

Exploring BERT Synonyms and Quality Prediction for Argument Retrieval

Notebook for the Touché Lab on Argument Retrieval at CLEF 2021

Tommaso Green¹, Luca Moroldo¹ and Alberto Valente¹

¹University of Padua, Department of Information Engineering, Via Gradenigo 6/b 35131 - Padova Italy

Abstract

This paper attests the participation of the Yeagerists team from University of Padua in the Touché @ CLEF 2021 challenge [1, 2], specifically in the Argument Retrieval for Controversial Questions shared task. The project has been submitted as part of the Search Engines course (a.y. 2020-21) held by professor N.Ferro for the Master's Degree in Computer Engineering. We show our retrieval pipeline architecture and discuss our approach, which employs a DirichletLM retrieval model coupled with transformer-based models for both query expansion and argument quality re-ranking. For the first, after having explored several possibilities, we decided to deploy a BERT-based synonym substitution technique. For argument quality re-ranking, we built an approach based on previous work and explored how different models from the BERT family performed in predicting a quality score for a given argument. As a summary of our main findings, we tested different configurations of our system and achieved a retrieval performance of 0.8279 nDCG@5 on Touché 2020 topics, slightly improving over the task baseline, while on 2021 topics we managed to align with the provided baseline at 0.6246 nDCG@5.

Keywords

Information Retrieval, Argument Retrieval, Automatic Query Expansion, Argument Quality, Transformer, BERT

1. Introduction

Search engines are nowadays one of the most important means to retrieve information, however, they are not specialised for tasks related to the retrieval of nuanced and complex information. As the influence of search engines in opinion formation increases, it is of paramount importance for them to be able to address citizens' enquiries on both general matters ("Should the death penalty be allowed?") or personal decisions ("Should I invest in real estate?") with relevant and high-quality results. This type of task, called *argument retrieval*, requires the system to respond to user queries with arguments, which could be defined as a set of premises with evidence that leads to a conclusion. Often arguments have a stance, i.e. the author's position (in favour or against) on the debated subject. Clearly, this scenario requires finding a proper trade-off between how relevant an argument is with respect to the user query and its intrinsic quality.


This paper summarizes our submission to the Touché @ CLEF 2021 shared task on Argument Retrieval for Controversial Questions [1, 2] which was centred on the usage of transformer-based

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

✉ tommaso.green@studenti.unipd.it (T. Green); luca.moroldo@studenti.unipd.it (L. Moroldo); alberto.valente.3@studenti.unipd.it (A. Valente)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

models for query expansion and argument quality re-ranking.

As a brief introduction, our approach consists of two phases: in the first, we use a simple Lucene-based searcher with our query expansion subroutine on top of it. In the second phase, the relevance scores of retrieved documents are combined with the document quality scores provided by our argument quality module. We show that by combining these components we can achieve competitive performance and, as far as Touché 2020 topics are concerned, even surpass the baseline score by properly selecting similarity functions, ranking strategies and other model-specific parameters.

The paper is organized as follows: Section 2 introduces related works; Section 3 describes our approach; Section 4 provides details of our experimental setup; Section 5 discusses our main findings; finally, Section 6 draws some conclusions and outlooks for future work.

2. Related Work

2.1. Argument Search Engines

As described in Wachsmuth et al. [3] an argument comprises a statement, i.e. the author's position on the topic, and premises, which are usually supported by evidence. Several sub-tasks constitute this area of research: argument mining, i.e. the extraction of an argument from raw text, argument retrieval, with the related techniques to increase the recall of the system such as *Query Expansion (QE)*, and argument re-ranking, so as to consider other parameters in addition to the relevance score, for example, argument quality.

Some recent approaches in developing self-contained argument search engines include the args.me search engine introduced in Ajjour et al. [4]. A similar approach was used by IBM's *Project Debater* [5], where a topic-classifier was used to mine arguments from recognized sources (e.g. Wikipedia) at index-time. Differently from the above mentioned, ArgumenText [6] is more similar to web search engines as it indexes entire documents and delays argument mining to query-time.

