TUW-IFS at TREC NEWS 2020 Wikification Task

Annisa Maulida Ningtyas

TU Wien / Vienna, Austria

 ${\tt annisa.ningty} as {\tt @student.tuwien.ac.at}$

Alaa El-Ebshihy

TU Wien / Vienna, Austria alaa.el-ebshihy@tuwien.ac.at

Florina Piroi

TU Wien / Vienna, Austria florina.piroi@tuwien.ac.at

Linda Andersson

Artificial Researcher IT GmbH / Vienna, Austria linda.andersson@artificialresearcher.com

Allan Hanbury

TU Wien / Vienna, Austria allan.hanbury@tuwien.ac.at

Abstract

This paper presents our work and submission to the TREC 2020 News Track on Wikification. This task aims to link word and phrases to entities in external knowledge bases, such as external news articles or Wikipedia. The idea behind this task is to increase user reading comprehension by linking entities in the text they are reading to external knowledge bases that contain information on those entities. In order to perform the Wikification task, we have built a processing pipeline consisting of three main parts: identification, candidate generation and disambiguation, and candidate pruning or nil detection. Our model is fully unsupervised, and it does not require data training. We use commonness, topic modeling distance, and embedding similarity as a disambiguation method.

The evaluation of the task submissions was done using newly introduced metrics: correctness of the link, the usefulness of the target link, the usefulness of the anchor span and the sensibility of the anchor text. Our method was designed to focus on coverage to also support the readers with limited knowledge in English, and thereby requiring more explanations and the possibility to have the explanation in another language. Based on the result, we conducted a failure analysis. We infer that our low scores on both evaluation are due to the use of Wikipedia as the only knowledge base to linked to, and lack of entity ordering during our final processing.

1 Introduction

The TREC 2020 News Track is proposed to encourage the use of the IR system to enhance the reading comprehension of the users. The TREC 2020 News Track had two tasks -

Background Linking and Wikification. The Background Linking task focused on linking a news article with a list of other news articles that provide useful context and background information to the reader. The second task, Wikification aimed to find hyperlinks from concept words or phrases in news articles to an external knowledge base. Wikipedia is one of the popular examples of such a knowledge base, though other external resources, such as news articles, can give meaningful insight to the reader.

This paper focuses on the Wikification task. The task of wikifying an input document can be divided into several interrelated sub-tasks or processing phases: (1) word and phrase identification (called *mentions*), in the input document, that are to be candidates to be linked to into a knowledge base; (2) detect the knowledge base entries (called *entities*) that semantically match the identified words and phrases selected in the 1st sub-task, and retain their links; (3) from all pairs of *entity* detected in the previous two sub-tasks, perform word sense disambiguation to determine which ones are the correct and useful entities to be linked to from the *mention*.

The general pipeline we developed has two main modules that address these sub-tasks. The first module handles *mention* identification and aims to spot entities from the given documents, that are candidates to be linked into a knowledge base. The second module first syntactically matches the *mentions* (word or phrases) extracted in the first phase with entries in the knowledge base (in this work, Wikipedia). Then it will apply a disambiguation mechanism to eliminate the matched knowledge base entries that do not, semantically, refer to the *mentions* they syntactically matched. The disambiguation step is an important one, since it must remove the Wikipedia *entities* that only syntactically match a *mention*, but not semantically. In the disambiguation step we use features like "commonness", "topic distance" and "embedding similarity" to decide if an *entity* is semantically matching a *mention*. Once this is done we apply a probability filter on the remaining *emention*, *entity* pairs that estimates which of them is most probably of use to a reader.

Our method is designed to focus on coverage (more *entities* linked) to support the readers with limited English knowledge, thereby requiring more explanations possibly having access to explanation in another language.

The structure of this paper as follows: Section 2 gives a brief description of the related work. Section 3 gives a very brief account on the dataset used in this Wikification task. Section 4 describes the methodology chosen for this work. Section 5 presents the experiments and the results given by our method. Lastly, in Section 6 we draw some conclusions about our work.

2 Related Work

The research on Wikification (or entity linking), can be divided into using two types of models: end-to-end and disambiguation-only. The end-to-end entity linking models process the input text to extract mentions (i.e Named Entity Recognition) and then disambiguate this mention to point to the correct entity in the knowledge base. The disambiguation-only methods focus mainly on the later part.

