literature. Since it has been observed that "the succession of events is an inherent property of our time perception. Memory is necessary, and the order of these events is fundamental" [26], we may consider textual descriptions of events found in text corpora as a sort of memory, and as a source for learning representations of time. Encoding time into KG embeddings has also several practical applications, in particular when evaluating entity similarity with text-based KG embeddings, where similarity depends on entity co-occurrence in similar contexts. Time can sneak into entity similarity in a way that cannot be controlled, because entities that share a temporal context are more likely to co-occur in the text (we refer to this implicit effect of time on entity similarity as to the *time effect hypothesis*). As a consequence, we may find that the most similar entity to `dbr:Winston Churcill` is the little-known `dbr:Harold McMillian`. This makes perfect sense, *if time is considered when evaluating the similarity*, but if we want to compare UK politician by international relevance and fame, rather then by their chronological order, we may prefer to find also more famous prime ministers like `dbr:Margaret Thatcher`

among the most similar entities to `dbr:Winston Churcill`. If we are able to explicitly incorporate time into KG embeddings, then we can control its effect when evaluating the similarity between entities, boosting or flattening time effect in entity similarity. Such control over similarity is helpful for example in knowledge exploration applications, which we investigated in previous work [3]. Other potential applications can be found in time-aware entity recommendations [18, 28] (e.g., find "related contemporary entities" vs. "related entities in the past" vs. "time-independent related entities") and in temporal information retrieval, where it is important to keep track of the time factor.

To the best of our knowledge, in this paper we propose a first approach to make time a first-class citizen in KG embeddings. We use the text-based TEEs model as background and learn explicit representations of temporal entities as part of this model. In particular, we encode representations of years, i.e., we embed regular time periods with a yearly granularity. We build year representations from the textual description of events occurring during each year, which are available in different web sources[2]. We generate year representations by aggregating the representations of the entities that that take part in events occurring in those particular years. These representations are then used to define two parametric *time-aware similarity functions*: time-flattening and time-boosting similarities.

In other words, in this paper we tackle the following research challenges: 1) to generate representations of time periods that are inspired by evidence found in cognitive psychology for memory being a fundamental aspect in time representation [26]; 2) to use these representations to control the effect of time over entity similarity for practical applications. The contributions with respect to these objectives can be summarized as follows:

- We learn representations of time periods at a yearly granularity starting from natural language descriptions of events occurring in these periods, showing that, even if the natural time flow is not explicitly encoded into the model, the generated year representations are highly correlated with the natural sequence of years.
- We provide evidence for the time-effect on entity similarity in text-based KG embeddings.
- We propose two parametric time-aware similarity measures to control the time effect in entity similarity.

Our approach to encoding time into text-based KG embeddings is explained in Section 2. Experiments to evaluate the time effect, properties of the year representations and the time-aware similarities are discussed in Section 3. Related work is discussed in Section 4. Conclusions and future work end the paper.

2 Typed Entity Embeddings with Time Periods

We use a minimal definition of Knowledge Graph (KG) as a directed labeled graph, where nodes are *entities* or *types*, and labeled edges represent *relations*

[2] Examples are Wikipedia pages for years, https://www.onthisday.com/events-by-year.php and https://www.history.com/this-day-in-history

between entities or between types or between entities and types. Relations between types define a sub-type graph (often a hierarchy). However, the relations between the entities are not considered for generating the embeddings in our work. An example of KG, which will be used in the rest of the paper, is DBpedia, where types are classes in the DBpedia Ontology. For simplicity, we assume that given an entity we can determine its minimum type (i.e., its most specific type) using the typing assertions and the sub-type graph. In case an entity has more minimal types none of which is the minimum, different strategies can be considered to select one of them as representative type. In DBpedia the most specific type of entities can be determined by using dedicated resources[3].

A Typed Entity Embedding (TEE) with Time Periods consists in:

- A set of typed entities E, which includes a subset E^τ of temporal periods.
- Two embedding functions; $\phi : E \setminus E^\tau \to \mathbb{R}^k$ and one $\omega : E^\tau \to \mathbb{R}^k$.
- A similarity function $\eta : E \times E \to [0, 1]$ constructed as operation over the typed entity vectors.
- A proximity function $\rho : E \to E^\tau$ that allows us to find the most representing time period for a given typed entity.
- A time-aware similarity function $\psi : E \times E \to [0, 1]$ that computes the similarity between two typed entities by considering their time distance as a factor.

