An Open-Domain Web Search Engine for Answering Comparative Questions

Notebook for the Touché Lab on Argument Retrieval at CLEF 2020

Tinsaye Abye, Tilmann Sager, and Anna Juliane Triebel

Leipzig University, Germany
{ta83jyga, ts99nami, wir13ljt}@studserv.uni-leipzig.de

Abstract. We present an open-domain web search engine that can help answer comparative questions like "Is X better than Y for Z?" by providing argumentative documents. Building such a system requires multiple steps that each includes non-trivial challenges. State-of-the-art search engines do not perform very well on these tasks, and approaches to solve it are part of current research. We present a system to process the following tasks: Detection of comparative relations in a comparative question, finding claims and arguments relevant to answering comparative questions and scoring the relevance, support and credibility of a website. We follow a rule-based syntactic NLP approach for the comparative relation extraction. To measure the relevance of a document, we combine results from the existing models BERT and CAM. Those results are reused to determine the support through an evidence-based approach, while the credibility consists of a multitude of scores. With this approach, we achieved the best NDCG@5 of all systems participating in task 2 of the Touché Lab on Argument Retrieval at CLEF 2020.

1 Introduction

When searching the web for the answer to a comparative question, popular search engines like Google or DuckDuckGo provide results by referring to question-and-answer[1] or debate[2] websites, where mostly subjective opinions are displayed [26]. Domain specific comparison systems rely on structured data which makes them inappropriate for answering open domain comparative questions since the data is not structured. Although modern search engines are advanced, answering comparative questions is still challenging [26] and therefore subject to current research in the field of information retrieval. We participate in CLEF 2020 for Task 2 [4], which sets the challenge to retrieve and re-rank documents of the

[1] `https://de.quora.com`
[2] `http://debate.com`

ClueWeb12[3] data set aiming to answer comparative questions that are not categorized in a specific domain with argumentative results [4]. The results will be assessed on the dimensions relevance, support and credibility. Our prototype is tested on TIRA [20].

2 Related Work

To create an open-domain web search engine for comparative question answering, we build upon results from numerous fields with a connection to information retrieval, like comparison mining, argument mining, comparative opinion mining and evidence mining.

Comparative Relation Extraction We receive user input as a natural language comparative question. Therefore the problem of detecting the entities and features, i.e. the comparative relation (CR) arises just like in comparative opinion mining. [28] Comparative relation extraction is used successfully by Xu et al. [30] by using a dependency graph to detect the CR. Comparative opinion mining from online reviews uses Part-Of-Speech (POS) tags as well as domain specific aspects [11]. Two techniques based on syntactic analysis were compared by Jindal et al. [15]. The use of label sequential rules that uses POS tags outperforms class sequential rules using keywords. The use of syntactic dependency trees was proven helpful by Gao et al. [11] and Xu et al. [30]. We conduct a syntactic analysis of the queries using POS tags and dependency trees. We follow a syntactic natural language processing (NLP) approach to provide a domain-agnostic, rule-based model for extracting the CR from the comparative user query.

Comparison and Argument Mining The CR then serve as input to comparison and argument mining models that rely on structured data. Hence we close the gap between user queries and structured input needed by comparison and argument mining models with comparative relation extraction. The Comparative Argumentative Machine (CAM) by Schildwächter et al. [26] is an open-domain information retrieval system capable of retrieving comparative argumentative sentences for two given entities and several features. Argument mining systems detect argumentative sentences including premises, claims or evidence sentences [17]. Fromm et al. [10] demonstrate that taking the context of an argument into account significantly boosts the performance of an argument detecting system, whereas most of traditional argumentative unit detecting systems [2, 7] are topic agnostic. The Bidirectional Encoder Representations from Transformers (BERT) model [9] proposed by Reimers et al. [24] finds arguments, and is also able to detect, if they support a certain topic. We make use of a combination of these models to find argumentative documents that are relevant to the user query and therefore help answer comparative questions.

[3] `http://lemurproject.org/clueweb12`

Support and Evidence Mining We further increase the quality of candidates presented by CAM and BERT with Support and Evidence Mining. As we aim to find documents that provide arguments for decision-making, the mining of context-dependent, evidence-based arguments is an important task. Braunstain et al. [5] rank support sentences in community-based question answering forums about movies. Evidence Mining provides many publications of different subtasks like extracting evidence sentences from documents [25],detecting claims and retrieving evidence [1, 13]. Since we are interested in a document's support for a query, we extract evidence sentences and analyze their relatedness to claims using methods presented by Rinott et al. [25]. A higher ranking of documents with a good support and evidence for the claims made, should further increase the usefulness of the search results in order to answer the comparative question asked by the user.