2.2. Query Expansion

Query Expansion (QE) is a technique that consists in expanding a user query to increase its effectiveness in the search process, mainly by reducing its ambiguity and inserting new related keywords. As reported in Azad and Deepak [7], one of the best candidates for implementing a successful query expansion routine is to make use of the recently developed transformer-based models [8], which proved impressive performance and ability to grasp semantic nuances. For this reason, our query expansion approach aligned to that of Victor [9], who experimented on how contextual embeddings produced by a *Masked Language Model (MLM)* such as BERT [10], can be applied in generating query expansion terms. Furthermore, we took inspiration from the idea [11] of using an end-to-end BERT-based terms substitution approach, which proposes and validates substitute term candidates based on their influence on the global contextualized representation of the query. A big difference from [11] is that we do not apply dropout to target word embedding to partially mask it, but we completely mask the word and let BERT generate a large enough batch of candidates from which to choose the best ones.

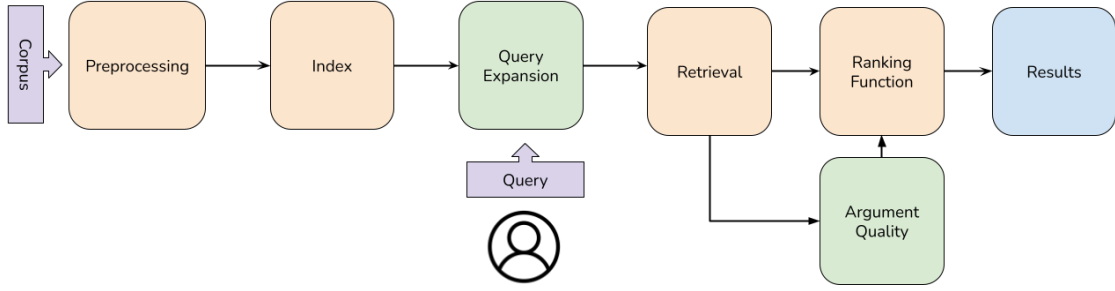


Figure 1: Summary of our pipeline: inputs in purple, neural modules in green.

2.3. Argument Quality

Wachsmuth et al. [12] provide a categorization of the key components that contribute to the overall quality of an argument: *logical quality* considers how sound an argument is, i.e. how the conclusion follows from the premises, *rethorical quality* measures how persuasive an argument is for the audience and *dialectical quality* represents the effectiveness of the argument in making the audience create their own stance. The authors stress that dialectical quality builds on rhetorical, and rhetorical builds on logical. The application of transformer-based pre-trained language models to argument quality prediction was explored in Gretz et al. [13], where they provide an extensive dataset on argument quality called *IBM-Rank-30k*. They compared the performance of a Bi-LSTM with GloVe embeddings and *Support Vector Regressor (SVR)* with 3 different variants of BERT: BERT-Vanilla, where the [CLS] tokens of the last 4 encoder layers were concatenated and fed to a feed-forward neural network with ReLU and sigmoid activation to produce a score in $[0, 1]$, BERT-FineTune where they fine-tuned the entire model (BERT encoder and linear layer on top), and finally BERT-FineTuneTOPIC, identical to BERT-FineTune but with the input enriched with the topic as in [CLS]<topic>[SEP]<query>[SEP].

3. Methodology

Our solution is composed of 3 parts: a Lucene-based indexer and searcher, a query expansion module and a quality measurement module. The procedure consists of reading the topics (i.e. queries) from a file, performing query expansion for each topic, running the searcher to retrieve a set of arguments, measuring the quality of a portion (of size n_{rerank}) of the retrieved arguments, and (re)ranking the arguments using the quality score. Figure 1 summarizes our pipeline.

When query expansion is enabled, the set of expanded queries obtained from a topic may retrieve duplicated arguments. Duplicated arguments are removed right after the retrieval step, and only the top n_{rerank} arguments are given as input to the argument quality module.