In the following, we explain each component of the model more in detail.

2.1 TEEs and their Generation

The ϕ embedding function has been intuitively explained in Section 1. For more details we refer to previous work [2, 4]. Here we provide some more insights about the generation process using Figure 1. As a corpus we use a set of documents, each one describing an entity in natural language. As shown in the figure, after the entity mentions are found in a document by an entity linking algorithm, we generate a second document that consists in the sequence of the entities found in the corpus text. This document is transformed into a third document, where entities are replaced by their most specific type, obtaining a sequence of types. As a result, we have two corpora, one for entities and one for types. At this point we run word2vec [16] on each corpus to generate Entity Embeddings **(EE)** and Type Embeddings [4] (**TE**) separately. These two embeddings can have different dimensionality. Finally, for each entity, we concatenate its entity vector with the vector of its most specific type, thus obtaining a typed entity vector, i.e., a Typed Entity Embedding **(TEE)**. Entities of the same type (or of similar types) are more likely to be closer to each other in the TEE space built upon this concatenation than in the EE space that consists of entity-only vectors [2]. Given an entity e, we use the bold notation $\mathbf{e}$ to refer to its typed entity vector.

Observe that since we have generated the typed entity vector space as the direct sum of the entity and the type vector spaces, we can easily drop the type-component in typed entity vectors and use entity vectors without representing

[3] http://wiki.dbpedia.org/services-resources/documentation/datasets#InstanceTypes

the entity types. We will use these simplified entity vectors in the experiments discussed in Section 3.1.

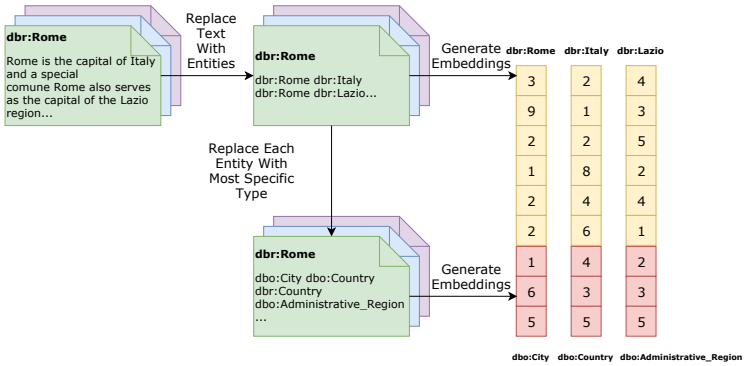

Fig. 1. Entity embedding process: textual content is replaced by entities. Each entity is replaced with its own type. Embeddings can be then generated using word2vec. Finally each entity is concatenated to its own type.

2.2 Encoding Temporal Periods into TEEs

In this work, we consider time as a set of connected time periods E^τ, i.e., a sequence of time periods totally ordered by a relation $<^\tau$. We represent time periods at the year granularity, meaning that each year represents the time period that spans over the year duration. In the following, we describe the function ω, which embeds time periods into $\mathbb{R}^k$.

Our main hypothesis is that discrete periods of times can be embedded in a vector space, where each period is represented by a vector, in such a way that years that are near in time have similar vectors. A second hypothesis that drives our approach is that a period of time, e.g., a year, can be described by the entities that take part in the events that occur during the time period. For example, years in the first half of the 40s are characterized by World War II events and by the entities that had a relevant role in these events. For the experiments conducted in this paper, we consider textual descriptions of events that appear in the Wikipedia pages that describe years [4].

To generate the representation of a year, we extract entities from the corresponding Wikipedia page and compute the average vector of the entity vectors defined in the EE space [5]. In other words, we drop the type component from the typed entity vectors, to use a more entity-centric representation (types occur

[4] E.g., https://en.wikipedia.org/wiki/1943

[5] Our representation of years is independent from the ϕ embedding function: other embedding algorithms could be used to compute the entities representation.

more regularly across years). To generate a TEE-compatible representation of the time periods in the vector space, we concatenate each embedding generated in the EE space with a vector consisting of 0s of the same dimension of type vector in TEE. This process is briefly summarized in Figure 2. Slight variations of this process are also possible, e.g., different vector aggregation methods, as discussed in Section 2.3. Finally, we empirically found that it is better to consider only the "Events" section in year Wikipedia pages for entity extraction because the sections "Births" and "Deaths" produce noisier representations.