3 Comparison Retrieval Model

In this section, the comparison retrieval model we designed to build an open-domain web search engine for answering comparative questions, as sketched in Figure 1, is described in detail. The retrieval model consists of four phases to retrieve and rank web search results to answer comparative questions. In phase one (blue), the question is analyzed for its comparative relation and expanded queries are sent to the ChatNoir [3] search engine. The retrieved documents then go through NLP processing. During the second phase (red) comparison and argument mining are conducted on the pre-processed documents. Through evidence mining, link analysis and diverse other sources, scores that quantify the quality of the documents are collected. In the third phase (yellow), the collected scores are summed up to build the meta-scores of relevance, support and credibility. The final phase four (green) delivers weighted scores and re-ranked documents.

This section is structured accordingly to the phases depicted in Figure 1: subsection 3.1 describes the pre-processing, subsection 3.2 the analysis of the documents and reranking is covered in subsection 3.3.

3.1 Pre-processing

In the pre-processing phase, the user query in analyzed, expanded and several queries are sent to the ChatNoir [3] search engine. A linguistic analysis is performed on the content of the websites returned by ChatNoir.

Comparative Relation Extraction A comparative relation consists of entities to compare and the features, the entities are compared by. Albeit a CR is easy to detect for a human, it is not trivial to extract it computationally [15]. Due to the given task, we know that the user query is a comparative question. But we must detect the comparative relation within that question. Using a syntactic NLP approach, we use spacy (model en_core_web_sm, trained on the OntoNotes

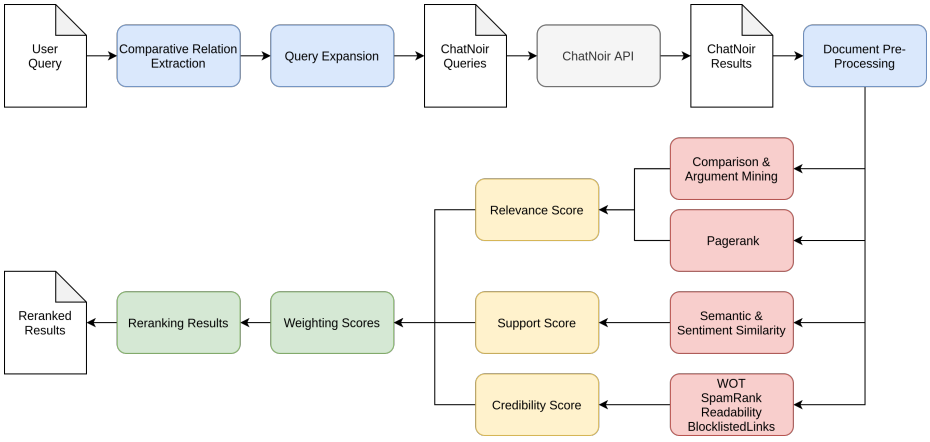

Fig. 1. Comparison Retrieval Model Overview

Web Corpus [29]) to extract the CR from the user query. It provides us with tokenization, chunking, POS tagging and dependency parsing. Two main types of comparative relations occur in the user query: comparative and superlative questions. As the syntactic structure of a comparison in a question does not differ from the CR in a statement, we use the term "superlative" accordingly to Jindal et al. The term "comparative" matches their "non-equal-grabbable" [15]. As the direction of the CR and the distinction between "feature" and "relation word" is not relevant in this case, our term "feature" covers them both.

The CR of superlative questions are detected by the following characteristics that result in high accuracy for the given topics: Superlative questions contain no grammatical or-conjunction but a superlative adjective ("highest") which is the feature. The child of superlative is the entity ("mountain") and the child of a prepositional modifier is another feature ("earth"). For queries with a syntactic pattern like: "What is the highest mountain on earth?", the presented method works perfectly.

