

# Time-Aware and Corpus-Specific Entity Relatedness

Nilamadhava Mohapatra<sup>1,2</sup>, Vasileios Iosifidis<sup>1</sup>, Asif Ekbal<sup>2</sup>,  
Stefan Dietze<sup>1</sup>, and Pavlos Fafalios<sup>1</sup>

<sup>1</sup> L3S Research Center, University of Hannover, Germany  
{iosifidis, dietze, fafalios}@L3S.de

<sup>2</sup> Indian Institute of Technology, Patna, India  
{nilamadhava.mtmc15, asif}@iiitp.ac.in

**Abstract.** Entity relatedness has emerged as an important feature in a plethora of applications such as information retrieval, entity recommendation and entity linking. Given an entity, for instance a person or an organization, entity relatedness measures can be exploited for generating a list of highly-related entities. However, the relation of an entity to some other entity depends on several factors, with *time* and *context* being two of the most important ones (where, in our case, *context* is determined by a particular corpus). For example, the entities related to the *International Monetary Fund* are different now compared to some years ago, while these entities also may highly differ in the context of a USA news portal compared to a Greek news portal. In this paper, we propose a simple but flexible model for entity relatedness which considers time and entity aware word embeddings by exploiting the underlying corpus. The proposed model does not require external knowledge and is language independent, which makes it widely useful in a variety of applications.

**Keywords:** Entity Relatedness · Word2Vec · Entity Embeddings

## 1 Introduction

Entity relatedness is the task of determining the degree of relatedness between two entities. Measures for entity relatedness facilitate a wide variety of applications such as information retrieval, entity recommendation and entity linking. Traditional approaches consider the structural similarity in a given graph and lexical features [5, 6, 10–12, 15], or make use of Wikipedia-based entity distributions and embeddings [1, 2].

Recent works on related topics have supported the hypothesis that the context of entities is temporal in nature. Specifically, [4] has shown that prior probabilities often change across time, while [17] showed that the effectiveness of entity recommendations for keyword queries is affected by the time dimension. In addition, [16] introduced the notion of contextual entity relatedness and showed that entity relatedness is both time and context dependent. In that work, context refers to *topic (aspect)*, i.e., the goal is to find the most related entities given

an entity and an aspect (e.g., *relationship* or *career* are two different aspects for a person). Finally, [13] showed that exploiting different temporal versions of a knowledge base, each reflecting a different time period, can improve the accuracy of entity relatedness.

On the other hand, entity relatedness is also strongly dependent on the *corpus context* at hand. For instance, given a search application operating over a collection of German news articles from summer 2014 and the query entity *2014 FIFA World Cup* ([https://en.wikipedia.org/wiki/2014\\_FIFA\\_World\\_Cup](https://en.wikipedia.org/wiki/2014_FIFA_World_Cup)), a list of related entities should include *Germany national football team*, *Argentina national football team*, *Mario Götze*, and *Brazil*. However, considering a collection of Greek articles of the same time period, the top related entities might include *Greece national football team*, *Costa Rica national football team*, and *Sokratis Papastathopoulos* (entities that are not important or popular in German news).

To this end, our work considers entity relatedness as a measure which is strongly dependent on both the time aspect and the corpus of the underlying application. Contrary to existing approaches that train embeddings on general-purpose corpora, in particular Wikipedia [1, 2], we compute entity relatedness given a specific collection of documents that spans a particular time period. Our approach exploits entities extracted from the underlying corpus for building time and entity aware word embeddings. The evaluation results show that the proposed model outperforms similar time and entity agnostic models.

The remainder of this paper is organized as follows: Section 2 defines the problem of *time-aware and corpus specific entity relatedness*, Section 3 details the proposed method, Section 4 presents evaluation results, and finally Section 5 concludes the paper and discusses interesting directions for future research.

## 2 Problem Definition

Let  $D$  be a corpus of documents, e.g., a collection of news articles, covering the time period  $T_D = [\tau_s, \tau_e]$  (where  $\tau_s, \tau_e$  are two different time points with  $\tau_s < \tau_e$ ). For a document  $d \in D$ , let  $E_d$  denote all entities mentioned in  $d$  extracted using an entity linking method [14], where each entity is associated with a unique URI in a knowledge base like Wikipedia. This list of extracted entities may include persons, locations, etc., but also events (e.g., *US 2016 presidential election*) and more abstract concepts such as *democracy* or *abortion*. Finally, let  $E_D$  denote all entities mentioned in documents of  $D$ .