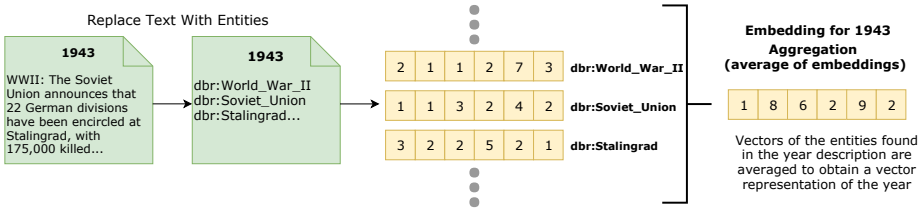

Fig. 2. Year embedding process.

Since non-temporal and temporal entities (i.e., time periods) are embedded in the same space, a comparison between entities of these two kinds is possible. In addition, these representations of time periods are generated using collective knowledge of what has happened during their time. However, since no explicit constraints over time ordering are used in the generation process, a natural question is which relation can be found between the vector-based representations of time periods and the natural time order. In Figure 3 we show an example of the 2D representation of the years from 1900 to 2015 using PCA. Interestingly, the years seem to follow a natural time order from left to right. A statistical correlation analysis between the one-dimensional projection of years using PCA and their natural order confirms this intuition (Kendall τ = **0.80**, Spearman Rank correlation coefficient = **0.94**).

2.3 Temporal Embeddings Alternative Configurations

There are different ways to use the entities that are found in the year description: 1) considering the entities only one time (i.e., as a set of entities); 2) considering the entities multiple times if they appear more than once (i.e., United States might appear more than one time in the text); 3) Using TF-IDF on the whole year corpus to weight each term by its own TF-IDF score and apply this with 1) and 2). We generated models using all these alternative configurations and we projected the embeddings into 1D using PCA and compared this ordering in 1D with the natural flow time order (i.e., the natural sequence from 1900 to 2015) and we obtained that the model that considers each entity only once (1) is the one that is most correlated to the natural time order. We thus decided to use this configuration to generate temporal embeddings.

Fig. 3. Average vectors represented in two dimensions using PCA.

2.4 Time-Aware Similarity Measures

We propose a new way of computing similarity that also considers the temporal factor in the embeddings. Given an entity $e \in E$ is it possible to find its most representing year in E^τ by considering the functions defined in the model. To get the most representative year for a given entity we select the most similar year in E^τ to a given entity.

$$\rho(e) = \underset{e^\tau \in E^\tau}{\operatorname{argmax}}\, cos(\phi(e), \omega(e^\tau))$$

We use $\mathbf{e}^\tau$ to denote the vector of $\rho(e)$, i.e., the closer year to a given entity in the vector space.

We can now define two time-aware similarity functions: a **time-flattened similarity** and **a time-boosted similarity**. The time-flattened similarity can be computed using the following formula.

$$\psi(e_1, e_2) = \alpha\eta(\mathbf{e_1}, \mathbf{e_2}) - (1-\alpha)\eta_n(\mathbf{e_1^\tau}, \mathbf{e_2^\tau})$$

Where $\mathbf{e_1}$, $\mathbf{e_2}$ are the embeddings of the entities e_1, e_2, η is the cosine similarity in the typed entity space, and α is a parameter that can be used to regulate the weight of the time flattening factor. Time flattening is obtained by subtracting the temporal similarity η_n of the most representative temporal periods (i.e., years) of both entities $\mathbf{e_1^\tau}$ and $\mathbf{e_2^\tau}$. The temporal similarity is defined as the cosine similarity between two years in the typed entity space normalized in the interval [0, 1] with a max-min approach, by considering the maximum similarity between two years in the representation and the minimal similarity between two years in the representation. We adopt this normalization to make the year factor of the similarity work as a weight factor.

A time-boosted similarity function can be defined analogously by adapting the formula in such a way that a time-boosting factor is summed to the similarity between the typed entities:

$$\psi(e_1, e_2) = \alpha\eta(\mathbf{e_1}, \mathbf{e_2}) + (1 - \alpha)\eta_n(\mathbf{e_1^\tau}, \mathbf{e_2^\tau})$$

3 Experimental Evaluation

The experiments that we discuss in this section have the following goals: 1) validate the time-effect hypothesis introduced in Section 1, that is, that time influences the distribution of entities in vector spaces, and the rationale behind the generation of time period representations from those of the entities that take part in the events that occur during these time periods; 2) evaluate the effectiveness of the time-aware similarity function.