Wikification is a task for extracting important words or phrases from a document and link them to the appropriate entities in a knowledge base, such as Wikipedia. In one of the early works on this task, Mihalcea and Csomai [8] divided the Wikification task into two task: keyword extraction and words sense disambiguation. To extract keywords that are to be linked to a knowledge base, the authors implemented a dictionary matching algorithm and for each match a rank of dictionary entries are given, to select the most salient mentions. In their word sense disambiguation step, the authors computed the contextual overlap between the contexts of the entities and those of the mentions. They also use machine learning approach, which is a Naive Bayes Classifier. The train data is a combination of feature vectors such as part-of-speech and surrounding words as a local context, and five most frequent word in the document as a global context. The target data is the correct entity among the ambiguous candidate.

With the advancement of deep neural network applied to language processing tasks, entity linking has also been one of the application areas [3,4,7,9,11,15]. In entity disambiguation, neural networks act to represent *mentions* as continuous vectors with the context awareness. These representation can be achieved with with Convolutional Neural Networks (CNN) [9], Long Short-Term Memory Networks (LSTM) [3], and Attention mechanisms [7].

In end-to-end models, mention detection and entity disambiguation can be treated as inference step in the trained model of neural network [4,11]. However, these approaches rely on big training datasets to build their model.

In contrast, a light-weight linguistic approach [15,16] that uses classical text processing and exploits the graph structure of the knowledge base can still reach high performance levels. The approach describes by Sakor et.al [15] fuse the knowledge graph of several datasets to increase the size of the available knowledge base for linking. The recent classical approach also described by Shnayderman et.al [16] that mainly uses Wikipedia redirects as a basis for entity linking. The classical approaches prove to be fast with respect to training time when compared to the use of Neural Network based models [16], and can cover multiple domains [15]. In this paper, we propose a simple linguistic processing for mention detection and local disambiguation with topic modeling and simple semantic similarity.

3 Datasets

The TREC News track organizers provided the participants with two collections of data. The first one, in collaboration with The Washington Post, contains 671,947 news articles and blog posts starting January 2012 through December 2019 ¹. The second dataset is an English Wikipedia Dump generated on January 1st, 2020. There were no topics and relevance assessments to train on from the previous years. This year the track organizers have created 50 topics² selected from the set of news articles which the track participants should wikify.

¹https://trec.nist.gov/data/wapost/

²http://trec-news.org/guidelines-2020.pdf

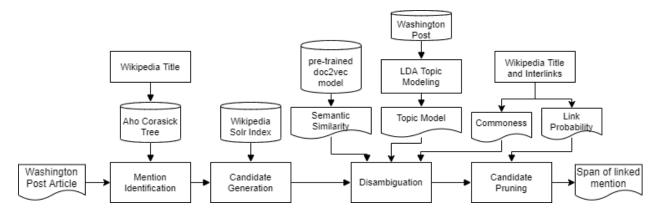


Figure 1: General Workflow of the system we used to solve the Wikification task.

4 Method

In this section, we describe the pipeline we used to approach the wikification task. As we mentioned in the introduction, our solution is divided into three phases: *mention* identification, matching Wikipedia *entities* to the *mentions* (with an additional disambiguation step), and a ranking of *entity* pairs to select those that are of most use to a reader.

4.1 Mention Identification

In the first phase of our approach we aim to identify words and sequences of words (phrases) in a given text that are candidates to be linked to a knowledge base. The knowledge base we use in this work is Wikipedia. We start, here, by building a dictionary of Wikipedia article titles, words or phrases, which we use as a look-up index to identify good candidates to be linked from the input news article. We do lower casing on the Wikipedia titles before adding them to the dictionary. The dictionary is constructed with the Aho-Corasick algorithm which uses a tree-like structure to store it. The aim of this method is to find words or phrases in the Washington Post articles that match the pre-built dictionary. If the matched word is a subset of another matched words or phrases, we will select the longest string in the news article that matches as a mention.

4.2 Candidate Generation and Disambiguation - to change

For this phase we have created a Solr index containing the Wikipedia articles in the TREC News Wikification corpus. In a first step we take the *mentions* obtained in Section 4.1 and send them as queries to the Solr index, where we apply the Jelinek-Mercer smoothing language model to the queries. We retrieve, thus, those Wikipedia articles for which we have an exact match between the *mention* and the Wikipedia article's title. For each *mention* we retain the top 10 results returned by the Solr index.

The next step in this phase aims to disambiguate between (matched) Wikipedia articles by adding a semantic component to the process. That is, we want to avoid proposing the user a link to a Wikipedia article about 'Pat Nixon' when the news article mentions 'Nixon'

and it is clear from the context that the article writes about 'President Nixon' or 'Richard Nixon'.