Consider now a collection of documents  $D$  covering a time period  $T_D$  and a set of entities  $E_D$  extracted from and prevalent in  $D$ . Given i) one or more *query entities*  $E_q$  and ii) a time period of interest  $T_q \subseteq T_D$ , the task “*time-aware and corpus-specific entity relatedness*” focuses on finding a top- $k$  list of entities  $E_k \subset E_D$  related to the query entities  $E_q$  in terms of both  $T_q$  and  $D$ . We model this task as a ranking problem, where we first generate a list of *candidate entities*  $E_c \subset E_D$  and then rank this list of entities based on their relevance to the query entities. For generating the list of candidate entities, one can follow an approach similar to [17] which exploits Wikipedia links, DBpedia and entity

co-occurrences in the annotated corpus, or consider the connectedness of the query entity with other entities in Social Media [3].

### 3 Approach

Word embeddings provide a distributed representation of words in the semantic space. We generate word vectors following the distributional models proposed in [8, 9] and the well-known Word2Vec tool.

#### 3.1 Time-Aware Word Vector Similarity

We group all documents in  $D$  into time-specific subsets  $C = (C_1, \dots, C_n)$  based on a fixed time granularity  $\Delta$  (e.g., week, month, or year). We then preprocess the documents of each subset  $C_i \in C$  and train a Word2Vec Continuous Bag of Words (CBOW) model. Each model generates a matrix  $keys \times dimension(d)$  where  $keys$  represents a word from the corpus  $C_i$  for which a  $d$ -dimensional vector exists in the trained model.

To find the relatedness score between a query entity  $e_q \in E_q$  and a candidate entity  $e_c \in E_c$ , we compute the cosine similarity of their word vectors in the corresponding trained model for the input time period  $T_q$ . In our experiments, for finding the words that represent a query or candidate entity, we use the last part of the entity’s Wikipedia URI by first replacing the underscore character with the space character and removing any text in parentheses. Based on the underlying knowledge base, one could use here other approaches, e.g., exploit the entity’s label in DBpedia. For multi-word entities, we compute the average vector of the constituent words, while for inputs consisting of more than one entity, we compute the average of all entity vectors.

#### 3.2 Considering the Entity Annotations

There are two limitations in dealing with the embedding vectors obtained in the previous model: (i) for multi-word entities the average of the word vectors of the constituent tokens is computed, i.e., n-grams are ignored (consider for example the entity *United Nations*), and (ii) the same entity mention (surface form) may refer to different entities (e.g., *Kobe* may refer to the basketball player *Kobe Bryant* or the *Japanese city*). To cope with these problems, we exploit the entity annotations  $E_D$ . Recall that each extracted entity is associated with a unique URI in a knowledge base (e.g., Wikipedia or DBpedia). As also suggested in [9] for the case of phrases, we preprocess the documents and replace the entity mentions with unique IDs. The modified text corpora are now used for training the Word2Vec model, where the word vector for an entity is now calculated using its unique ID.

Now, the similarity score between a query entity  $e_q \in E_q$  and a candidate entity  $e_c \in E_c$  is the cosine similarity of their word vectors using the corresponding modified (with entity IDs) Word2Vec model for the input time period  $T_q$ .

Formally:

$$Sim(e_q, e_c, T_q) = cosine(wordvec(id(e_q), T_q), wordvec(id(e_c), T_q)) \quad (1)$$

where  $id(e)$  is the ID of entity  $e$ , and  $wordvec(w, T_q)$  returns the word vector of token  $w$  using the corresponding model for the input time period  $T_q$ .

### 3.3 Relaxing the Time Boundaries

An important event related to the query entities may have happened very close to the boundaries of the time period of interest  $T_q$ . This means that two entities might be highly related some time before or after  $T_q$ . To cope with this problem, the similarity score can also consider the Word2Vec models for the time periods before and after  $T_q$ . Let  $T_q^{-1}$  and  $T_q^{+1}$  be the time periods of granularity  $\Delta$  before and after  $T_q$ , respectively. The similarity score between a query entity  $e_q \in E_q$  and a candidate entity  $e_c \in E_c$  can now be defined as follows:

$$Sim'(e_q, e_c, T_q) = Sim(e_q, e_c, T_q) \cdot w_1 + Sim(e_q, e_c, T_q^{-1}) \cdot w_2 + Sim(e_q, e_c, T_q^{+1}) \cdot w_3 \quad (2)$$

where  $w_1$ ,  $w_2$  and  $w_3$  are the weights of the models of  $T_q$ ,  $T_q^{-1}$  and  $T_q^{+1}$ , respectively, with  $w_1 + w_2 + w_3 = 1.0$ .