To determine the entities in a comparative question we source the syntactic information from the question's dependency graph. This strategy allows for more than two entities to be detected in one sentence. One entity is the parent of a conjunction and the other entities. First we look for this pattern in chunks, i.e. nominal phrases, of the question. Finding a chunk provides the advantage that it contains descriptive adjectives or compounds. If no chunks could be found, the same rule is applied to the tokens of the question. For queries without a conjunction there is no simple rule to detect them. A feasible strategy was to assume that if there are up to two nouns in the query, that are no attributes, they are the entities to compare. Entities can also be verbs if there are no non-stop-word nouns in the question.

Features turned out to be more diverse than entities, but most of them are comparative adjectives, superlative adjectives, verbs in base form or children of

adjectival complements. If there are direct objects or nominal subjects in the question that were not detected as entities, they are assumed to be features. Finally adjectives, compounds and numeric modifiers are added to all entity and feature tokens, e.g. to be able to compare "green tea" to "black tea".

The two main reasons for failing the CR-detection are errors of POS tagger [15] and features detected as entities and vice versa. Since the system is customized for the topics of the task, it will not scale for comparative questions with different syntactic structure, especially more complex ones. In general the achieved results for detecting comparative relations from the user queries can be seen as satisfying.

Query Expansion To expand the queries that will be send to ChatNoir, [3, 21] we collect synonyms and antonyms of the comparative relation's features. Antonyms are fetched from the WordNet lexical database [4]. Synonyms are retrieved through a Gensim continuous skip-gram model [23] that was trained on a dump of the English Wikipedia from 2017 [16]. In some cases the Gensim model also returned antonyms, but as we do not care for the direction of the CR, that is not a problem. From the comparative relation and the expanded features we send four queries to ChatNoir using the index generated from the ClueWeb12 data set. First the original comparative question raised by the user, second the entities combined with 'AND', third the entities and the features and fourth the entities, features and their synonyms and antonyms combined with 'OR'. Multiple expanded queries increase the number of results and therefore the recall reachable through further re-ranking.

From ChatNoir we receive a list of results consisting of snippets, page titles, URIs, page ranks and spam ranks. We fetch the API again to get the full HTML-document for every result. From the HTML-documents the external links and the body content are extracted. We remove any CSS and JavaScript code from the body, as well as header-, footer- and nav-tags by using the Python package BeautifulSoup[5]. The body is then segmented into sentences. Then tokenization, POS tagging, dependency parsing and named entity recognition is performed on the sentences [12]. Sentences with a minimum length of four tokens are selected for further analysis and reranking of the results.

3.2 Analysis

Since our interest is finding documents that help answering comparative questions, we aim to detect comparative, argumentative and supportive sentences in the retrieved documents. We analyze them for sentences which compare the entities extracted from the user query, for sentences which contain arguments regarding the decision to be made and for sentences which support such claims. The documents should discuss both entities, may be favoring one of them and ideally justify the decision.

[4] `https://wordnet.princeton.edu/`
[5] `https://www.crummy.com/software/BeautifulSoup/`

Comparative Sentences In order to find sentences that compare the entities from the user query, we choose two of the best performing classification models according to Panchenko et al. [19]: BOW and InferSent [8]. Both models are based on the gradient boosting library XGBoost [6]. BOW uses bag-of-words word embeddings and InferSent uses the sentence embeddings method for feature representation [8]. To assess the models, we crafted a small evaluation data set with 100 sentences, 60 of them being comparative taken from the ClueWeb12 corpus covering 11 different topics. Both classifiers are able to distinguish between three cases: the sentence is comparative in favor of the first entity, in favor of the second entity or contains no comparison. We collect all sentences detected as comparison and discard the non-comparative ones. Both BOW and InferSent have a high precision, while BOW performed slightly better. Although both models reach the same recall at .48, we observed that the true positives they return are partially distinct. The strategy of running both models in combination leads to a significantly higher recall of .66. To achieve that improvement, we first run BOW. On the sentences that were not recognized as comparative in the first step, we run the detection with InferSent.

Argumentative Sentences We exploit the importance of topic awareness for detecting argumentative sentences by using the fine-tuned BERT [9] model proposed by Reimers et al. [24]. For a sentence and a topic, which in our case is one of the entities, the BERT classifier can detect if the sentence is an argument for, argument against or no argument regarding the topic. This enables us to collect arguments that aid the decision-making, because the arguments detected are relevant to the question to be answered. Despite the good performance compared to other models, with BERT we detected systematic errors as well. Comparative sentences were not classified properly. These are according to BERT for or against both entities at the same time, leading us to exclude comparative sentences.

