Development of a Search Engine to Answer Comparative Queries

Notebook for the Touché Lab on Argument Retrieval at CLEF 2020

Johannes Huck[1]

Martin-Luther Universität Halle-Wittenberg, Halle, Germany
johannes.huck@student.uni-halle.de

1 Introduction

Comparative search queries are often used to express the information need that demands argumentation for and against compared options, e.g. whether X is better than Y or Z. For example a user could ask: *Is Windows better than Linux or Mac?* With this query, the user is telling implicitly that they want to know reasoned arguments about the aforementioned operating systems rather than biased opinions. Current search engines will retrieve web pages with information about one of the operating systems, or there will be the manufacturer's sites that are heavily biased. Ideally, we want to retrieve web pages which contain a comparison between the arguments X, Y and Z.

During the winter semester 2019/2020 I have participated in the Shared Task 2 of the *Touché Lab on Argument Retrieval at CLEF 2020* [2]. Therefore, I have developed a search engine to answer comparative queries. The task was to retrieve ranked documents from the web crawl *ClueWeb12* to answer the user's comparative questions. A collection of 50 topics, that represent a user information need, has been provided. I used *ChatNoir* [1] to retrieve and rank document candidates for futher re-ranking. To extract arguments from the documents, I used a library *TARGER* [5] that provides machine learning based-algorithms for arguments mining. After mining arguments, I am able to create one document per web page based on the arguments found on this particular web page. The extracted arguments are then used to construct a search index. In the search scenario, the question will be searched on the index. The developed search engine comprises three components and a XML parser, which reads the XML file, the file contains the aforementioned 50 topics. The first component retrieves web pages via *ChatNoir* and extracts the arguments found on the web pages. Then the second component constructs the search index for my search engine. For this task I decided to use the library *whoosh* [4] written in *Python*. The third component searches on the created index. The mentioned components will be discussed in chapter 2. I decided to code the entire search engine with *Python*.

2 Approach

2.1 Argument Extraction

The first component mines the arguments from web pages. I am sending the found questions to *ChatNoir*. This results in a response from *ChatNoir* which contains the first 20 web pages returned by *ChatNoir*. Then I am using the content extraction tool *BoilerPipe* to extract the plain text from the web pages. Because there is a lot of unnecessary text on the web pages, I need *BoilerPipe* to only extract the main body instead of the whole web page. So each web page returned by *ChatNoir* will be then cleaned by *BoilerPipe* and transmitted to *TARGER*. The response of *TARGER* contains information about the arguments found in the extracted main body of the web page. This information is encoded using the *IOB* format. IOB is short for inside, outside, beginning. It is a common format for tagging tokens. For argument mining this means: The *B* tag indicates that the associated token is the beginning of a new argument. The *I* tag indicates that the token is within an argument, and the *O* tag indicates that this token does not belong to an argument. Additionally, a *C* for a found claim or a *P* for a found premise will be used. An example would look like this:


```
{
  "label": "C–B",
  "prob": "0.7497254",
  "token": "Quebecan"
},

{
  "label": "P–B",
  "prob": "0.99999774",
  "token": "In"
},
{
  "label": "P–I",
  "prob": "0.9999943",
  "token": "the"
},
```


With this I am able to extract the arguments from the web page. Each block within the response of *TARGER* is one argument, so I can generate a list of arguments for each web page. Then I am creating one document per web page, and I am writing one argument per line into these documents. For the question *What is the difference between sex and love?* some extracted arguments are:

```
Differences Between Love & Sex,
Love and sex are NOT the same thing,
Love is an emotion or a feeling,
Sex may or may not include penetration,
Love involves feelings of romance and/or attraction,
Sex : Sex is an event or act ( physical ),
```

2.2 Constructing the Index

After processing all the results from *ChatNoir*, my second component will construct the index. The index will only contain the arguments from the documents created in the first component and the title of the document which is in the form *Question_ClueWeb12ID*. I am using the Stemming-Analyzer provided by *whoosh* [4]. This means that I use a *Tokenizer*, a *Porter-Stemmer*, a *Stop-Word-Filter* and a *Lower-Case-Filter* to extend the index. In addition to my arguments there are tokens and word stems, stop words have been removed and all words have been converted to a lower case version. With this index, I am able to search efficiently and my system can retrieve relevant documents based on the extracted arguments.

2.3 Searching and Ranking

I have chosen the *Okapi BM25* ranking function because it is a common and well-known ranking function and is sufficient for my approach. For each topic, the approach retrieves the top 20 documents using the index. The question will be tokenized into their words. Each word will be linked with a logical OR. Additionally, documents that contain more words from the question will be ranked higher. A document is ranked higher the more frequent the question's words appear in it. Because we want documents that answer a comparative question, documents that only contain arguments for one important term are not as relevant as documents that contain arguments for more terms or all terms from the question. As the last step, a result file will be created with the standard TREC format.