Experimental Settings (for all the Experiments). Our entity embeddings were generated using the DBpedia's abstracts (4M of textual documents) from the 2016 dump[6]. We used the skip-gram algorithm for obtaining the entities embeddings [16]. We used a window of 5 in the algorithm and types and entities are embedded into 100 dimensional vector spaces. We annotated text using DBpedia Spotlight [7]. For the year embeddings we decide to concentrate our experiments on the years from 1900 to 2015. Code and dataset are freely available online so that experiments can be replicated [7].

3.1 Time Effect and Temporal Representations in Text-based Entity Embeddings

In a first experiment, we evaluate if the time effect can be noticed in text-based entity embeddings, i.e., temporal contexts shared by entities have an effect on their similarity. Then, with a second experiment, we provide evidence that years that are close in time are more likely to have descriptions that share a larger number of entities, thus supporting the rationale behind using entity representations to generate temporal representations. The time effect validated with the first experiment adds even more substance to this idea, since the time effect suggests that time is implicitly encoded in entity representations. As a consequence, temporal representations are generated using entity representations that implicitly encode some temporal characterization. Finally, in a third experiment, we investigate if the space that jointly represents years (temporal entities) and other entities can support entity ordering over time, to further evaluate the quality of the temporal representations and their relation to the representations of the other entities. We remind that some properties of our model, e.g., the correlation between projection on one dimension of temporal entities and natural flow of time, have been discussed in Section 2.

[6] http://wiki.dbpedia.org/dbpedia-version-2016-04
[7] https://github.com/vinid/time-aware

Classifying World War I and World War II Battles

Our assumption is that entity embeddings share time-based context, and thus entities that live nearby times are closer to each other in the vector space. We collect battles from World War I and World War II using the list provided by Wikipedia military engagements pages[8][9]. We used a clustering algorithm to understand if there is an underlying pattern that puts entities in two different groups. We use K-means (number of clusters equal to 2 to cluster the vector representations of the battles in two different groups represented by the two wars). We will evaluate the performance of the clustering algorithm by considering how many years were clustered in the correct group.

Dataset. Our dataset contains 152 battles linked to Wikipedia (and thus DBpedia) from the two different periods 1914-1918 and 1939-1945. 63 battles are from World War I while 89 are from World War II.

Results. In Table 1 we show the confusion matrix that we obtained after clustering the embedded vectors of the battles with K-Means. Out of 152 samples, 146 were correctly associated to the same cluster, while 8 were classified in the wrong one. Accuracy is around 95%. Another interesting result is that the two cluster centroids are closer to the respective war years: the first centroid is near the years of World War I, while the second centroid is close to the years of World War II.

Table 1. Confusion Matrix for World War I and II clustering.

Actual Values (n=152)		**Predicted**	
		World War I	**World War II**
	World War I	57	6
	World War II	2	87

Adherence to Natural Time Order

Our intuition suggests that the descriptions of contiguous years (e.g. 1943 and 1944) share more entities than the descriptions of years that are not contiguous (e.g. 1901 and 1992).

Methodology and Dataset. We collect every possible combination of two and three contiguous years (e.g. 1900-1901,1901-1902; 1900-1901-1902, 1901-1902-1903) and we compute the average number of shared entities. We compare the values of these two samples with the average values of shared entities of *every* possible combination of two and three years (e.g. 1902-1992, thus considering also noncontiguous years).

Evaluation. Results on the average number of shared entities are reported in Table 2. Pairs and triples of contiguous years have a higher amount of shared entities with respect to noncontinuous years.

[8] https://en.wikipedia.org/wiki/List_of_World_War_I_battles
[9] https://en.wikipedia.org/wiki/List_of_military_engagements_of_World_War_II

Table 2. Average number of shared entities between continuous and non contiguous years.

	Contiguous-2	**All-2**	**Contiguous-3**	**All-3**
Average	55.6	33.5	27.7	12.6
Std	30.3	19.2	13.3	8.22

We use the Kolmogorov-Smirnov test to detect if the average number of shared entities of contiguous years is statistically different from the respective value for of all the combinations of years. A p-value lower than 0.05 confirms our hypothesis.