Support Sentences Next to the number of arguments, a well-balanced argumentation structure is also crucial for satisfying the user's information need. Neither a document with a high number of claims, that are not supported by any argument, nor a document with a high number of arguments, that are not connected to any relevant claims, helps to find well-founded statements. Therefore we want to extract the arguments included in the document that directly support one or several claims. Defining support sentences turned out to be challenging, see section 2. Therefore we used the definition of an *Context-Dependent Evidence (CDE)* by Rinott et al. [25]. Their definition of a CDE sentence is very similar to the definition of a support sentence of Braunstain et al. [5]:"[A Context Dependent Evidence is] a text segment that directly supports a claim in the context of the topic." Nevertheless, we continue using the term support sentence. Rinott et al. also provide important characteristics of a support sentence: semantic relatedness, relative location between claim and support sentence and sentiment-agreement between them. Following the steps of Rinott et al. as a

guideline, we implemented a support sentence classifier. Since support sentences are arguments as well, we take the BERT result (see section 3.2) as input for the candidate extraction. Therefore we rank the BERT-classified arguments by their context independent features, e.g. named entity labels like PER or ORG, certain terms like nevertheless, therefore or but, and filter the first 70%, except there might be less than 10 sentences after thresholding. We used a lower threshold because BERT often returns only a few sentences. Taking the CAM-classified sentences as claims regarding to our task to provide arguments for comparisons, we determine semantic and sentiment relatedness between every claim and every candidate in the context-dependent stage. The semantic relatedness is measured by BERT, the sentiment similarity by TextBlob.[6]

3.3 Reranking

In order to compare and finally re-rank the retrieved documents, we define several measures for each document, that are assigned to the scores relevance, support and credibility.

Relevance As defined by Manning et al. [18]: "A document is relevant if it is one that the user perceives as containing information of value with respect to their personal information need." Therefore our measures for the relevance are mainly comprised from the comparative sentences and argumentative sentences described in subsection 3.2. We establish a ComparisonCount that results from counting the comparative sentences detected according to section 3.2. CAM is able to classify which entity is described as better or worse than its competitor in the sentence. First we considered introducing a score that provides a well-balanced measure between the two compared entities. But there is not always an equal amount of arguments for both entities. If one of the entities is not as good as the other, one can not assume to find many arguments for it. Comparative sentences in general take both entities into account giving sufficient reflection on both of them. But only counting the argumentative sentences returned by BERT (ArgumentCount) could favor documents only dealing with one of the entities. To prevent such unbalanced results, we put together a formula that considers both the amount of arguments but also the distribution between the entities:

$$ArgumentRatio = total\ arguments - \frac{|arguments\_1 - arguments\_2|}{total\ arguments + 1} \quad (1)$$

In Equation 1 `arguments_n` means arguments related to entity `n`, while `total arguments` represents the total amount of arguments in the document. The fraction takes their distribution over all entities into account. Further, we divided the term with a threshold, took the hyperbolic tangent to flatten the function and generalizing it for more than two entities. But the repetition of the same,

[6] `https://textblob.readthedocs.io/en/dev`

possibly rephrased, argument can spoil the measure. To overcome this issue we measured the similarity between the sentences using BERT for detecting argument similarity. This method was also presented by Reimers et al. [24].

Support A support sentence is defined as "a text segment that directly supports a claim in the context of the topic" [25, 1]. To convert the output of the support analysis into a measure, we defined a good document with respect to the given task: a good document has a high number of support sentences, that directly support claims included in the document. The connection between claim and support sentence is described by their semantic and sentiment similarity. To score the argumentation structure of a document, two measures were defined: SemanticRatio and SentimentRatio. SemanticRatio describes the number of support sentences per claim that are semantically similar. Since Braunstain et al. [5, 138] and Rinott et al. [25, 6] point out that especially the sentiment similarity between a claim and a support sentence is an indicator for a coherence, SentimentRatio was added as well. To counterbalance SemanticRatio and SentimentRatio, SupportCount as the number of distinct support sentences was added.

