SentArg: A Hybrid Doc2Vec/DPH Model with Sentiment Analysis Refinement

Notebook for Touché: Argument Retrieval at CLEF 2020

Christian Staudte[1] and Lucas Lange[2]

[1] Leipzig University, Germany
cs47rica@studserv.uni-leipzig.de
[2] Leipzig University, Germany
ll95wyqa@studserv.uni-leipzig.de

Abstract In this work we explore the yet untested inclusion of sentiment analysis in the argument ranking process. By utilizing a word embedding model we create document embeddings for all queries and arguments. These are compared with each other to calculate top-N argument context scores for each query. We also calculate top-N DPH scores with the Terrier Framework. This way, each query receives two lists of top-N arguments. Afterwards we form an intersection of both argument lists and sort the result by the DPH scores. To further increase the ranking quality, we sort the final arguments of each query by sentiment values. Our findings ultimately imply that rewarding neutral sentiments can decrease the quality of the retrieval outcome.

1 Introduction

In this notebook we propose our *SentArg* model for the first task of *Touché 2020: 1st Shared Task on Argument Retrieval*[3] [6]. The task is performed on the *args.me corpus* [1]. The source code is available on our public GitHub repository[4]. SentArg is based on the *Dual Embedding Space Model* by Mitra et al. [11], although we replace BM25 with DPH [2] and adjust their scoring function. For further refinement we use sentiment analysis to classify the emotional level (positive, neutral, negative) of every argument utilizing the *Google Cloud Natural Language API*[5]. Analysing the author's attitude, we thus re-rank arguments. Our main objective is to find out whether sentiments can increase the quality of an argument ranking.

The remainder of our notebook is organized as follows. Section 2 outlines all related works we used as theoretical and practical foundation. In Section 3 we describe our SentArg model configuration. Our evaluation as well as experiments are then presented in Section 4. Section 5 concludes our notebook.

[3] https://touche.webis.de

[4] https://github.com/luckyos-code/ArgU

[5] https://cloud.google.com/natural-language/

2 Related Work

Words used in a mathematical model must be mapped onto mathematical objects. *One-hot* encodings are particularly well-known mappings for this: Each word is represented by an element = 1 in vector $\overrightarrow{v} \in \{0,1\}^{|Vocab|}$, whereas all other elements are 0. However, these representations scale with vocabulary size and are therefore extremely sparse. Further, semantic relations between words are not modeled [7]. To overcome these limitations, Mikolov et al. [10] introduce two neural network models: *Continuous Bag-of-Words* (CBOW) and *Skip-Gram* (SG)[6]. Both architectures typically apply one input-, one hidden-, and one output-layer. For any given word w, its left- and right-sided context serves as input for a CBOW model, which tries to predict w. After training, weight matrix $W_{IN} \in \mathbf{R}^{M \times N}$, with N being the embedding's vector size, contains all available word embeddings. Words that are semantically similar also occur in similar contexts, hence cosine similarities of their embeddings are close to 1.

Regarding argument detection, similar contexts between queries and arguments may not be enough. To further restrict argument result sets, Mitra et al. [11] suggest a Dual Embedding Space Model. They calculate BM25 scores and word embedding similarities for any query Q regarding all arguments. Afterwards they form a weighted sum of both scores, which acts as argument relevance score. One special feature is the usage of CBOW's W_{OUT} matrix. By calculating the cosine similarity of two word embeddings, each from a different weight matrix, co-occuring words reach scores close to 1. In their final results, IN-OUT-similarities provide the best results.

Regarding typical information retrieval models, Potthast et al. [13] compare BM25, TFIDF, DirichletLM and DPH on more than 300.000 arguments (documents), containing 40 topics. Assessors evaluated different query results and analyzed various aspects of found arguments, such as relevance, as well as rhetorical, logical and dialectical quality. Their evaluation shows that DitrichletLM and DPH are superior to BM25 and TFIDF. With these results in mind, we decide to replace BM25 in the Dual Embedding Space Model with DPH.

The goal of sentiment analysis is to identify the emotional character of a given text. This consideration is especially interesting in the case of arguments because research shows that arguments are fundamentally emotional [8,15]. The idea to include emotions in our ranking is based on the hypothesis that an emotional attachment to a topic can indicate involvement and thus stronger argumentation, which is backed by several studies [4,8,15,16]. We therefore expect a ranking in favor of emotional arguments to perform better than one favoring neutrality or one without sentiment analysis.

A survey conducted by Lawrence and Reed [9] shows that works on sentiment analysis in argument retrieval mainly focus on opinion mining, i.e. guessing the stance of an argument. Therefore they do not use an already annotated corpus, rather they instead aim to create one. Transitioning sentiment analysis from retrieving to ranking arguments is a new approach provided through our model.

[6] We primarily focus on CBOW in this notebook, so please refer to the referenced paper [10] for more information on SG.