Previous work on disambiguation classifies disambiguation features into local and global features [12,14]. Local features are, for example, 'commonness' and 'topic distance', and use only the surrounding context of a *mention* in a sentence or paragraph. Global features take into account the semantic context of a *mention* in the whole document. In the system proposed by us, we only use local features to disambiguate each *mention*, namely 'commonness', 'topic distance' and 'embedding similarity.' We combine these metrics to calculate a final disambiguation score using a weighted sum of these features in order to select an appropriate Wikipedia *entity* in the knowledge base for each selected *mention* in the documents.

4.2.1 Commonness

A mention can be used as an anchor text for entities, such as Wikipedia pages. A mention, **m**, can refer to different entities, **e**, depending on the context of the mention. Commonness is defined to be a prior probability for a mention, **m**, to a specific entity, **e**, [13]:

$$P(e|m) = \frac{freq(m \mapsto e)}{\sum_{e' \in W} freq(m \mapsto t')}$$
 (1)

where $freq(m \mapsto e)$ is the number of *mentions*, m, that are a hyperlink for the *entity*, e, and $freq(m \mapsto e')$ is the number of *mentions*, m, that are also a title in Wikipedia. Commonness is not a robust metric across domains and results may be misleading when we disambiguate a *mention* as it does not take into consideration the context of the document [13].

4.2.2 Topic Modeling

We use topic modeling a local feature in our disambiguation step. As commonness is context-independent, we use topic modeling to estimate the topic similarity between the input document and documents in the knowledge base. We build a topic modelling using Latent Dirichlet Allocation (LDA) on The Washington Post news articles, and use it to capture the topic distribution over the Washington post news articles. This model will be used for inference process in the disambiguation step. We transform the articles from Washington Post as well as Wikipedia to become a vector of topic distribution using the topic model. The topic distribution of Wikipedia and Washington Post articles, will be used to measure the topic similarity between the Washington Post and the Wikipedia articles.

The basic intuition of the LDA is that every document consists of a mixture of topics, that can be seen as a topic distribution per document. Blei [2] defines topics as a probability distribution over of a set of words. For example, the 'politics' topic has words about politics with high probability and words about sports with low probability, while a 'sports' topic is related to words about sports with high probability, but words referring to music have a low probability in a document about sports. We assume that document topics have been specified before corpus documents were released.

To implement our topic modeling method we use *gensim* and run it on the 50 topics of the Washington Post track. We use these topics to disambiguate between a *mention*, **m**, and an *entity*, **e**. We compare the paragraph where the *mention* occurs (**pm**) with the first

paragraph of each *entity*, **e**, initially associated with **m**. We, then, calculate the distance between the two probability distribution using the Hellinger Distance (HD) [1]. By definition, the Hellinger Distance between two probability distributions, p and q, that represent two topics, is defined as (3):

$$H(p,q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n_v} (\sqrt{p_i} - \sqrt{q_i})^2}$$
 (2)

p is the topic distribution detected on the Washington Post article' paragraphs, and q is the topic distribution detected on the first paragraph of the Wikipedia target article. The minimum H(p,q) distance is 0 when p is equal to q. The maximum Hellinger Distance between two probabilities is 1.

4.2.3 Doc2Vec

An additional feature that we use in the disambiguation step is a score derived from applying Doc2Vec. This step aims to calculate the semantic similarity between the Washington Post paragraph which contain the *mention*, with the first section of Wikipedia articles.

Doc2vec is an unsupervised algorithm that learns feature representations from texts, such as sentences, paragraphs, and documents. Doc2Vec represents each document as a dense vector of real numbers.

There are two main algorithms to learn paragraph vectors, namely Paragraph Vector - Distributed Memory(PV-DM) and distributed Bag-of-Words (dBOW) [6].

We use the pre-trained model created by Lau and Baldwin [5] which is built using the gensim's doc2vec library. This model is trained on the English Wikipedia corpus using a dump dated December 1st, 2015. We use this model to infer the paragraph context of the mention, **m**, as well as each of the paragraph of candidate entity, **e**. Then, we calculate the similarity between them using a cosine measure and select the most similar pairs of paragraph context and candidate entity.