3.1. Indexing and searching

We indexed the Args.me corpus [14] by tokenizing and storing the body and title of each argument. Besides tokenization, only a lowercase filter was applied. For the retrieval part, we used the Dirichlet similarity by matching the query terms with both the title and the body of the arguments, we used a title boost parameter to weight the score assigned to a match of a query term with a title term, while the score assigned to a match with a body term is left untouched.

3.2. Query Expansion

The Query Expansion subroutine generates a new set of queries for each query of a list, which in our case is the list of Touché topics. It works as follows:

1. The query is tokenized and *Part of Speech (PoS)* tagged; the query tokens which are recognized as nouns, adjectives or past participles are masked and then replaced with BERT's [MASK] special token.
2. Use BERT to generate the best 10 tokens that, according to its bidirectional attention mechanism, fit in place of each [MASK] token.
3. Compute the BERT embeddings of these 10 tokens and compare them, using cosine similarity, to the embedding of the original token: more about this in section 4.4.
4. Perform a two-phase screening, where at first we keep only the best tokens among the 10 that have a similarity score above 85%. If none of the generated tokens is good enough, we lower the similarity score threshold to 75% and use BERT again to generate batches of 20 new candidates each, until at least one good substitute is found or 100 candidates in total are generated.
5. Using the lists of new tokens for each position of [MASK], we compute their cartesian product to compose the list of all the possible new queries and take from it a set of *max_n_query* queries at random.

3.3. Argument Quality

Building on Gretz et al. [13], we decided to explore the possibility of using pre-trained language models for predicting the overall quality of an Argument. We make a distinction in the training process of models made of a transformer-based encoder and a traditional feedforward neural network:

- Adaptation: while keeping the weights of the encoder frozen, only the dense layer on top is trained on the new task;
- Fine-tuning: the whole model is fine-tuned on the new task.

Both approaches have advantages and disadvantages, and in the end, we decided to go with the first approach as it reduces model complexity (lowering the possibility of overfitting) and requires fewer computational resources. In addition to this, as reported in [15] fine-tuning seems to be more appropriate when pre-training task and transfer tasks are closely related. This was not the case as BERT [10] for example is pre-trained on MLM and *Next Sentence Prediction (NSP)*

(both can be framed as classification tasks) and the target task required to produce a real-valued score (regression task), so we decided to proceed with adaption.

The model works as follows: given an argument (truncated at 512 tokens), an encoded representation is computed using the [CLS] embedding of one of the models mentioned below. Finally, this representation is passed to a feedforward neural network with 2 hidden layers which computes the *Mean Square Error (MSE)* loss w.r.t. the original target value.

3.4. Ranking Functions

We decided to apply the argument quality re-ranking only to $D_{n_rerank}^q$ i.e. the top n_rerank arguments according to the relevance score for each query q . For the final list of results, we had to combine relevance and quality scores. We tried three different ranking functions, using a parameter $\alpha \in [0, 1]$ that represented the importance of the quality score for a document d with relevance score $r(d)$ and a quality score $q(d)$:

- Normalization function: w.r.t. a query q , we re-ranked the list by normalizing relevance and quality scores:

$$R(q, d) = (1 - \alpha) r_{norm} + \alpha q_{norm} = (1 - \alpha) \frac{r(d)}{\max_{d' \in D_{n_rerank}^q} r(d')} + \alpha \frac{q(d)}{\max_{d' \in D_{n_rerank}^q} q(d')} \quad (1)$$

- Sigmoid Function: in order to compress relevance and quality scores, we decided to use the sigmoid function, $\sigma(x) := \frac{1}{1+e^{-x}}$, with a scale parameter β . The function used was:

$$R(q, d) = (1 - \alpha) \sigma(\beta r(d)) + \alpha \sigma(\beta q(d)) \quad (2)$$

The parameter β was used to counter the typical “squishing” of the sigmoid, which maps large positive or negative values into close output values (closer to 1 or 0 respectively), while values near the origin can assume a wider spectrum of values. As β tends to 0, the steepness of the sigmoid decreases and allows for a wider neighbourhood of values centred in the origin to get more diverse values.

