Evaluation of related news recommendations using document similarity methods

Marko Pranjić*§, Vid Podpečan†, Marko Robnik-Šikonja‡, Senja Pollak†

* Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia

§ Trikoder d.o.o.
Ulica Miroslava Miholića 2, 10010 Zagreb, Croatia
marko.pranjic@styria.ai

† Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
vid.podpecan@ijs.si senja.pollak@ijs.si

‡ University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, 1000 Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si

Abstract

A set of related articles is a useful addition to the newly published news. Such news articles contain more context and background information and provide a richer experience to the reader. Currently, the work of finding related articles is often done manually by the journalists writing the news story. The process can be automatized by suggesting relevant articles based on the similarity with the new article. We compare several link recommendation methods on the news archive of popular Croatian website 24sata. Our results show that the tf-idf weighting applied to bag-of-words document representation offers better matching with manually selected links by journalist than more sophisticated approaches, such as latent semantic indexing, doc2vec, and multilingual contextual embeddings BERT and XLM-R.

1. Introduction

Modern technology is changing the journalism and readers are developing new habits and expectations. Media houses throughout the world are embracing the opportunities of digital publishing and distributing the content on the web and through mobile applications. A common feature of online news are references (i.e. links) to other relevant news articles that provide more context or relevant background information. This makes other content, relevant for the current story, more accessible to readers, while media houses benefit from more efficient use of existing content and improved business metrics such as user engagement and the total time spent on a site.

Large media houses employ modern natural language processing (NLP) technologies to ease the task of content production and improve readers' satisfaction. Currently, advanced technological support to journalists is limited to large media houses in high-resource languages like English. In other cases, the task of finding and recommending related news articles remains the job of journalists writing the news story. The work described in this paper was done within the context of the EU H2020 EMBEDDIA project[1], whose objective is to develop and adapt state-of-the-art NLP technologies to less-resourced languages, and test them in real-world news and media production contexts.

In this paper, we compare several existing NLP methods that can be used to automate search for the related news. As the working dataset we use articles from 24sata[2], a major Croatian news web portal. We approach the problem with several text embedding methods and evaluate semantic similarity between articles using the cosine similarity on the selected representations.

The paper consists of seven sections. In Section 2, we present the related work on linking related news. In Section 3, we present the dataset of news articles used in this evaluation and its preprocessing. Section 4 describes the methods used. In Section 5, we describe the experimental settings, and in Section 6, we present the results. In Section 7, we conclude the work and suggest ideas for further work.

2. Related Work

Various research has explored linking news articles with the additional information. A task of finding background news given a Twitter message is introduced in Guo et al. (2013) and is performed using a named-entity recognition, matrix factorization methods, and graph analysis. In Gamon et al. (2008) and Ikeda et al. (2006), systems for linking news article with blog posts are described.

In recent years, there was an increase in the interest for document linking in the context of the news articles. In

[1] http://embeddia.eu/

[2] http://www.24sata.hr

2018, NIST in a partnership with the Washington Post, created a new TREC track known as the News Track (Huang et al., 2018; Soboroff et al., 2018). The main task of the News Track is the Background Linking task, defined as a problem of retrieving news articles that provide important context or background information that helps readers to better understand the query article. A number of researchers evaluated approaches to linking the query article with the background information. They attempted keyword extraction coupled with reranking on the number of matched paragraphs (Lu and Fang, 2018), graph-based analysis (Essam and Elsayed, 2019), or named-entity recognition (Bimantara et al., 2018). In Foley et al. (2019), authors evaluated approaches using keyphrase detection, clickbait classification and topic detection. In Lu and Fang (2019) and van der Sluis et al. (2018), authors describe systems for background linking of the news articles based on the entity linking. Most of those methods extract parts of text and use them as a query instead of comparing the whole document. An alternative approach is to represent the entire document as a vector in a semantic vector space. In this way, related documents can be found in the local neighborhood of the query (Soloshenko et al., 2015; Sitikhu et al., 2019). This is the approach we focus on in this paper.