Relative Ordering of Entities by Time

In this experiment, we show that time actually affects the position of entities in the space.

Dataset. We pick 101 entities from different groups of people and events (United States Presidents, British Prime Ministers, French Presidents, Fifa World-Cup Years, Wars over in 1900, Olympics Events). Entities and groups have been chosen so as to select pairs of entities for which a chronological order can be established upon a reasonably objective criterion (e.g., dbr:Barack Obama is a president elected after dbr:Woodrow Wilson). We acknowledge that ordering people by considering one single feature is a strong assumption. However, being prime minister or president is a very discriminant feature for people.

Methodology. For each entity pair, we compare the manually determined relative order with the order of their most representative years according to our model. The most representative year of an entity is the the closest year to the entity in the vector space. We want to show that given two entities $e_1, e_2 \in E$ such that e_1 is known to chronologically come before e_2, it is likely that $\rho(e_1) >^{\tau} \rho(e^2)$. For each pair, we also computed the number of *time steps* separating the two entities. This is a measures that indicates a relative distance between two entities: the number of time steps between Barack Obama and George W. Bush is 1, because Bush was the US president before Obama, while between Barack Obama and Bill Clinton it is 2, because Bush was between Obama and Clinton. The same is applied to events like the Fifa World Cup (e.g., the time step between the 2006 world cup and the 2002 world cup is 1).

Results. The accuracy of the relative orders was 70%. For 217 pairs the model was not able to decide a relative ordering since $\rho(e_1) = \rho(e_2)$[10]. In Figure 4, we show the distribution of correctly relative ordered pairs, incorrect relative ordered pairs and pairs that the model could not order by time steps involved in the relative order. The radius of the points is used to indicate the number of time pairs with a certain time step in each category. It is clear that the farther in time two entities are, the easier it is to determine a correct relative order. Thus,

[10] If for the generation of e^{τ} we consider the average of the nearest 10 years to an entity all the 902 pairs can be compared and the accuracy reaches 92%.

time influences the position of entities in the vector space and the estimation of a relative time order between entities using their representative year is quite accurate.

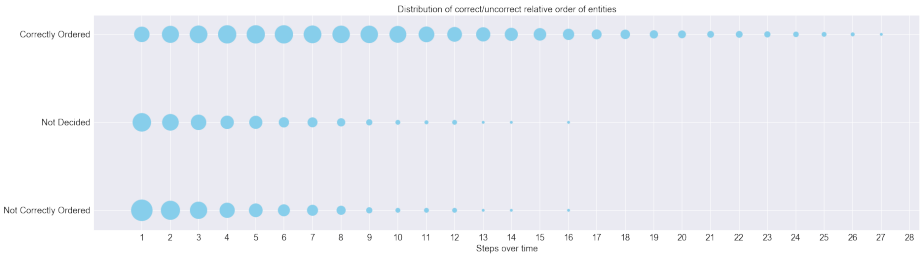

Fig. 4. Distribution of correctly and not correctly ordered pairs with the use of EE.

3.2 Time-Aware Similarity

To test time-aware similarity measures we concentrate on time-flattened similarity for two reasons: it is useful to mitigate the time-effect that we have discussed in the previous experiment and it can be evaluated more objectively using data available in KGs. Defining what is similar when considering the time variable is a challenging task. In this experiment we decided to provide a small-scale experiment on the possible use of the time-flattened similarity by considering reasonably objective orders is time. A time-flattening similarity should reduce the effect that a shared temporal context has on finding similar entities. A time-flattened similarity can be used to find entities that are similar independently from the temporal context they share. For example, given a prime minister, we would expect to find many other prime ministers among its most similar entities if we neglect time, but the time-effect moves many prime ministers down in the ranking with cosine similarity.

Dataset. In line with other experiments done on similarity and relatedness [12] we create a dataset containing entities that are related but distant in time. Given an input entity the task consists in finding similar entities that are distant in time (e.g., given dbr:Barack Obama, a time-flattened similarity should rank in higher position the entities dbr:Theodore Roosevelt and dbr:William Howard Taft). Given a set of 12 prime ministers, the task is therefore to compare the number of prime ministers found in the 5/10 most similar entities retrieved using non-time flattened similarity measures and time-flattening similarity measures. We used prime ministers because this is a salient feature in making entities "ontologically" similar (e.g., very few entities share this feature). We selected 6 entities representing the most recent US presidents (from a list of the most recent 19 presidents) and 6 entities representing the most recent British Prime Ministers (from a list of the most recent 19 prime minister).