This modification can increase the ranking of an important candidate entity that co-occurs frequently with a query entity some time before or after  $T_q$ , but at the same time can decrease the ranking of an entity co-occurring with a query entity during  $T_q$ . Thus, we should avoid using a very small  $w_1$  value.

## 4 Evaluation Results

Our objective is to evaluate the effectiveness of the proposed approach and compare it with similar but time and entity agnostic models. We use the dataset and ground truth provided by [17] for the problem of time-aware entity recommendation<sup>3</sup>. The dataset provides candidate entities and relevance judgments for 22 keyword queries, where each query corresponds to a particular date range (month). The dataset also provides a set of more than 8M news articles spanning a period of 7 months (Jul'14-Jan'15), annotated with Wikipedia entities. For each of the keyword queries, we manually specified the corresponding entities (since, in our problem, the input is an unambiguous entity URI). For instance, for the query *Tour de France Nibali* (with date range 07/2014), the query entities are: *Tour de France* ([https://en.wikipedia.org/wiki/Tour\\_de\\_France](https://en.wikipedia.org/wiki/Tour_de_France)) and *Vincenzo Nibali* ([https://en.wikipedia.org/wiki/Vincenzo\\_Nibali](https://en.wikipedia.org/wiki/Vincenzo_Nibali)). We also remove the query entities from the set of candidate entities since these are the input in our entity relatedness problem.

We build 7 CBOW models using the proposed approach for each month from Jul'14 to Jan'15 by considering only the articles in English and using the default

<sup>3</sup> <http://km.aifb.kit.edu/sites/ter/>

Word2Vec setting: 300 dimensions, 5 words window size, 5 minimum word count (as also used in [9]). We also experimented with varied dimension (300, 400, 500, 600, 700) and context size (5, 8, 10) but the results were not significantly improved. Then, we compare the effectiveness of our approach on ranking the candidate entities with two *entity-agnostic* baselines (using the same setting): i) one that considers word embeddings computed from the entire collection of documents (i.e., time-agnostic), and ii) one that considers month-wise word embeddings (i.e., time-aware).<sup>4</sup>

Table 1 shows the results of normalized Discounted Cumulative Gain (nDCG) [7] for different top-k lists, without considering time boundary relaxation (i.e., for  $w_1 = 1.0$ ). Regarding the two entity-agnostic baselines, we notice that the time-aware modeling outperforms the time-agnostic one in all the cases. This improvement is statistically significant for all values of  $k$  apart from  $k = 5$  (paired t-tests, a-level 5%). Regarding the proposed model (time and entity aware), the evaluation results show that it outperforms the baselines in all the cases, while the improvement is statistically significant for all values of  $k$ . This justifies that this model solves the problem of ambiguity that occurs due to the variations of different entity mentions.

**Table 1.** Evaluation results ( $\ddagger$  indicates statistically significant improvement).

nDCG@k	Time+Entity Agnostic	Entity Agnostic	Time+Entity Aware
k=5	0.3210	0.3653	<b>0.4999</b> $\ddagger$
k=10	0.3748	0.4113 $\ddagger$	<b>0.5402</b> $\ddagger$
k=20	0.4546	0.4971 $\ddagger$	<b>0.6115</b> $\ddagger$
k=30	0.5092	0.5704 $\ddagger$	<b>0.6562</b> $\ddagger$

Table 2 shows the effect of relaxing the time boundaries for different  $w_1$  values, where  $w_2 = w_3$ . In general we see that the effect of relaxing the time boundaries is very small in almost all cases. We notice that considering only the query time period  $T_q$  provides the best results in all cases apart from  $k = 5$  where  $w_1 = 0.9$  slightly performs better. Moreover, we notice that as the value of  $w_1$  decreases (which means increased consideration of  $T_q^{-1}$  and  $T_q^{+1}$ ), the effectiveness of the model gets worse. This is an expected result since, for the majority of the query entities in the dataset, the important event related to these entities did not happen very close to the time boundaries.