3 Argument Retrieval Model

Our model features three steps: 1. Pre-processing; 2. Training of word embeddings and DPH to calculate argument and query similarities; 3. Sentiment analysis to re-sort arguments.

Concerning Google's sentiment analysis API, we first remove noise and formatting errors in as many arguments as possible. We manually define the following rule set:

- Remove URLs and square brackets with their content
- Remove some punctuation $\{\sim$, #, §, &, @, =, *$\}$
- Reformat punctuation with correct spacing
- Replace identical letters in a row (>2) with one representative: *hellooooooo* → *hello*

Short arguments often express approval or disapproval with previous arguments, hence we delete arguments containing less than 26 words. For our dual embedding's CBOW model we replace/add the following rules:

- Numbers and URLs are replaced by tags <NUM> and <URL>
- Remove all punctuation
- Remove stop words

We keep upper- and lower-cases to reduce word ambiguities[7] and train the CBOW model on all given arguments[8]. However, this gives rise to problems concerning rare query terms: "E-Cigarettes" occures so rarely that no word embedding is trained. In this case we try different combinations of upper- and lower-case to find appropriate (and possibly multiple) word embeddings, which are all taken into account.

After preprocessing and training a CBOW model, we compute document vectors for each argument as proposed by [11]:

$$\overline{A} = \frac{1}{|A|} \cdot \sum_{a_j \in A} \frac{a_j}{||a_j||} \tag{1}$$

Each a_j represents the word embedding for word j. Afterwards we calculate similarities between a query and all arguments:

$$DESM(Q, A) = \frac{1}{|Q|} \cdot \sum_{q_i \in Q} \frac{q_i^T \overline{A}}{||q_i^T|| \cdot ||\overline{A}||} \tag{2}$$

DPH is the second relevant component we model with the help of the Terrier framework [12]. Contrary to the original dual embedding architecture, we do not calculate a weighted sum of scores. Mitra et al. [11] state that word embeddings are helpful for finding appropriate contexts, whereas BM25 is better at finding concrete details given any query. We therefore run DESM and DPH in parallel, both select the top 1.000 arguments.[9] Then, we form an intersection of both sets and sort every argument by its DPH score. Only context relevant scores thus have influence on the result set.

[7] For example "US" $\neq$ "us"

[8] Configuration: vector size = 300; window size = 3; min word count = 5

[9] A manually selected and adjusted parameter.

Sentiment analysis of the Google Cloud Natural Language API was created for a wide range of applications and is therefore most suitable for discussions in the data set. For every argument we send a request to the API, which in return provides us with a sentiment value. This value represents the direction of emotion on a scale from -1 to 1, with -1 and 1 being the strongest. Values between -0.1 and 0.1 represent neutral (N) sentiments, while values lower than -0.1 and greater than 0.1 express negative (-) and positive (+) sentiments, respectively.

Obtained sentiments are combined with DPH scores to compute the final ranking. To confirm our expectations we introduce two variants: (i) We encourage emotional arguments (values closer to -1 and 1); (ii) We encourage neutral arguments (values closer to 0):

$$DPH^{*}_{Q,A} = DPH_{Q,A} \pm DPH_{Q,A} \cdot \frac{|S_A|}{2} \tag{3}$$

S_A refers to the sentiment value for argument A. Adding the weighted DPH refers to variant (i) and subtracting it to variant (ii).

4 Experiments and Evaluation

Since opinion mining states that sentiment analysis is capable of determining the stances of arguments [3,9], we tested if information is gained from sentiment analysis by comparing each argument's stance with its sentiment value. In the dataset, each stance is stored as Boolean (pro and con), neutral stances are therefore not given. We enriched these stances by defining a sentiment value range that includes neutral (N), positive (+) and negative (-) arguments. Table 1 displays the distribution of sentiment values compared to annotated stances. As can be seen, most arguments (57%) are neutral, while 11% are positive and 32% are negative. However, positive and negative arguments do not match the dataset's stance distribution of 53% pro and 47% con[10]. Further, every sentiment class contains at least 42% candidates that are either pro or con. In spite of the opinion mining hypothesis, arguments for every sentiment class are nearly evenly distributed and no real correlation can be seen. In conclusion, by utilizing sentiment values we can gain new perspectives on the arguments.

Part of the task was the submission and evaluation of our model on the TIRA platform [14].[11] In the following, we refer to a new set of qrels (query relevances). To identify the influence of our sentiment strategy and model architecture on the retrieval, we evaluate six different runs representing the different variants (see Section 3): R_0 No sentiment analysis, R_E Emotional is better [cp. (i)], R_N Neutral is better [cp. (ii)] and the two embedding types (IN-IN, IN-OUT). The results are shown in Table 2. R_N delivers the lowest scores for both embedding types and all nDCG-measures. That means subtracting the absolute sentiment value from the DPH score penalizes arguments, which are relevant. The assumption would be that adding sentiment values can reward more relevant arguments. As can be seen in Table 2, this is partly true: For IN-OUT, R_E

[10] These numbers are not mentioned in Table 1, even though the distribution for neutral sentiments looks this way.