4.2.4 Scoring

Each of the three processing steps described above gives back a score between 0 and 1. We combine all these scores with a weighted sum to obtain a final ranking of the pairs < mention, entity>:

$$score = w_{cm} * P(e|m) + w_{td} * (1 - H(p,q)) + w_{es} * sim$$
 (3)

where w_{cm} is the commonness weight, P(e|m) is a prior probability of a specific *entity*, **e**, given a *mention*, **m**, wtd is the topic model weight, H(p,q) is the distance between two topic distributions (p for the the paragraph topic distribution in the topic document, and q for the topic distribution of the first *entity* paragraph), w_{es} the doc2vec weight, and sim is the cosine similarity between the mention's context and the Wikipedia page (i.e. the *entity*). We experimented with different values for w_{cm} , w_{td} , and w_{es} , which we report on in the next subsection. Using this score we order the pairs $< mention, wiki_link >$ and take the top 10 and pass them on to the next phase.

Table 1: Run Statistics

Runs	Weight			Nil	Number of Entities Retrieved per Topic				
Tulis	211	au -	211	1111	Mean	Median	Min	Max	Standard
	w_{cm}	w_{td}	w_{es}						Deviation
tuw-ifs-1	0.5	0.25	0.25	No	43.26	40	18	89	17.83
tuw-ifs-2	0.4	0.3	0.3	No	42.96	38	18	89	17.92
tuw-ifs-3	0.5	0.25	0.25	Yes	26.28	23	7	61	12.05
tuw-ifs-4	0.7	0.2	0.1	Yes	26.06	23	7	59	11.96

4.3 Candidate Pruning

From the *<mention*, *entity>* pairs that were retained in the previous processing steps, the last phase of our method will select those that are estimated indeed relevant to a user. In our interpretation, the Wikification task aims to provide useful information to a reader, contributing to the reader's comprehension skills.

For each of the < mention, entity> pairs we compute the link probability of a mention to be attached a Wikipedia link. With this probability we prune the irrelevant semantic annotations from the < mention, entity> pairs. The formula by which we compute this is as follows [10]:

$$lp(m) = \frac{link(m)}{freq(m)} \tag{4}$$

where link(m) represents how many times the *mention*, **m** has been selected as an anchor text in a Wikipedia page, while freq(m) is the number of times the *mention*, **m** appears in the Wikipedia page.

5 Results

In this section we present the outcome of the Wikification submission. The statistics on retrieved entities in our submissions are shown in Table 1. We ran the experiments with various weights and turning candidate pruning on or off for each run. The weights are commonness (w_{cm}) explained in Eq. 3, topic distance (w_{td}) explained in Eq. 3 and embedding similarity (w_{es}) . This score is calculated based on the Eq. 3. The NDCG@5 can be seen in Table 2. NDCG is the main measure indicated by the the TREC Wikification guidelines. Thus, the ranking for a relevant entity will be taken into measure. However, the NDCG@5 results were not satisfying. This is because we did not re-rank the entities in the final list of extracted entities. Since our method does not directly produce a ranking, we listed the mentions and their attached Wikipedia entities in the order of the character position in the document. The overall NDCG results are shown in Fig.2. On this graph, we observe that the performance is better for higher NDCG cuts. In Table 2, we can see that our best run is tuw-ifs-4. As stated in the previous section, commonness plays a big role in determining the correct entity to be associated to a mention.

Table 2: NDCG@5 Statistics

	NDCG@5								
Runs	Mean	Median	Standard	Min	Max				
			Deviation	101111					
tuw-ifs-1	0.0990	0.0000	0.1480	0	0.4704				
tuw-ifs-2	0.0477	0.0000	0.1118	0	0.5087				
tuw-ifs-3	0.1121	0.0131	0.1633	0	0.4913				
tuw-ifs-4	0.1191	0.0140	0.1659	0	0.4913				

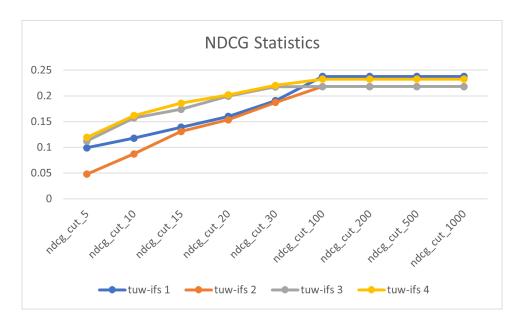


Figure 2: NDCG Statistics

The organizers of the track also provided four new metric to asses the correctness of the Wikification task. These measures are: (1) The correctness of the link, (2) the usefulness of the target link, (3) the usefulness of the anchor span, and (4) the sensibility of the anchor text. The result is in line with the earlier assessment, which shows a low performance. From 47 topics (three were left out by the track organizers for lack of mentions to be linked to Wikipedia), there are 20 topics which have a zero median, with 0.078 on the average of median. This means that that half of our results that were checked by the track assessors didn't meet the target.