An example for which the relaxation of the time boundaries has a positive impact on the ranking is for the entity *2014 FIFA World Cup*. For  $w_1 = 0.8$ , nDCG@5 increases from 0.45 to 0.51. Notice that the time period of interest for this entity is July 2014, however the tournament started on June 12, 2014. Another example is the query entity *Tim Cook* (the CEO of Apple). For  $w_1 = 0.8$ , nDCG@5 increases from 0.58 to 0.62. Here the query time period is October 2014,

<sup>4</sup> Note that our approach cannot be compared with [17] because this work addresses the different problem of *entity recommendation*, where the input is a *free-text query*, not an unambiguous entity URI like in our case.

however an important event related to Tim Cook happened at the end of the month (he publicly announced that he is gay on October 29).

We see that the relaxation of time boundaries can positively affect entity relatedness when: i) an important event related to the query entity happened very close to the boundaries of the query time period, and ii) the query entity actually corresponds to an event which spans a long time period. Detecting such cases where time boundaries relaxation should be applied is beyond the scope of this paper but an important direction for future work.

**Table 2.** Effect of time boundary relaxation.

nDCG@k	$w_1 = 1.0$	$w_1 = 0.9$	$w_1 = 0.8$	$w_1 = 0.7$	$w_1 = 0.6$
k=5	0.4999	<b>0.5017</b>	0.4990	0.4933	0.4890
k=10	<b>0.5402</b>	0.5332	0.5358	0.5296	0.5291
k=20	<b>0.6115</b>	0.6039	0.5971	0.5932	0.5893
k=30	<b>0.6562</b>	0.6517	0.6451	0.6403	0.6371

## 5 Conclusion

We have proposed a flexible model for entity relatedness that considers time-dependent and entity-aware word embeddings by exploiting the corpus of the underlying application. The results of a preliminary evaluation have shown that the proposed approach significantly outperforms similar but time and entity-agnostic models.

As regards future work, an interesting direction is to extend the proposed method for supporting arbitrary time intervals, which may require joining the results from many models of smaller granularity. However, for supporting very short time periods (e.g., day), this may also require the creation of a large number of Word2Vec models. Regarding the relaxation of time boundaries, there is a need of methods that can identify the most reasonable time window to consider given the query entity and time period (by detecting, for example, periods of increased entity popularity [3]). Finally, we plan to extensively evaluate the effectiveness of our approach and compare it with state-of-the-art entity relatedness methods using a variety of corpora of different contexts and time periods.

## Acknowledgements

The work was partially funded by the European Commission for the ERC Advanced Grant ALEXANDRIA under grant No. 339233.

## References

1. Aggarwal, N., Buitelaar, P.: Wikipedia-based distributional semantics for entity relatedness. In: 2014 AAAI Fall Symposium Series. vol. 192 (2014)

2. 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)
3. Fafalios, P., Iosifidis, V., Stefanidis, K., Ntoutsis, E.: Multi-aspect Entity-centric Analysis of Big Social Media Archives. In: *21st International Conference on Theory and Practice of Digital Libraries (TPDL'17)*. Thessaloniki, Greece (2017)
4. Fang, Y., Chang, M.W.: Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics* **2** (2014)
5. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: Kore: keyphrase overlap relatedness for entity disambiguation. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. pp. 545–554. ACM (2012)
6. Huluş, I., Prangnawarat, N., Hayes, C.: Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In: *International Semantic Web Conference*. pp. 442–457. Springer (2015)
7. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* **20**(4), 422–446 (2002)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
10. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: *Proceedings of AAAI 2008* (2008)
11. Nunes, B.P., Dietze, S., Casanova, M.A., Kawase, R., Fetahu, B., Nejdil, W.: Combining a co-occurrence-based and a semantic measure for entity linking. In: *Extended Semantic Web Conference*. pp. 548–562. Springer (2013)
12. Ponzani, M., Ferragina, P., Chakrabarti, S.: A two-stage framework for computing entity relatedness in wikipedia. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. pp. 1867–1876. ACM (2017)
13. Prangnawarat, N., Hayes, C.: Temporal evolution of entity relatedness using wikipedia and dbpedia. *3rd Workshop on Managing the Evolution and Preservation of the Data Web* (2017)
14. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* **27**(2), 443–460 (2015)
15. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: *AAAI*. vol. 6, pp. 1419–1424 (2006)
16. Tran, N.K., Tran, T., Niederée, C.: Beyond time: Dynamic context-aware entity recommendation. In: *European Semantic Web Conference*. Springer (2017)
17. Zhang, L., Rettinger, A., Zhang, J.: A probabilistic model for time-aware entity recommendation. In: *International Semantic Web Conference*. Springer (2016)