Algorithms and Methods. We consider different algorithms to test the time-flattened similarity: we want to test if our model is actually able to retrieve entities that are far in time starting from input entities. We select the 100 nearest entities given an input entity using cosine and order them using time-flattened similarity. We then compute Precision@K and Recall@K. This task is tested on both TEE and the EE model: we will compare standard neighborhood of the input entity (based on cosine similarity) and the time-flattened one (considering the time-flattened similarity).

Baseline. As a Baseline we consider a similarity measure that uses a skip-gram model trained on a corpus that also contains mentions to entity years[11]. Time-flattening in the baseline is computed considering the closest *entity* year to a given entity as the most representative year, similarly to what we do in our similarity. The difference with our representation is that, in this Baseline, the entity year representations are learned by considering the co-occurrence in text as in standard embedding models and do not have an explicit representation generated by a dedicated embedding function like in our model. The tested models are:

- Time-aware Similarity TEE (TATEE), with time-flattened similarity;
- Similarity TEE (STEE) (standard neighborhood with cosine);
- Time-Aware Similarity EE (TAEE), with time-flattened similarity;
- Similarity EE (SEE) (standard neighborhood with cosine);
- Time-flattened similarity Baseline (Baseline).

Experiments on the time-flattened similarity were run with $\alpha = 0.7$.

Results. Table 3 shows the results. The use of a time-flattening factor can improve the retrieval of entities that are distant in time. Models that use types have an advantaged: the tasks, in fact, consists in finding entities that share more or less of the same type. However, the performance of the model that does not use types, but time-flattened (TAEE) is better than the baseline. We can conclude that the use of both the TEE model and time-awareness (time-flattening in this case) allows achieving better performance on this task.

Table 3. Results for time-flattened similarity (0* means small values)

	Precison@5	**Precison@10**	**Recall@5**	**Recall@10**
TATEE	**0.40**	**0.40**	**0.20**	**0.21**
STEE	0.14	0.21	0.07	0.10
TAEE	0.05	0.04	0*	0.01
SEE	0.02	0.02	0*	0.01
Baseline	0.01	0.01	0*	0.01

[11] https://github.com/idio/wiki2vec

Qualitative Evaluation. To provide further insights on the behavior of our time-flattening similarity, we discuss an example in details. If we consider the entity *dbr:Winston Churchill* and the top-100 entities more similar to it, recent but popular British prime ministers are found distant in the TEE model when retrieved using plain cosine similarity: *dbr:Tony Blair* and *dbr:Gordon Brown* are respectively in the 49th and the 41th position. If we use our time-flattening similarity, the two entities are found respectively at the 16th and 14th position. The nearest entity to *dbr:Winston Churchill* is *dbr:Harold Macmillan* (Member of Churchill government and British prime minister two years after Winston Churchill) if we use plain cosine similarity, and *dbr:Margaret Thatcher* if we use our time-flattened similarity.

Time-flattening/time-boosting. In Table 4, we list the top-10 most similar entities to Barack Obama, when retrieved with time-flattened, plain cosine, and time-boosted similarity. For time-aware similarities we also show differences for different values of α, to show the effect of this parameter (remember that 0.7 was used as value in previous experiments). We believe that this example shows an interesting behavior: removing the time effect with time-flattened similarity pushes old presidents of the United States in higher positions. Otherwise, if we use the time-boosted similarity, members of the Obama government and his rivals during the elections (i.e., John McCain and Mitt Romney) are the ones pushed in higher positions.

Table 4. Time-flattened and time-boosted similarity on the entity Barack Obama.

