OzU-NLP at TREC NEWS 2019: Entity Ranking

Kenan Fayoumi Özyeğin University kenan.fayoumi@ozu.edu.tr Reyyan Yeniterzi Sabancı University reyyan@sabanciuniv.edu

Abstract

This paper presents our work and submission for TREC 2019 News Track: Entity Ranking Task. Our approach utilizes Doc2Vec's ability to represent documents as fixed sized numerical vectors. Applied on news articles and wiki-pages of the entities, Doc2Vec provides us with vector representations for these two that we can utilize to perform ranking on entities. We also investigate whether background linked articles can be useful for entity ranking task.

1 Introduction

TREC News Track is intended to encourage the use of IR systems to improve the news reading experience of the users. TREC 2019 News Track offers two tasks: Background Linking and Entity Ranking. Background Linking task is concerned with linking a news article with a list of other news articles that provide the reader with context or background information. As for the second task, Entity Ranking, each news article mention a number of namedentities. Some of these entities are of high importance to the article, some are not. For each article, the task is to rank the list of named-entities mentioned in that article, in terms of importance to the article and usefulness to the reader.

This paper investigates two different aspects for entity ranking task. The first is whether representing news articles and entities with distributed vectors can be useful in finding the relation between them. Secondly, to analyze whether background linking articles can be utilized to improve the entity ranking.

The structure of this paper is as follows: Section 2 explains the data collections, Section 3 summarizes the prior work on entity linking from last year's track, Section 4 describes our approach; experiments and results are presented in Section 5 and finally Section 6 concludes the paper.

2 Dataset

In cooperation with the Washington Post, TREC has provided a collection of 595,037 news articles. Originally the dataset had more than 600,000 articles, but these included some duplicates. TREC organizers have removed many of the duplicate articles, but still a number of duplicates remains in the dataset¹. TREC has also provided a Wikipedia dump dated from August 2017 for this task.

The subset of articles used for TREC 2019 News Track Entity Ranking task submission contains 52 articles². Entities mentioned in each of those articles were manually linked into the corresponding Wikipedia article from the Wikipedia dump. We also used last year's (TREC 2018 News Track Entity Ranking) testing set (50 articles) for running experiments and tuning our models.

3 Prior Work on Entity Ranking

TREC News Track Entity Ranking task was started in 2018 and this is the second time TREC is organizing this task. There were 2 participants in last year's task: Team Signal and TREMA-UNH.

Team Signal [1] proposed a supervised approach using Salience Entity Linking (SEL) algorithm [2]. SEL is a unified algorithm for entity linking and salience detection. The authors focused on the salience component of SEL to rank the provided entities.

TREMA-UNH's [3] approach was based on BM25 retrieval model. They experimented with different combinations of article title and paragraphs to rank the entities represented by their Wikipedia pages.

4 Our Approach

In this section we give a brief description of Doc2Vec model and how we use these vectors for our task of ranking entities.

¹http://trec-news.org/guidelines-2019.pdf

²Originally 2019 Entity Ranking testing dataset contained 60 topics/articles. We've been informed by TREC organizers that 8 articles out of these 60 were dropped from the testing set because the entities mentioned in these do not provide any background information or context.

4.1 Doc2Vec

Doc2Vec or document distributed representations is an unsupervised model used to learn a vector representation for sentences, paragraphs, or documents [4]. Since it's unsupervised, it can be trained on large unlabeled text corpora. It is applicable to documents of different lengths as it generates vectors of fixed size for all documents. This is an advantage over other similar approaches which requires input padding or cutting off documents to a certain length. Summarizing documents in one-dimensional vectors, these paragraph vectors encapsulate contextual information along with semantic and syntactic structure.

Doc2Vec was introduced as an extension to Word2Vec [5]. Word2Vec is a model that learns a representation of words and phrases. Word2Vec uses two different approaches to learn the word representations: Continuous Bag-of-Words (CBOW) predicts a target word given its context, and Skip-Gram, given a word, predicts the context of that word. Doc2Vec extends Word2Vec's learning of word representation for the purpose of learning document representation.

As described by Le and Mikolov [4], there are two main algorithms to learn paragraph vectors, either with Distributed Memory (PV-DM) as shown in Figure 1(b) or with distributed Bag-Of-Words (dBOW) as shown in Figure 1(a). The main difference between these two algorithms is that PV-DM uses the context words to predict the next word in the context, while dBOW forces the model to predict randomly sampled words from a sentence without using the context or word-ordering information. In that sense, dBOW is similar to Skip-Gram model in Word2Vec [5]. Since dBOW does not consider word ordering, it's conceptually simpler than PV-DM model and it stores less information. Therefore, from the memory perspective, dBOW has an advantage over PV-DM.