[11] `https://www.tira.io` ; Group: *ir-lab-ul-t1-detroitnitz*

reaches slightly better results than R_0. But R_0 moderately outmatches most scores of R_E when it comes to IN-IN. Emotional arguments seem to be preferred by the DPH model at default, thus scores from runs R_0 and R_E are close to each other. In sum however, the overall top nDCG@X scores are reached by the configuration [R_E, IN-OUT].

Table 1. Comparison of the sentiment (Sent.) values (N, +, -) of arguments and their stances (Pro, Con).

Sent.	Argument's share in number (%)	Stance in % per share	
		Pro	Con
N	169,152 (57%)	53%	47%
+	33,406 (11%)	58%	42%
-	94,460 (32%)	50%	50%

Table 2. Evaluation for different configurations: Embedding types and sentiments' influence.

	IN-OUT				IN-IN			
	nDCG	nDCG@5	nDCG@10	QrelCov@10	nDCG	nDCG@5	nDCG@10	QrelCov@10
R_0	0.365	0.649	0.553	6.24	**0.390**	0.635	0.538	6.06
R_E	0.369	**0.699**	**0.559**	6.24	0.385	0.625	0.528	5.96
R_N	0.337	0.517	0.456	5.18	0.359	0.500	0.437	4.94

5 Conclusion

By (1) combining IN-OUT argument embeddings and DPH to retrieve relevant arguments and (2) sorting arguments in regards to their sentiment value we could learn two things: Not only can IN-OUT embeddings improve the context space for a queries argument list, but also can rewarding neutral arguments in a final ranking reduce the quality of a retrieval. Our empirical findings match the expectations set forth in our analysis of existing literature, that more emotional arguments are more relevant than neutral ones. That means prioritizing arguments with high sentiment values can have a positive influence on the relevance ranking.

Future work may benefit from replacing the CBOW model with a FastText [5] architecture, which generates good results especially for noisy data. Further, static parameters and the influence equation for sentiments can be optimized. This way different sentiment values could have a more diverse influence on the final score.

References

1. Ajjour, Y., Wachsmuth, H., Kiesel, J., Potthast, M., Hagen, M., Stein, B.: Data Acquisition for Argument Search: The args.me corpus. In: Benzmüller, C., Stuckenschmidt, H. (eds.) 42nd German Conference on Artificial Intelligence (KI 2019). pp. 48–59. Springer (Sep 2019). https://doi.org/10.1007/978-3-030-30179-8_4
2. Amati, G.: Frequentist and bayesian approach to information retrieval. pp. 13–24 (01 1970). https://doi.org/10.1007/11735106_3
3. Bakshi, R.K., Kaur, N., Kaur, R., Kaur, G.: Opinion mining and sentiment analysis. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). pp. 452–455. IEEE (2016)
4. Blanchette, I., Caparos, S.: When emotions improve reasoning: The possible roles of relevance and utility. Thinking & Reasoning **19**(3-4), 399–413 (2013)
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
6. Bondarenko, A., Fröbe, M., Beloucif, M., Gienapp, L., Ajjour, Y., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M., Hagen, M.: Overview of Touché 2020: Argument Retrieval. In: Working Notes Papers of the CLEF 2020 Evaluation Labs (Sep 2020)
7. Braud, C., Denis, P.: Comparing word representations for implicit discourse relation classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2201–2211. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1262, `https://www.aclweb.org/anthology/D15-1262`
8. Gilbert, M.A.: Coalescent argumentation. Routledge (2013)
9. Lawrence, J., Reed, C.: Argument mining: A survey. Computational Linguistics **45**(4), 765–818 (2020)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
11. Mitra, B., Nalisnick, E., Craswell, N., Caruana, R.: A dual embedding space model for document ranking (2016)
12. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Losada, D.E., Fernández-Luna, J.M. (eds.) Advances in Information Retrieval. pp. 517–519. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
13. Potthast, M., Gienapp, L., Euchner, F., Heilenkötter, N., Weidmann, N., Wachsmuth, H., Stein, B., Hagen, M.: Argument search: Assessing argument relevance. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1117–1120. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331327, `https://doi.org/10.1145/3331184.3331327`
14. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. The Information Retrieval Series, Springer (Sep 2019). https://doi.org/10.1007/978-3-030-22948-1_5
15. Villata, S., Cabrio, E., Jraidi, I., Benlamine, S., Chaouachi, M., Frasson, C., Gandon, F.: Emotions and personality traits in argumentation: an empirical evaluation 1. Argument & Computation **8**(1), 61–87 (2017)
16. Walton, D.: The place of emotion in argument. Penn State Press (2010)