3. Dataset description

The data used for analysis contains news articles from *24sata*, the biggest Croatian news publisher. The dataset contains 546,801 articles published online between 2007-03-12 and 2019-04-24. The length of news articles in the datasets vary from 11 to 19,695 words, with an average length of 272 words. The distribution of article lengths is depicted in Figure 1. All articles are written in Croatian language and each entry contains the **title**, **lead_text**, **content** of the article, **author**, content **tags**, the **section** of the newspaper where the article appears, **published_date** (the date when the article was published on the website), **created_date**, (the date when the article was originally written), and **related_articles** containing a list of references to other articles. The *published_date* and *created_date* can differ as some articles are written in advance to be published at a later time. The list of related articles is chosen by the journalist when the article is written and links to related articles are embedded in the content of the article when the article is published. In this paper, this information about related articles is used as a ground truth for evaluation of different related news recommendation methods.

Articles in our collection published before 2017-08-16 do not have any related article assigned, and also many later articles are missing this information. In total, there are 35,289 articles that have some *related_articles* assigned to them.

The dataset was split in such a way that 75% of the articles with the oldest *created_date* are available for training, and the rest form the testing set. The *related_articles* links for items that are found in the test set but reference news articles in the training set are discarded. Among 410,101 articles in the training set, there are 2,654 with related articles. Those related articles reference only articles from the training set and we use them to finetune the mBERT and XLM-R models as well as to determine the hyperparameters for Doc2Vec model. In the remaining 25% of the data used as the test set, there are 32,635 items with related articles and all their references are part of the test set.

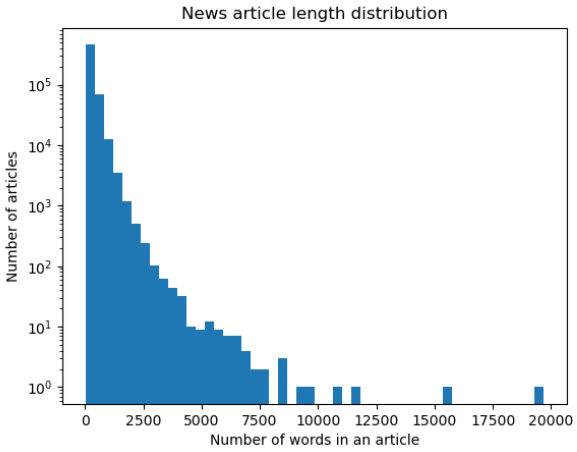

Figure 1: Distribution of number of words for the news articles in the dataset. The number of articles is shown on the logarithmic scale.

3.1. Data preprocessing

The data contains several attributes such as tags, sections, author names etc. which might provide relevant additional information and improve document retrieval. For example, focusing the search to a subset of articles from the same section or to articles containing matching tags should improve results if such information is available during the model inference. Another requirement present in the real system is to take into account the date of the article and preferably return newer articles. In this paper, our goal is to compare the methods working on text and not on evaluating the whole system for linking the background information so this supporting metadata is ignored.

The text of an article is spread between title, lead and content fields and the first step was to concatenate these three fields. We used three different preprocessing settings, suitable for three data representations used by document matching algorithms (see Section 4.), bag-of-words representation, paragraph embeddings, and contextual embeddings.

1. Bag-of-words text representation benefits from preprocessing which removes noise and performs normalization. Following tokenization based on regular expression that preserves alphanumeric characters, we filtered out tokens of length 1 and all numbers, performed lemmatization with the updated Lemmagen lemmatizer[3] (Juršič et al., 2010), and filtered stopwords using a list of 325 Croatian stopwords.

2. The paragraphs which serve as an input to the Doc2Vec model are tokenized with the regular expression that preserves only alphanumeric characters and the obtained tokens are lemmatized.

[3] https://github.com/vpodpecan/lemmagen3

3. While the input to contextual embedding models (mBERT and XLM-R) may be lightly preprocessed (e.g., removing the URLs), in our case we performed no preprocesing and used tokenizers provided with the implementation of these models.