- Hybrid Function: in this case, we applied the sigmoid to the quality scores while normalizing relevance scores:

$$R(q, d) = (1 - \alpha) r_{norm} + \alpha \sigma(\beta q(d)) = (1 - \alpha) \frac{r(d)}{\max_{d' \in D_{n_rerank}^q} r(d')} + \alpha \sigma(\beta q(d)) \quad (3)$$

4. Experimental setup

To reproduce our results, please refer to our GitHub repository¹. Our submission was done through the provided TIRA [16] virtual machine to allow for our approach to be deployed on different data of the same format in the future.

¹<http://github.com/tommaso-green/yeagerists-clef-touche-2021>

4.1. The Args.me corpus

The Args.me corpus [14] consists of 5 collections of arguments obtained from different sources. Each collection is a list of arguments containing a variety of fields: the body, which is the text written by the user to support a claim; a stance, that can be “PRO” or “CON” w.r.t. the parent discussion; a title, which summarizes the discussion that the argument belongs to. The title can be written by the author of the argument or inherited from the discussion and is sometimes empty. We decided to discard duplicated arguments having the same ID as well as arguments having an empty body.

4.2. Indexing

We developed a Lucene-based program to index the Args.me corpus. Each indexed document contains the ID, body, title, and stance of an argument parsed from the Args.me corpus. The original content of the body and the title is stored in the index. The Analyzer, which is responsible for the tokenization of the body and title of an argument, is composed of a Lucene StandardTokenizer and a LowerCaseFilter as we found that using any other kind of analyzer downgraded the nDCG@5 score, especially any stemming filter. Stopwords were not removed.

4.3. Searching

Given the text of a query, e.g. any Touché topic, we parsed it using Lucene’s MultiFieldQuery-Parser. This parser allowed us to search for any match in both the body and the title (when present) of an argument, assigning different boosts depending on the field where the match occurred. By default, the boost assigned to a match with the title is set to zero as we disabled this feature because it did not improve our solution. The retrieval model that we used is the Lucene LMDirichletSimilarity, with $\mu = 2000$, and for each topic we always retrieved 100 documents. As a consequence, when query expansion is enabled, a set of k expanded queries produces up to $k \times 100$ arguments, some of which may be duplicated. We remove duplicated arguments from the results obtained from the same topic, we select the top n_{rerank} arguments using the DirichletLM score, and pass them to the argument quality module.

4.4. Query Expansion

As an implementation choice, we decided to consider as embeddings of the tokens generated by BERT the 3072-dimensional vectors obtained concatenating the token representations (of size 768) from the last 4 layers of the *bert-base-uncased* deep neural network. This choice was based on a tradeoff between the fact that they should be framed to retain as much information as possible in terms of *context classification* while remaining small enough. Moreover, this is the best performing configuration for using BERT contextual features according to the authors themselves [10]. To perform PoS tagging on each query to expand we used the model *averaged-perceptron-tagger*² from the NLTK³ (Natural Language Toolkit) Python library. We set the number *max_n_query* of new queries generated from each query to 10.

²<https://www.kaggle.com/nltkdata/averaged-perceptron-tagger>

³<https://www.nltk.org/>

4.5. Argument Quality

4.5.1. The Argument Quality Dataset

To build our argument quality predictor, we decided to use the Argument Quality Dataset of Gienapp et al. [17], as it contains 1271 arguments extracted from the args.me corpus, each having detailed quality scores as well as topical relevance. Quality scores were obtained by almost 42k pairwise judgements. We made our model predict only the overall quality, which can be obtained from the rhetorical, logical and dialectical quality as described in [17], using a train-validation-test split of 80%-10%-10%. Both hyperparameter selection and training were performed in a deterministic way by setting a specific seed.