Time Flattened Similarity - Time ←		**Cosine Similarity**	**Time Boosted Similarity → + Time**	
$\alpha = 0.1$	$\alpha = 0.7$		$\alpha = 0.7$	$\alpha = 0.1$
G. Ford	B. Clinton	B. Clinton	B. Clinton	G. Bush
C. Coolidge	Reagan	Reagan	G. Bush	**J. Kerry**
H. Hoover	Carter	G. Bush	Reagan	D. Cheney
Truman	Al Gore	Carter	Kerry	**McCain**
F. Roosevelt	Nixon	Al Gore	D. Cheney	**Biden**
W. Wilson	**G. Ford**	Nixon	McCain	Ron Paul
Eleanor Roosevelt	G. Bush	J. Kerry	Biden	H. Humphrey
D. Eisenhower	**C. Coolidge**	D. Cheney	Carter	**Romney**
W. Harding	T. Kennedy	McCain	Al Gore	C. Powell
G. Cleveland	**H. Hoover**	Biden	Ron Paul	W. Mondale

4 Related Work

Qualitative temporal representation and reasoning is a topic covered by a vast literature in Artificial Intelligence and related fields, for which we refer to several surveys [31, 25, 23]. Different mathematical models of time such as point-based

vs. interval-based, linear time vs. branching time, have been proposed [31]. Models to support reasoning with approximate time intervals have also been proposed [5]. Previous work has surveyed models to represent temporal information in RDF [23] and reason about time in natural language processing [25]. In our work, we use a simple model of time as a sequence of regular time periods and do not tackle logic-based temporal reasoning. Otherwise, none of the previous approaches has addressed the problem of generating temporal representations from texts. Extraction of temporal information from text (see, e.g., [15]) and imputation of temporal validity intervals for RDF triples (see, e.g., [24]), are tasks also very different from the one addressed in this paper.

Many approaches for KG embeddings that consider knowledge graph structure have been introduced in literature [20, 14, 32, 6, 21, 30, 19]. For example, TransE [6] embeds entities in a space in which for each triple (s, p, o), $\mathbf{s} + \mathbf{p} \approx \mathbf{o}$ holds. All these methods are able to efficiently represent entities and relations of a KG into a vector space, but none of them take explicit steps towards the representation of temporal entities. Other methods have been introduced to represent temporal information in KGs [29, 13, 9] and have obtained good results in tasks like time-aware link prediction. The main difference with our approach is that we explicitly embed temporal entities inside a vector space.

We used a semantic annotator to extract entities from text. If we consider Wikipedia, text can be replaced with the use of links as done in other models [1]. Our approach can be generalized to any kind of text, even those that do not contain links, such as books or newspapers.

Worth mentioning in this context are the works on temporal word embeddings [27]. These representations are often called diachronic embeddings [10], since they start from collections of documents coming from different periods in times and build and embedding for each of the periods. The study of embeddings at different points in time has shown that words are subject to a shift in meaning that can be quantified using distance measures between different embeddings across the vector space. The main difference between our work and theirs is that we are embedding temporal entities inside a KG, while often the task proposed in other approaches is to study the changes of meaning in words over during time [10]. Following this methodology, a recent work on time-aware entity relatedness that uses word embeddings learned from a collection of documents that spans different time periods has been proposed [17].

5 Conclusions and Future Work

In this paper we have presented an approach to encoding temporal periods into text-based entity embedding models. In particular, we used our previous work to generate Typed Entity Embeddings (TEEs) from textual descriptions of entities and encoded into this model the representations of years. These representations are generated using natural language descriptions of events occurring during the years. To the best of our knowledge, this is the first attempt to explicitly encode time into KG embedding models. In addition, we have defined a parametric

time-aware similarity function that can be tuned to boost or flatten the effect of time when computing the entity similarity.

In our experiments we have shown that time has an effect on entity embeddings built from text, thus validating the main hypothesis behind this work. Then we have tested our time-aware similarity function to show that it can capture aspects of similarity that other time-agnostic similarity measures cannot capture. Such a similarity measure can provide novel knowledge exploration methods where time can be factored when finding entities similar to each other.

Our results provide a first contribution to the problem of encoding time into KG embeddings built from text, which poses several challenges that we want to address in future work. So far, we have considered sequences of regular time periods at a yearly granularity. An important challenge would be to consider different granularity levels and, even more important, to study the compositional nature of temporal representations extracted from text. For example, we would like to generate a vector for the 70s by composing vectors of years 197X. More in general, we would like to investigate how vector-based representations of time periods can be composed so as to provide a soft account of relations between time intervals that are considered in qualitative models of temporal reasoning like Allen algebra.