Credibility Jensen et al. define credibility as the "believability of a source due to message recipients' perceptions of the source's trustworthiness and expertise" [14, 1]. Since Rafalak et al. [22] claim credibility as very subjective, we added multiple different measures to balance the score. Web Of Trust (WOT)[7] provides user ratings for websites. This measure describes the Bayesian averaged opinion of at least 10 users for a website's host. Additionally, the SpamRank, the likelihood of spam, was added, which is delivered by ChatNoir. We assume that the richer the language used by the author of the document, the more credible is the information. With other words, the more complex a text is written, the more effort was put into writing this text by the author. Therefore we calculate three independent readability scores: Automated Readability Index (ARI), Dale-Chall Readability (DaleChall) and Flesch Reading Ease (Flesch). ARI [27] describes the understandability of a text. Since ARI, DaleChall and Flesch inspect different aspects of a document, e.g. the usage of different words or the number of syllables per word, all the measures were included to cover a wide range of the understandability and readability of a document. However, the actual scores calculated for the received documents were out of the ranges proposed by the respective authors. This is partly due to the difficulty of extracting clean texts out of HTML documents. To prevent the top results from containing a lot of advertisements or links that lead to block-listed hosts, the external links of a document are checked against a list of block-listed domains.[8] The number of "bad" links is added as the negative measure BlocklistedCount.

[7] `https://www.mywot.com`
[8] `https://github.com/hemiipatu/Blocklists.git`

Reranking The measures (as shown in Table 1) are weighted, normalized between 0 and 100, and then combined into the scores relevance, support and credibility.

Table 1. Assignment of measures to scores and their weights

Scores	Weights	**Measures**	Weights
Relevance	.4	ArgumentRatio	.4
		ComparisonCount	.4
		PageRank	.2
Support	.4	SemanticRatio	.5
		SentimentRatio	.3
		SupportCount	.2
Credibility	.2	WOT	.4
		SpamRank	.3
		BlocklistedLinks	.2
		ARI	.02
		DaleChall	.04
		Flesch	.04

Finally, the three resulting scores are also weighted and then combined into the final score by which the documents are re-ranked as the final result of our search engine.

4 Evaluation

We now present the results of the evaluation conducted by the CLEF committee to rate all participating systems. The ranked list of retrieved documents was judged by human assessors on the three dimensions document relevance, argumentative support, and trustworthiness and credibility of the web documents and arguments. With the introduced search engine, we reached the best submitted run according to NDCG@5 with a score of 0.580. The combination of different techniques and approaches has proven promising. As they have different strengths and weaknesses, there is a potential to balance each other out. Nevertheless, a processing pipeline consisting of so many steps suggests a detailed evaluation and examination of the propagation of errors through the phases of the model.

5 Discussion

We participated in the Touché Lab on Argument Retrieval at CLEF 2020 with a web-scale search engine capable of answering comparative questions resulting in the best submitted run according to NDCG@5. However, each step in the

comparison retrieval model could be explored further, refined or be tackled with other methods. Whereas the task at hand requires to build a complete search engine, the extensive study of each part could have been subject to a research project alone. Future work can tie in at various points. From comparative relation extraction, over identifying comparative, argumentative and support sentences, to a learning-to-rank algorithm, the question how a machine learning approach could perform almost imposes itself upon the research community. Widening the capabilities of the system to cope not only with the given set of user queries but with any comparative question in natural language can be seen as a further challenge.

References

1. Adler, B., Bosciani-Gilroy, G.: Real-time claim detection from news articles and retrieval of semantically-similar factchecks. In: Proceedings of the NewsIR'19 Workshop at SIGIR (2019)
2. Aker, A., Sliwa, A., Ma, Y., Lui, R., Borad, N., Ziyaei, S., Ghobadi, M.: What works and what does not: Classifier and feature analysis for argument mining. In: Proceedings of the 4th Workshop on Argument Mining. pp. 91–96 (2017)
3. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl. In: Azzopardi, L., Hanbury, A., Pasi, G., Piwowarski, B. (eds.) Advances in Information Retrieval. 40th European Conference on IR Research (ECIR 2018). Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York (Mar 2018)
4. 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)
5. Braunstain, L., Kurland, O., Carmel, D., Szpektor, I., Shtok, A.: Supporting human answers for advice-seeking questions in cqa sites. In: Ferro, N., Crestani, F., Moens, M.F., Mothe, J., Silvestri, F., Di Nunzio, G.M., Hauff, C., Silvello, G. (eds.) Advances in Information Retrieval. pp. 129–141. Springer International Publishing, Cham (2016)
6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Conference: the 22nd ACM SIGKDD International Conference. pp. 785–794 (08 2016). https://doi.org/10.1145/2939672.2939785
7. Chernodub, A., Oliynyk, O., Heidenreich, P., Bondarenko, A., Hagen, M., Biemann, C., Panchenko, A.: TARGER: Neural argument mining at your fingertips. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 195–200. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-3031, `https://www.aclweb.org/anthology/P19-3031`
8. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 670–680. Association for Computational Linguistics, Copenhagen, Denmark (September 2017), `https://www.aclweb.org/anthology/D17-1070`