4. Document comparison methods

In our approach, the search for the most similar documents compares vectors of documents using cosine similarity. We compare two classical pre-neural approaches using bag-of-words representation, tf-idf and latent semantic indexing (LSI), with approaches using dense vector embeddings computed with neural networks. In the later group, we use the Doc2Vec (Le and Mikolov, 2014a) model, as well as the contextual multilingual BERT (mBERT) and XLM-RoBERTa (XLM-R) models. Below we shortly describe each of the approaches.

4.1. Tf-idf

A standard pre-neural approach to document retrieval is to transform the documents into a bag-of-words (BOW) representation which is essentially a collection of sparse integer vectors (word counts). As BOW vectors do not take into account the relative importance of words in a document with respect to the whole corpus, they are typically re-weighted. A successful word reweighting method, term-frequency inverse-document-frequency (*tf-idf*), aims to reflect word importance in a document (*tf)* relative to its importance in the corpus (*idf*).

The preprocessing of a dataset (see Section 3.1.) returns a list of tokens for every document. These lists are transformed into sparse numeric vectors by first extracting the corpus vocabulary, computing word frequencies for each document (*tf*), and computing *tf-idf* weighted vectors using the vocabulary and overall word counts.

When compiling the corpus vocabulary, additional filtering parameters can be set. We used the default settings where tokens which appear in less than 5 documents or in more than 50% of all documents are filtered out. In addition, we experimented with setting these limits to 2 and 25%, respectively. The effects of different settings on the performance are presented in Section 6.

In our implementation, we used the `TfidfModel` from the Gensim library (Řehůřek and Sojka, 2010) and the default *nfc* SMART setting[4] for *tf-idf* weighting and normalization which used raw frequency for term frequency weighting, inverse document frequency for document frequency weighting and cosine document length normalization. This translates into the following set of equations for term frequency, document frequency, and normalization function:

$$tf(f_{i_k}) = f_{i_k}$$

$$idf(N, n_k) = log_2 \frac{N}{n_k}$$

$$g(G, D_i) = \sqrt{\sum_{k=1}^{t} w_{i_k}^2}$$

where $D_i = \{w_{i_1}, w_{i_2}, ..., w_{i_t}\}$ represents a document vector with t unique terms in D_i and w_{i_k} and f_{i_k} are weight and frequency of each term T_k in D_i, N is the number of documents, n_k is the number of documents containing term T_k, and $g(G, D_i)$ describes a cosine document length normalization factor expressed as a function of some global collection statistics G, and document D_i.

4.2. LSI

Latent semantic indexing (Deerwester et al., 1990) performs singular value decomposition (SVD) on the weighted term-document matrix which is typically composed of BOW or *tf-idf* vectors. The computed SVD is truncated which has the effect of retaining only the most important semantic information while the noise and other artifacts are reduced. The result of LSI is a dense matrix where each document is represented with a fixed dimensional numeric vector (with a few hundred dimensions). In this respect, LSI is similar to neural embedding methods although the elements of the resulting vectors have a very different meaning.

We used the LSI implementation available in Gensim to transform the vectors from the *tf-idf* representation. We tested number of dimensions set to 100, 300, and 500.

4.3. Doc2Vec

Le and Mikolov (2014b) introduced two variants of Doc2Vec approach for dense document representations, called PV-DM (Paragraph Vector-Distributed Memory) and PV-CBOW (Paragraph Vector-Continuous Bag-of-Words). In the PV-DM model, a training context is defined with a sliding window moving over the text. The training task is defined as a prediction of the central word of the context window with a shallow neural network. An input to this neural network are embeddings for all the other words from the context together with the embedding of the whole document. During training, the network learns both the word embeddings and the embedding for the whole document. The simpler PV-CBOW model does not have the access to the context words from the document. The neural network predicts randomly sampled words from the document given only the embedding of the whole document.