4.5.2. Model Selection

Differently from Gretz et al. [13], we decided to explore 4 different models of the BERT family: BERT [10], DistilBERT [18], RoBERTa [19] and ALBERT [20]. BERT is by far the most famous one using the transformer encoder to create bi-directional contextualized word representations, and several variants have been published due to its performance in state-of-the-art NLP. Specifically, we used the HuggingFace [21] implementations of the aforementioned models: bert-base-uncased, distilbert-base-uncased, albert-base-v2 and roberta-base.

For the hyperparameter selection, an exhaustive grid search was performed over the following set of parameters: learning rate (lr), Adam weight decay (w) [22], batch size (b) and dropout probability (p). If the latter is set to 0, a simple feedforward neural network with 2 hidden layers and ReLU activations was used, otherwise AlphaDropout [23] was applied to the [CLS] embedding and to the hidden activations of the feedforward neural network. In that case, ReLUs were substituted with SeLUs [23].

For all configurations, we tracked the R^2 score on both the training set and validation set and picked the best combination for each of the 4 models according to R^2 on the validation set. The 4 selected models were then trained for 30 epochs, using early stopping and saving the best model according to validation R^2 . Finally, the model performance was assessed on the remaining test set. Selected models and results are available in Table 1. Every experiment was logged using the online WandB logger [24].

5. Results and Discussion

5.1. Argument Quality Prediction Results

The four final models had very similar results in terms of validation R^2 score. This also holds true for the results on the test set, in particular with BERT and DistilBERT being close and slightly better than the other two as you can see in Table 1. Compared to results on the same dataset obtained by [25], we were able to produce a more powerful regressor since they report that the MSE loss of their best (ensemble) model on the same dataset is 1.322 for combined quality, while our best result in terms of loss is 0.718. This is to be expected, as they used a simple SVR using pre-processed textual features while in our case we could count on the full word-level attention typical of transformer models.

Table 1

Results on the test set for argument quality prediction.

Model	lr	w	b	p	Test R^2	Test MSE Loss
BERT	0.005	0.0005	16	0	0.7439	0.7280
DistilBERT	0.005	0.0001	16	0	0.7412	0.7440
RoBERTa	0.005	0.0001	16	0	0.735	0.7187
ALBERT	0.0001	0.0005	32	0	0.703	0.7494

Table 2

List of 10 selected runs, sorted in decreasing order of nDCG@5 (2020). Runs sent to Touché are highlighted (our baseline is in light blue, while Swordsman baseline is in yellow). The title boost was disabled.

Run name	QE	α	β	n_{rerank}	Quality model	$R(q, d)$	nDCG@5 (2020)	nDCG@5 (2021)
lunar-sweep-201	no	0.75	-	5	BERT	normalize	0.8279	0.6241
chocolate-sweep-50	no	0.1	2	5	BERT	sigmoid	0.8273	0.6241
volcanic-sweep-138	no	0.5	0.8	5	BERT	sigmoid	0.8271	0.6246
swordsman baseline	no	-	-	-	-	-	0.8266	0.6260
lunar-sweep-58	no	0.1	0.2	5	RoBERTa	hybrid	0.8230	0.6104
vague-sweep-rev-204	no	0.75	1.1	5	ALBERT	sigmoid	0.8229	0.6211
lucene baseline	no	0	-	-	-	normalize	0.8224	0.6095
sage-sweep-rev-160	no	0.5	1.5	5	RoBERTa	sigmoid	0.8093	0.6211
distinctive-sweep-110	no	0.1	0.8	15	ALBERT	hybrid	0.7992	0.6177
good-sweep-85	yes	0.1	0.3	15	DistilBERT	hybrid	0.6857	0.5357
lucene-query-exp	yes	0	-	5	-	normalize	0.6801	0.5384

5.2. Overall Pipeline Results

5.2.1. Touché 2020 Topics

To provide an overview of our retrieval pipeline performance, we run it on topics from Touché 2020 with different combinations of parameters by using WandB Sweeps [24]. Of these we selected the 10 most interesting runs: we made this choice not only by considering their nDCG@5 score, but also their diversity in terms of hyperparameters. To compute nDCG@5 we used *trec_eval*⁴(version 9.0.7), and thus the scores may present very slight differences with respect to those in the official Touché leaderboard as the evaluators used a different tool, meaning that the reported *swordsman baseline* (*DirichletLM*) scores are not precisely comparable to the others.