9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
10. Fromm, M., Faerman, E., Seidl, T.: Tacam: Topic and context aware argument mining. IEEE/WIC/ACM International Conference on Web Intelligence on - WI '19 (2019). https://doi.org/10.1145/3350546.3352506, http://dx.doi.org/10.1145/3350546.3352506
11. Gao, S., Tang, O., Wang, H., Yin, P.: Identifying competitors through comparative relation mining of online reviews in the restaurant industry. International Journal of Hospitality Management **71**, 19–32 (2018)
12. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings. convolutional neural networks and incremental parsing **7**(1) (2017)
13. Janardhan Reddy, A., Rocha, G., Esteves, D.: Defactonlp: Fact verification using entity recognition, tfidf vector comparison and decomposable attention. arXiv pp. arXiv–1809 (2018)
14. Jensen, M., Lowry, P., Jenkins, J.: Effects of automated and participative decision support in computer-aided credibility assessment. Journal of Management Information Systems **28**, 201–233 (07 2011). https://doi.org/10.2307/41304610
15. Jindal, N., Liu, B.: Mining comparative sentences and relations. In: Aaai. vol. 22, p. 9 (2006)
16. Kutuzov, A., Fares, M., Oepen, S., Velldal, E.: Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In: Proceedings of the 58th Conference on Simulation and Modelling. pp. 271–276. Linköping University Electronic Press (2017)
17. Lippi, M., Torroni, P.: Argumentation mining. ACM Transactions on Internet Technology **16**, 1–25 (03 2016). https://doi.org/10.1145/2850417
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008), http://nlp.stanford.edu/IR-book/
19. Panchenko, A., Bondarenko, A., Franzek, M., Hagen, M., Biemann, C.: Categorizing Comparative Sentences. In: Stein, B., Wachsmuth, H. (eds.) 6th Workshop on Argument Mining (ArgMining 2019) at ACL. Association for Computational Linguistics (Aug 2019), https://www.aclweb.org/anthology/W19-4516
20. 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
21. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2012). p. 1004. ACM (Aug 2012). https://doi.org/10.1145/2348283.2348429
22. Rafalak, M., Abramczuk, K., Wierzbicki, A.: Incredible: is (almost) all web content trustworthy? analysis of psychological factors related to website credibility evaluation. Proceedings of the companion publication of the 23rd international conference on World wide web companion pp. 1117–1122 (04 2014). https://doi.org/10.1145/2567948.2578997

23. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
24. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and Clustering of Arguments with Contextualized Word Embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 567–578. Florence, Italy (07 2019), https://arxiv.org/abs/1906.09821
25. Rinott, R., Dankin, L., Perez, C.A., Khapra, M.M., Aharoni, E., Slonim, N.: Show me your evidence - an automatic method for context dependent evidence detection. In: EMNLP (2015)
26. Schildwächter, M., Bondarenko, A., Zenker, J., Hagen, M., Biemann, C., Panchenko, A.: Answering Comparative Questions: Better than Ten-Blue-Links? In: Halvey, M., Ruthven, I., Azzopardi, L., Murdock, V., Qvarfordt, P., Joho, H. (eds.) 2019 Conference on Human Information Interaction and Retrieval (CHIIR 2019). ACM (Mar 2019). https://doi.org/10.1145/3295750.3298916
27. Smith, E., Senter, R.: Automated readability index. AMRL-TR. Aerospace Medical Research Laboratories (6570th) **iii**, 1–14 (06 1967)
28. Varathan, K.D., Giachanou, A., Crestani, F.: Comparative opinion mining: a review. Journal of the Association for Information Science and Technology **68**(4), 811–829 (2017)
29. Weischedel, R.: OntoNotes Release 5.0 LDC2013T19. Web Download. Linguistic Data Consortium, Philadelphia (2013)
30. Xu, K., Liao, S.S., Li, J., Song, Y.: Mining comparative opinions from customer reviews for competitive intelligence. Decision support systems **50**(4), 743–754 (2011)