Both Doc2Vec models have a number of tuneable hyperparameters that can significantly impact the performance of the model. In order to determine those parameters, we used the Bayesian optimisation. This optimisation strategy optimizes black-box functions by placing a prior belief about the function, and updating it with each evaluation. The parameters of the function for the next evaluation are selected based on a predefined criterion that takes into account previous evaluations of the function. We optimized the evaluation metric MAP@10 (introduced in Section 5.) on related articles contained in the training set. To guide the search for parameters, we used a Gaussian Process (GP) prior and Expected Improvement (EI) criterion

[4]The SMART (System for the Mechanical Analysis and Retrieval of Text) Information Retrieval System defines notation for term weighting and normalization where different formulas are allowed for computing term frequency, document frequencty and document length normalization.

that maximize the evaluation metric. After 120 evaluations, we selected the best performing hyperparameters. The size of the resulting vector was set to 180 dimensions, the context window covered 5 words left and right from the central word, the vocabulary size was set to 36,000 words, and we ignored words with with less than 35 occurences. The training procedure used negative sampling with 30 negative words and a threshold for downsampling of frequent words set to 3.7×10^{-4}. In Section 6., we report only results using those hyperparameters.

We used the Doc2Vec (PV-DM and PV-CBOW) implementation available in Gensim library and the Bayesian optimisation from the Scikit-Optimize[5] library.

4.4. mBERT

Multilingual BERT (mBERT) is a 12 layer Transformer model (Vaswani et al., 2017) proposed by Devlin et al. (2019). The mBERT was simultaneously trained on Wikipedia pages of 104 languages using masked language modelling (MLM) and next sentence prediction (NSP) objectives. All 104 languages for the mBERT model use shared word piece vocabulary without an explicit way to denote different languages. Maximum length of the input sequence for the model is 512 tokens and each token is represented with 768 dimensions. News articles that require more tokens than the maximum length are trimmed and only the first 512 tokens are kept. The input to BERT begins with *[CLS]* token and ends with a *[SEP]* token denoting the end of a sequence. Additionally, model receives an attention mask to avoid performing attention on padding token indices in case of shorter articles. Running the model produces a context dependant representations of the input tokens. The whole input sequence can be represented with a single vector by using the context dependant representation of the *[CLS]* token. In Devlin et al. (2019), this representation is used as an aggregate sequence representation for classification tasks. Another way to represent the whole sequence, used in Reimers and Gurevych (2019), is to take an average of all input tokens. The BERT model allows for an input to contain a pair of sequences and it is possible to finetune the model with a classification objective that will classify the news article as related or not. At the inference time, in order to find related articles for a newly written article, this approach would require running the model for each article from the archive paired with the new article. Due to a very large number of articles, this is not feasible - determining which articles are related would take too much time and prolong the publication of the new article. Instead, we finetune the model using the Siamese BERT-Networks, a network architecture proposed by Reimers and Gurevych (2019) that avoids this problem. We hold out 10% of the training data and use it to evaluate the finetuning progress. Once the loss on the evaluation data starts increasing, the training is stopped. We use Adam optimizer with a learning rate of 1×10^{-5} with linear warmup on 10% of the training data and a batch of 16 examples. The objective that is minimized is the triplet margin loss:

$$L(a, p, n) = \max\{\|\mathbf{a} - \mathbf{p}\|_2 - \|\mathbf{a} - \mathbf{n}\|_2 + \mathit{margin}, 0\}$$

In our experiments, the margin for the loss function was set to 1. Anchor and positive example of the triplet objective is the pair of related articles, while a negative example is a related article of a randomly chosen article. We evaluate the model that is finetuned using the averaged token representation, as well as the one finetuned on the initial *[CLS]* token.

The pretrained model used for finetuning was the *bert-base-multilingual-cased* variant available in the Transformers (Wolf et al., 2019) library.

4.5. XLM-RoBERTa

The XLM-RoBERTa (XLM-R) (Conneau et al., 2019) is a large multilingual BERT-like model based on RoBERTa (Liu et al., 2019). It uses the sentence piece tokenizer and it is trained with the masked language model objective (MLM) on the CommonCrawl data in 100 languages, including Croatian. Similar to the mBERT, all languages share the same vocabulary (a larger one than mBERT) and the model does not need an explicit marker to denote the language of the input. The maximum size of the input is 512 tokens and each token is represented by a vector with 1024 dimension. Long news articles that do not fit within 512 tokens are trimmed and only the first 512 tokens are kept. All tokenized sequences begin with the '$<s>$' token that denotes a beginning of the sequence and end with the '$<\backslash s>$' token. The initial token can be used as a whole sequence representation in the similar way to the *[CLS]* token of the mBERT model. This model also accepts a pair of sequences as input, and the same caveats apply as with the mBERT model. Finetuning uses the same Siamese BERT-Networks architecture with the Adam optimizer, 4×10^{-5} learning rate with a linear warmup on 10% of the data and batches of 16 examples. A triplet margin objective is optimized on the related articles with the negative sampling as described for the mBERT model. We evaluate the model that is finetuned on the averaged token representation, as well as the one finetuned on the initial token.