Figure 1: Doc2Vec different frameworks reused from [4]

4.2 Approach description

We train the Doc2Vec model on the Washington Post corpus provided by TREC. All 595,037 news articles are used. Despite the model size advantage of dBOW algorithm over PV-DM, we choose PV-DM as our training framework for Doc2Vec. Through experiments on TREC 2018 topics, we have found that PV-DM outperforms dBOW for this task. As for the embedding size (the size of the vectors) we use 200, which returned good performance with an acceptable training time with TREC 2018 topics.

Trained Doc2Vec model is used to map news articles and Wikipedia pages of entities' existing in these articles into distributed vector representations. Later on, these vectors are used for ranking entities for a given news article.

Below we define the main variables in our system:

D: the news article (TREC topic)

V(D): the Doc2Vec representation of D

 E_D : the provided list of entities $[e_1, e_2, e_3, e_4...]$ that are mentioned in D $V(e_i)$: the Doc2Vec representation of entity e_i 's corresponding Wikipedia article

 $R_{e_i}^D$: estimated rank of entity e_i for article D

Our task is basically to find ranking R_e^D for each e in E using the similarity between vectors V(D) and $V(e_i)$. Cosine similarity is used as the measure to calculate the distance between these vectors. This process is depicted in Figure 2.



Figure 2: Pipeline of ranking entities

Other news articles, which are related to the current news article, can be useful in ranking the entities as well. Therefore, in addition to using only the topic news article itself, we explore using background linked articles in ranking entities. In a realistic setting, since we don't have those background linked articles given explicitly, we again use Doc2Vec vector similarity to fetch the similar articles given a news (topic) article. Based on this we use the followings:

 SIM_1^D : article from Washington News corpus that is most similar to D SIM_2^D : article from Washington News corpus that is the second most similar to D

 $SIM_n^D {:}$ article from Washington News corpus that is the nth most similar to D

Below is the list of submitted runs and their descriptions of how we utilize Doc2Vec in both finding similar articles and using those articles (including the topic article itself) for ranking entities. Our first run only uses the topic article itself to rank the entities while the other four runs also use the other similar news articles given the topic article.

For an article D we would like to evaluate:

1) OzU_wiki: This is the approach shown in Figure 2. Get the Doc2Vec representations of the (topic) news article and entity's Wikipedia page, and calculate the cosine similarity between them to estimate the ranking score of the entity.

$$R_{e_i}^D = \cos_sim(V(D), V(e_i)) \tag{1}$$

2) OzU wiki top1: We fetch the most similar article to D which is (SIM₁^D), and then compute cosine similarity between this similar article and entities of the original queried article D.

$$R_{e_i}^D = \cos_sim(V(SIM_1^D), V(e_i))$$
⁽²⁾

This submission shows the effect of using the most similar news article instead of the query article.

3) OzU_wiki_top5: We first fetch the vectors of the 5 most similar articles $(SIM_1^D, SIM_2^D...SIM_5^D)$ to article D and then calculate cosine similarity between each similar article and the entities. $R_{e_i}^D$ is the average of these similarities.

$$R_{e_i}^D = \frac{1}{5} \sum_{n=1}^{5} \cos_sim(V(SIM_n^D), V(e_i))$$
(3)

4) **OzU_wiki_1_ws:** This is a combination of OzU_wiki and OzU_ wiki_top1. We use similarity scores from these approaches to calculate $R_{e_i}^D$ in a weighted sum fashion using the following formula:

$$R_{e_i}^D = 0.9 * \cos_sim(V(D), V(e_i)) + 0.1 * \cos_sim(V(SIM_1^D), V(e_i))$$
(4)

5) **OzU_wiki_5_ws:** This is a combination of OzU_wiki and OzU_ wiki_top5. We perform a weighted sum to calculate $R_{e_i}^D$ giving higher weights to articles that are more similar.

$$\begin{aligned} R^{D}_{e_{i}} &= 0.8 * cos_sim(V(D), V(e_{i})) \\ &+ 0.05 * cos_sim(V(SIM^{D}_{1}), V(e_{i})) \\ &+ 0.05 * cos_sim(V(SIM^{D}_{2}), V(e_{i})) \\ &+ 0.04 * cos_sim(V(SIM^{D}_{3}), V(e_{i})) \\ &+ 0.03 * cos_sim(V(SIM^{D}_{4}), V(e_{i})) \\ &+ 0.03 * cos_sim(V(SIM^{D}_{5}), V(e_{i})) \end{aligned}$$

Weights used in Equation 4 and 5 are optimized through experiments on TREC 2018 Entity Ranking testing dataset.