5.1 Failure Analysis

The main findings of our failure analysis is that the low scores we obtained are due to our omission to order the results according to their relevance. We also only use Wikipedia as the candidate's knowledge base, while the entity-linking task may be linked to another sources. In the new measure we found the usefulness of the target link is including the background information, which sometimes is not captured in Wikipedia.

By looking at the worst-performing topic (topic id 899), we noticed ambiguities in handling entities representing person's proper names. For example, 'Ben Penrod' was linked to 'Guy Penrod.' This is due to the high commonness of the entity. This happened for other topics, too, which reduced our system's efficiency. Thus, we conclude that candidate pruning can be counter-productive if it only uses commonness as a parameter.

6 Conclusions

This paper presents our work and our submission for the TREC 2020 News Track on Wikification. For this task we constructed a pipeline that consisted of 3 main phases: mention identification, candidate generation and disambiguation, and candidate pruning. Our experiments show that our system setting has a high recall, but lower precision. This is seen in the results of the evaluation on the terms of the NDCG and the new measures used by the track organisers.

The main findings of our failure analysis are using Wikipedia only as a knowledge base, not ordering the entity according to the relevance ranking in submission file, and only using commonness as a parameter for candidate pruning. In a future iteration of this system we plan to use use the relevance assessments of this year and improve our model. We also consider using supervised model for the disambiguation task.

References

- [1] Beykikhoshk, A., Arandjelović, O., Phung, D., and Venkatesh, S. Discovering topic structures of a temporally evolving document corpus. *Knowledge and Information Systems* 55, 3 (2018), 599–632.
- [2] Blei, D. M., Edu, B. B., Ng, A. Y., Edu, A. S., Jordan, M. I., and Edu, J. B. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [3] Gupta, N., Singh, S., and Roth, D. Entity linking via joint encoding of types, descriptions, and context. *EMNLP 2017 Conference on Empirical Methods in Natural Language Processing, Proceedings* (2017), 2681–2690.
- [4] KOLITSAS, N., GANEA, O.-E., AND HOFMANN, T. End-to-End Neural Entity Linking. CoRR abs/1808.0 (2018).
- [5] Lau, J. H., and Baldwin, T. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. 78–86.
- [6] Le, Q., and Mikolov, T. Distributed representations of sentences and documents. 31st International Conference on Machine Learning, ICML 2014 4 (2014), 2931–2939.
- [7] LOGESWARAN, L., CHANG, M. W., LEE, K., TOUTANOVA, K., DEVLIN, J., AND LEE, H. Zero-shot entity linking by reading entity descriptions. *ACL* 2019 57th

- Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (2020), 3449–3460.
- [8] MIHALCEA, R., AND CSOMAI, A. Wikify! Linking Documents to Encyclopedic Knowledge.
- [9] Murty, S., Verga, P., Vilnis, L., Radovanovic, I., and McCallum, A. Hierarchical losses and new resources for fine-grained entity typing and linking. *ACL 2018* 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) 1 (2018), 97–109.
- [10] PICCINNO, F., AND FERRAGINA, P. From Tagme to WAT: A new entity annotator. ERD 2014 - Proceedings of the 1st ACM International Workshop on Entity Recognition and Disambiguation, Co-located with SIGIR 2014 (2014), 55–61.
- [11] RAIMAN, J., AND RAIMAN, O. DeepType: Multilingual entity linking by neural type system evolution. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 (2018), 5406–5413.
- [12] RATINOV, L., ROTH, D., DOWNEY, D., AND ANDERSON, M. Local and Global Algorithms for Disambiguation to Wikipedia. 1375–1384.
- [13] ROTH, D., JI, H., CHANG, M., AND CASSIDY, T. Wikification and Beyond: The Challenges of Entity and Concept Grounding. *ACL tutorial* (2014).
- [14] Sajadi, A. Semantic Analysis Using Wikipedia Graph Structure. PhD thesis, 2018.
- [15] SAKOR, A., MULANG, I. O., SINGH, K., SHEKARPOUR, S., VIDAL, M. E., LEHMANN, J., AND AUER, S. Old is gold: Linguistic driven approach for entity and relation linking of short text. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference 1 (2019), 2336–2346.
- [16] Shnayderman, I., Ein-Dor, L., Mass, Y., Halfon, A., Sznajder, B., Spector, A., Katz, Y., Sheinwald, D., Aharonov, R., and Slonim, N. Fast Endto-End Wikification.