The pretrained model used for finetuning is the *xlm-roberta-large* variant of the model available in the Transformers library.

5. Evaluation setting

As described in Section 3., the evaluation data consists of 25% of the latest articles from the whole dataset. This mimics the realistic scenario, where all older articles are considered when a new article is published. When considering eligible articles in our document retrieval task, we considered their age stored in the *published_date* attribute. This is consistent with the real world scenario where a journalist must not link older but unpublished articles in order to avoid dead links.

The algorithms were trained on the older 75% of articles. The *tf-idf* model thus discarded any newly introduced tokens (words) and used *idf* estimates from the training data when computing *tf-idf* vectors. The same *tf-idf* model was used for the LSI model. All embeddings-based models used trained models to infer vectors of the training data. We used cosine similarity for all document retrieval operations[6].

[5] `https://scikit-optimize.github.io`

[6] Note that the cosine similarity might not be the best choice

The performance of all algorithms was assessed using the mean average precision (MAP) score. The evaluation score for a single article query returns all other articles sorted by decreasing similarity from the query. The score for this query is the average precision of elements of the ranked result list. Evaluation over all articles produces a list of such average precisions. The final score for the model is a mean across the average precisions. In our experiments, we considered only the top ten retrieved documents when computing the MAP score, i.e. MAP@10. A model that would return articles at random (i.e. the random baseline model) would have a 0.0 score in MAP@10.

6. Results

The performance of compared document linking methods is presented in Table 1. The results are surprising, as the best result is achieved with the simple *tf-idf* model. This is disappointing concerning the state-of-the-art neural embeddings. However, we evaluated different representations using the links between articles selected by the journalists. This does not necessarily mean that the retrieved articles are not good recommendations, but verifying this hypothesis would require human evaluation.

The *tf-idf* representation consistently performs best across all evaluated hyperparameters of the model, which do not significantly influence the score. Nevertheless, results in Table 1 suggest that including rare words ($m = 2$) improves the performance, while excluding frequent and rare words ($M = 25\%, m > 2$) decreases the performance. While the difference is small it suggests that *tf-idf* models trained on this domain may benefit from preprocessing settings that do not remove what is typically considered as artifacts or noise. From a journalist's perspective this corresponds to linking articles based on a few rare keywords. This may also offer an explanation why LSI does not achieve scores comparable to *tf-idf*. Since LSI is designed to retain only the most important semantic information, the rare words which could improve the models in this particular domain are filtered out.

Both Doc2Vec representations perform similarly are much better then LSI. They approach the performance of *tf-idf* but cannot match it. LSI is less successful with the reduction of dimensions but is competitive with much larger mBERT and XLM-R embeddings that use 768 and 1024 dimensions to represent a document. For both mBERT and XLM-R, using an average of contextual token embeddings shows better results than using the result of only the *[CLS]* token, which is consistent with the conclusions of Reimers and Gurevych (2019).

Model	MAP@10
tf-idf (m=5, M=50%)[7]	0.279
tf-idf (m=2, M=50%)	**0.281**
tf-idf (m=2, M=25%)	**0.281**
tf-idf (m=10, M=50%)	0.277
tf-idf (m=10, M=25%)	0.277
LSI (d=500)	0.186
LSI (d=300)	0.166
LSI (d=100)	0.124
Doc2Vec (PV-DM)	0.248
Doc2Vec (PV-CBOW)	0.240
mBERT (AVG)	0.142
mBERT (CLS)	0.094
XLM-R (AVG)	0.246
XLM-R (CLS)	0.186

Table 1: The performance of different approaches on the task of similar news article retrieval.