5 Experiments

The primary metric for evaluating entity ranking task is nDCG@5. MAP is also being used as an additional metric. Table 1 presents the results of our first run (OzU_wiki) on both TREC 2018 and 2019 entity ranking task topics together with the results of the other submitted runs from TREC 2018, more specifically the runs from Team Signal and TREMA-UNH.

	TREC 2018		TREC 2019	
Run	MAP	nDCG@5	MAP	nDCG@5
OzU_wiki	0.8184	0.7232	0.7906	0.7579
signal-ucl-slst	0.6894	0.5772	-	-
signal-ucl-sel	0.7158	0.6071	-	-
signal-ucl-eff	0.7144	0.6084	-	-
UNH-ParaBm25Ecm	0.6828	0.3261	-	-
UNH-ParaBm25	0.7859	0.4278	-	-
UNH-TitleBm25	0.7741	0.4220	-	-

Table 1: Results of OzU_wiki run

According to Table 1, our proposed approach of using Doc2Vec similarity to rank entities outperform all approaches from last year. This indicates that distributed representation of documents is effective for ranking entities in general.

Two submissions from TREC 2018 have different strengths. One performed better in MAP while its performance on NDCG@5 is much lower. On the other hand, the other performed better in NDCG@5 but not so well in MAP. Compared to these two, Doc2Vec based Ozu_wiki approach outperform both systems in both metrics which shows the generality of the approach.

	TREC 2018		TREC 2019	
Run	MAP	nDCG@5	MAP	nDCG@5
OzU_wiki	0.8184	0.7232	0.7906	0.7579
OzU_wiki_top1	0.8023	0.6913	0.7812	0.7509
OzU_wiki_top5	0.8154	0.6954	0.7785	0.7426
OzU_wiki_1_ws	0.8143	0.7161	0.7551	0.7287
OzU_wiki_5_ws	0.8199	0.7234	0.7561	0.7319

Table 2 summarizes all of our runs including the ones in which similar news articles to the given news article are also used for ranking the entities.

Table 2: Results of all of our runs

According to Table 2, using similar news articles either alone or interpolated together does not provide any significant gains or loses. There is not any single approach that consistently outperforms others in both TREC 2018 and 2019. This may be due to our approach on finding similar articles. Since both document similarity and entity similarity is using the same Doc2Vec model, identified similar news articles may not provide additional useful information to the entity ranking. Therefore, comparing entity vectors with vectors from similar articles return similar results.

The motivation behind using these similar news articles was to see whether background linking can be useful in entity ranking task. However, Doc2Vec may not be the ideal way to identify background linked articles. One way to understand this is to see how Doc2Vec performs on background linking task. Therefore, we evaluate the similar news articles identified through Doc2Vec with background linking articles' gold standard qrels. The performance is not very high and even lower than some other experimented approaches from TREC 2018. Doc2Vec is not the best way to identify background linking articles and that may be the reason why we do not get any improvements with entity ranking as well.

In order to test whether using actual background linked articles (from qrels) can be useful in entity ranking, we apply the same four approaches $(topn \text{ or } n_ws)$ with those articles' vectors. The results do not change significantly. This shows that even using the actual background linking articles in entity ranking through Doc2Vec is not effective.

6 Conclusion

In this paper we present our work and experiments for the entity ranking task. We train a Doc2Vec model and use it to generate vector representation for news articles and their entities' wikipages. Our original approach OzU_wiki, where we rank entities using the cosine similarity between their wiki-page vector and the query article's vector, return promising results by surpassing all of last year's approaches. Furthermore, we investigate whether the use of similar articles can help boost our model. Through many experiments, we conclude that using similar or background-linked articles through Doc2Vec doesn't bring any noticeable improvements to our original model.

References

- D. van der Sluis, D. Albakour, and M. Martinez, "Signal at trec 2018 news track.," in *TREC*, 2018.
- [2] S. Trani, D. Ceccarelli, C. Lucchese, S. Orlando, and R. Perego, "Sel: A unified algorithm for entity linking and saliency detection," in *Proceedings* of the 2016 ACM Symposium on Document Engineering, DocEng '16, pp. 85–94, ACM, 2016.
- [3] S. Kashyapi, S. Chatterjee, J. Ramsdell, and L. Dietz, "Trema-unh at tree 2018: Complex answer retrieval and news track.," in *TREC*, 2018.
- [4] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, pp. 1188–1196, 2014.
- [5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, pp. 3111– 3119, 2013.