3 Evaluation

The system has been successfully deployed on the evaluation platform *Tira* [6]. The described approach can be found under the user name *ir-lab-mlu-gruppe-2*, the evaluation can be found under the name *Inigo Montoya*. *Tira* is a site that is being used to evaluate the results of the participants of the Shared Task, and it enables the option to reproduce the results with the described approach. The reported nDCG@5 of my approach on *Tira* is 0.567 while the nDCG@5 of the baseline is 0.568 [3]. Additionally, the approach has been evaluated manually with the top 5 answers of 5 chosen topics, to see why my approach is slightly worse than the baseline.

3.1 First Topic

The first topic we will look at is *What is the difference between sex and love?* The first thing I encountered while looking at the results was that the first 10 results for this question were the same web pages, but with 10 different Clue Web12 IDs. After those 10 results, there are different web pages. While comparing the results, I realized that the web page from rank 1 to 10 does not answer the question in a way someone would hope. It is rather about the difference between rape and love. The second result after those results is way better than the former result. It has a good and quick overview of

love and sex with some bullet points about the two terms. The third best result is just a block of text, which explains the difference between the two terms. So manually I would rank the second result as rank 1, the third as rank 2 and as rank 3 the first 10 results.

3.2 Second Topic

The next topic we will look at is *Which browser is better Internet Explorer or Firefox?*. The first result gives a brief overview of the two browsers. The second one is a lot more in-depth about the two browsers and even covers more browsers than the user wanted. Sometimes the user does not know there are more alternatives, and this would help to show the user those alternatives. The third result is way more biased than the first two results. It is more an opinion, and it is not based on reasoned arguments, opposite to the user's expectation of evidence-based arguments. The fourth and fifth results are very similar pages with statistics about the usage of the Internet Explorer. Firefox is completely missing on those pages. There are some change notes about different versions of the Internet Explorer. The fourth result is a slightly newer version of the fifth result. I would only swap rank one with two if I would rank it per hand. The other results are correctly ranked.

3.3 Third Topic

The next topic will be *Which is better, Canon or Nikon?*. The first result is a very in-depth review about the Nikon J1, but there are comparisons between this Nikon model and a similar Canon model. The next four results are comparisons between Nikon and Canon models. While the third result is the comment section of a review page which contains heavily biased opinions rather than facts, the second, fourth and fifth results are written by the same author who belongs to the technology information site *Lifewire*. This seems trustworthy and the reviews are written shortly and informatively. I would rank the second, fourth and fifth results as rank one, two and three and the first and third results should be ranked lower because they are not as helpful as the other results.

3.4 Fourth Topic

The fourth topic will be *Which is better, laptop or desktop?*. The first result is a review about the Dell Vostro laptop, but there is also a link to the review page about the Dell Vostro desktop. The second result is a page about different laptop manufacturers with pros and cons of their products. The third result is a comparison between consumer and business laptops, and answers the question which of them is better. The fourth result is a comparison between desktop and laptops, and there are information about which group of people should rather buy a laptop or a desktop. The fifth result is about the advantages of MacBooks in comparison to other laptop manufacturers. I would rank the fourth result as rank one and then result one, two, three and then finally five.

3.5 Fifth Topic

The last topic we are looking at is *Which is a better vehicle: BMW or Audi?*. The first result is a review of the 2010 BMW X6 and gives information about the specific car model. The second result is a comparison between a BMW, an Audi and a Mercedes-Benz model. This page gives a detailed overview of the models and compares them in different categories. The third and fourth results are reviews of two different BMW models. Those results are from the same site as the first one. The fifth one is a review page about an Audi model, also from the same site as the aforementioned results. I would swap rank one with two if I would rank it manually. The other results are correctly ranked.

4 Conclusion

The evaluation has shown that the ranking of my system is not perfect. For the topics we have looked at, there are some ranks I would rank differently when ranking it manually. The first results seem never to be the most relevant document. We have also seen that there are some duplicates that need to be found and eliminated because the user needs a good variety of facts rather than the same page over and over again. Topics including the question which brand is better than another brand are quite challenging because there are different models of products which are having their pros and cons and we will never be able to find an in-depth comparison of all relevant products. This is even worse when companies are producing different kinds of products and the question is therefore ambiguous. Better performance can be probably achieved by optimizing the parameters of BM25. Boilerpipe seems to produce some noisy content, there are a lot of pages that contain boilerplate or other unnecessary content like ads or links to other irrelevant pages. That unnecessary content is not filtered enough. The noisy content produced by Boilerpipe is then negatively affecting subsequent steps like argument mining. So the extracted arguments contain a lot of useless strings which need to be filtered out.

To conclude, the proposed system shows promising results, but requires improvements in terms of content extraction and tuning the retrieval model.

References

1. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Elastic chatnoir: Search engine for the clueweb and the common crawl. In: ECIR (2018)
2. 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)
3. Bondarenko, A., Hagen, M., Fröbe, M., Beloucif, M., Biemann, C., Panchenko, A.: Touché task 2: Comparative argument retrieval - results (2020), https://events.webis.de/touche-20/shared-task-2.html#results
4. Chaput, M.: Whoosh. https://whoosh.readthedocs.io/en/latest/intro.html (2007–2012)
5. Chernodub, A.N., Oliynyk, O., Heidenreich, P., Bondarenko, A., Hagen, M., Biemann, C., Panchenko, A.: Targer: Neural argument mining at your fingertips. In: ACL (2019)
6. 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