References

1. Basile, P., Caputo, A., Rossiello, G., Semeraro, G.: Learning to rank entity relatedness through embedding-based features. In: International Conference on Applications of Natural Language to Information Systems. pp. 471–477. Springer (2016)
2. Bianchi, F., Palmonari, M.: Joint learning of entity and type embeddings for analogical reasoning with entities. In: NL4AI Workshop, co-located with the International Conference of the Italian Association for Artificial Intelligence (AI* IA) (2017)
3. Bianchi, F., Palmonari, M., Cremaschi, M., Fersini, E.: Actively Learning to Rank Semantic Associations for Personalized Contextual Exploration of Knowledge Graphs. In: ESWC. pp. 120–135. Springer International Publishing (2017)
4. Bianchi, F., Soto, M., Palmonari, M., Cutrona, V.: Type Vector Representations from Text: An Empirical Analysis. In: DL4KGS workshop, co-located with the ESWC (2018)
5. Bittner, T.: Approximate qualitative temporal reasoning. Annals of Mathematics and Artificial Intelligence **36**(1-2), 39–80 (2002)
6. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS. pp. 2787–2795 (2013)
7. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: I-Semantics (2013)
8. Damasio, A.R.: Remembering when. Scientific American **287**(3), 66–73 (2002)
9. Esteban, C., Tresp, V., Yang, Y., Baier, S., Krompaß, D.: Predicting the co-evolution of event and knowledge graphs. In: 2016 19th International Conference on Information Fusion (FUSION). pp. 98–105 (July 2016)
10. Hamilton, W.L., Leskovec, J., Jurafsky, D.: Diachronic word embeddings reveal statistical laws of semantic change. arXiv preprint arXiv:1605.09096 (2016)

11. Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954)
12. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: Kore: keyphrase overlap relatedness for entity disambiguation. In: CIKM. pp. 545–554. ACM (2012)
13. Jiang, T., Liu, T., Ge, T., Sha, L., Li, S., Chang, B., Sui, Z.: Encoding temporal information for time-aware link prediction. In: EMNLP. pp. 2350–2354 (2016)
14. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. pp. 2181–2187 (2015)
15. Ling, X., Weld, D.S.: Temporal information extraction. In: AAAI. vol. 10, pp. 1385–1390 (2010)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
17. Mohapatra, N., Iosifidis, V., Ekbal, A., Dietze, S., Fafalios, P.: Time-Aware and Corpus-Specific Entity Relatedness. In: DL4KGS workshop, co-located with the ESWC (2018)
18. Nguyen, T.N., Kanhabua, N., Nejdl, W.: Multiple models for recommending temporal aspects of entities. In: ESWC. pp. 462–480. Springer (2018)
19. Nickel, M., Rosasco, L., Poggio, T.A., et al.: Holographic embeddings of knowledge graphs. In: AAAI. pp. 1955–1961 (2016)
20. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of ICML-11. pp. 809–816 (2011)
21. Ristoski, P., Paulheim, H.: Rdf2vec: Rdf graph embeddings for data mining. In: ISWC. pp. 498–514. Springer (2016)
22. Rizzo, G., Troncy, R.: Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In: EACL. pp. 73–76. ACL (2012)
23. Rula, A., Palmonari, M., Harth, A., Stadtmüller, S., Maurino, A.: On the diversity and availability of temporal information in linked open data. In: ISWC. pp. 492–507. Springer (2012)
24. Rula, A., Palmonari, M., Ngomo, A.N., Gerber, D., Lehmann, J., Bühmann, L.: Hybrid acquisition of temporal scopes for RDF data. In: ESWC. LNCS, vol. 8465, pp. 488–503. Springer (2014)
25. Sanampudi, S.K., Kumari, G.V.: Temporal reasoning in natural language processing: A survey. International Journal of Computer Applications **1**(4), 68–72 (2010)
26. Snaider, J., McCall, R., Franklin, S.: Time production and representation in a conceptual and computational cognitive model. Cognitive Systems Research **13**(1), 59–71 (2012)
27. Szymanski, T.: Temporal Word Analogies: Identifying Lexical Replacement with Diachronic Word Embeddings. In: ACL. Association for Computational Linguistics, Vancouver, Canada (August 2017)
28. Tran, N.K., Tran, T., Niederée, C.: Beyond time: Dynamic context-aware entity recommendation. In: ESWC. pp. 353–368. Springer (2017)
29. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: ICML. pp. 3462–3471 (2017)
30. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML. pp. 2071–2080 (2016)
31. Van Beek, P.: Reasoning about qualitative temporal information. Artificial intelligence **58**(1-3), 297–326 (1992)
32. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. pp. 1112–1119 (2014)