7. Conclusions and future work

We evaluated several document representations to recommend related news articles. The results show that the *tf-idf* representation produces results more consistent with manual selection of journalists compared to more sophisticated representations. We do not yet have a definite explanation for this result and plan to explore it in the future work. Our current belief is that journalists use a keyword search to locate related articles. The articles found in this way contain the same words as the query and bias the evaluation in favor of the *tf-idf* method.

It is possible that the dataset contains a significant amount of noise and that better results cannot be achieved. For example, one possible source of noise would be journalists that misuse the related article information by including their own articles as related, even when they are not, in order to increase the view count of their own articles. In future work, we plan to perform a manual evaluation of the returned related articles on a selected subset.

Although mBERT and XLM-R models achieve state-of-the-art results in many NLP tasks, they did not fare well in this evaluation. Another reason for this might be that document representations created by those models are not suitable for comparison with the cosine similarity. Reimers and Gurevych (2019) offer this explanation when evaluating BERT representations and we plan to explore the impact of similarity measures on the performance of mBERT and XLM-R models.

Acknowledgements

The research was supported by the Slovene Research Agency through research core funding no. P6-0411 and P2-103. This paper is supported by European Union's Horizon 2020 Programme project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media, grant no. 825153). The results of this publication reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

8. References

Agra Bimantara, Michelle Blau, Kevin Engelhardt, Johannes Gerwert, Tobias Gottschalk, Philipp Lukosz, Shenna Piri, Nima Saken Shaft, and Klaus Berberich. 2018. htw saar @ TREC 2018 news track. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018*.

for some embeddings models.

[7]This is the default setting for filtering extremes from the dictionary in Gensim.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, F. Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *ArXiv*, abs/1911.02116.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Marwa Essam and Tamer Elsayed. 2019. bigIR at TREC 2019: Graph-based analysis for news background linking. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*.

John Foley, Ananda Montoly, and Mayeline Pena. 2019. Smith at TREC2019: learning to rank background articles with poetry categories and keyphrase extraction. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*.

Michael Gamon, Sumit Basu, Dmitriy Belenko, Danyel Fisher, Matthew Hurst, and Arnd Christian König. 2008. Blews: Using blogs to provide context for news articles. In *ICWSM*.

Weiwei Guo, Hao Li, Heng Ji, and Mona T. Diab. 2013. Linking tweets to news: A framework to enrich short text data in social media. In *ACL*.

Shudong Huang, Ian Soboroff, and Donna Harman. 2018. TREC 2018 news track. In *Proceedings of the Second International Workshop on Recent Trends in News Information Retrieval co-located with 40th European Conference on Information Retrieval (ECIR 2018)*, pages 57–59.

Daisuke Ikeda, Toshiaki Fujiki, and Manabu Okumura. 2006. Automatically linking news articles to blog entries. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

Matjaz Juršič, Igor Mozetič, Tomaž Erjavec, and Nada Lavrač. 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *J. UCS*, 16(9):1190–1214.

Quoc Le and Tomas Mikolov. 2014a. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196.

Quoc Le and Tomas Mikolov. 2014b. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kuang Lu and Hui Fang. 2018. Paragraph as lead - finding background documents for news articles. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018*.

Kuang Lu and Hui Fang. 2019. Leveraging entities in background document retrieval for news articles. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP/IJCNLP*.

Pinky Sitikhu, Kritish Pahi, Pujan Thapa, and Subarna Shakya. 2019. A comparison of semantic similarity methods for maximum human interpretability. *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, 1:1–4.

Ian Soboroff, Shudong Huang, and Donna Harman. 2018. TREC 2018 news track overview. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018*.

Anastasia Soloshenko, Yulia Orlova, Vladimir Rozaliev, and A.V. Zaboleeva-Zotova. 2015. Establishing semantic similarity of the cluster documents and extracting key entities in the problem of the semantic analysis of news texts. *Modern Applied Science*, 9, 04.

Dwane van der Sluis, Dyaa Albakour, and Miguel Martinez. 2018. Signal at TREC 2018 news track. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.