Table 2 shows the results in decreasing order of nDCG@5 (2020) score: we can see that the *swordsman baseline* is not very far from our *lucene baseline*, obtained by using just *DirichletLM* with the title boost set to zero, and the best score of our solution, which presents an improvement of 0.16%. We expected this difficulty in improving the baseline with deep learning methods as this was also proven in Touché 2020 [26] and may be due to the challenge of developing

⁴https://trec.nist.gov/trec_eval/

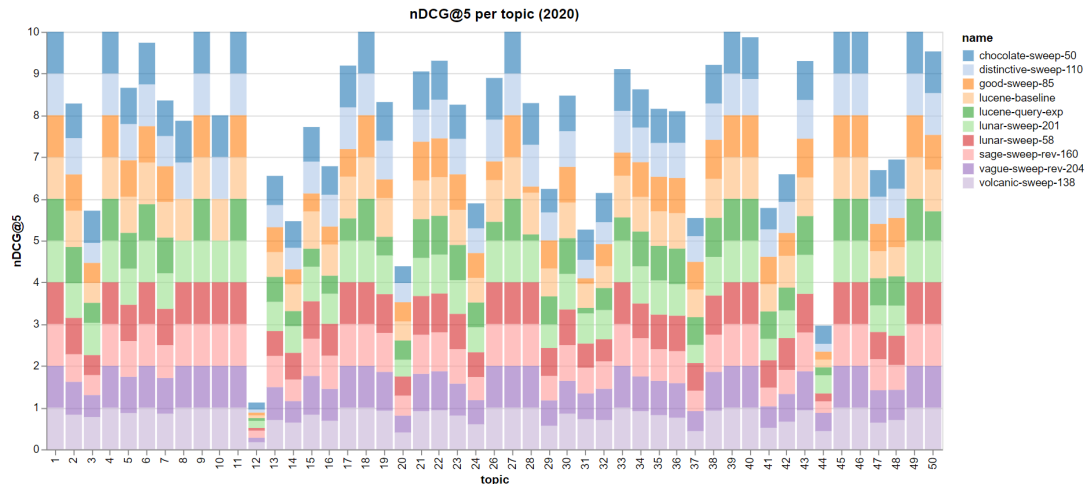


Figure 2: Bar plot of nDCG@5 scores for the 10 selected runs grouped by topic (Touché 2020 topics).

efficient modules for argument quality prediction and query expansion.

As it is clear from the last two rows of Table 2, we did not get any improvement by using query expansion, both combining it with argument quality reranking (as in *good-sweep-85*) or applying it directly on our DirchetLM baseline (see *lucene-query-exp*). This was an expected outcome since some queries were well expanded, so they managed to produce a set of synonyms that allowed the searcher to retrieve a wider range of arguments that fall in the same topic, while many others were subject to changes in their original meaning, leading to a score that is lower than what they would have achieved without these incoherent queries. This phenomenon is linked both to the size of the original query, as a masked short query is more likely to lack the context BERT needs to work properly, and to how easily replaceable its terms are, so even setting a higher number of generated queries (*max_n_query*) would not have improved the expansion quality in any noticeable way.

A few combinations of parameters allowed the quality re-ranker to slightly improve the baseline score. This happens when the re-ranker permutes the top-5 arguments, i.e. changing only the order of the results that influence nDCG@5 without adding any new argument.

As mentioned in 4.3, the boost assigned to matches occurring on the title of an argument did not lead to better results: one reason may be that this boost pushes up in the ranked list arguments that inherit the title from the parent discussion without containing a relevant body. For this reason, we did not log any test with titleboost different than 0 to concentrate on other parameters.

Figure 2 summarizes the performance of the 10 selected runs on each Touché 2020 topic. This allows us to see that, while at least ten topics lead consistently to optimal performance among all the runs, a few others perform especially bad. Let’s discuss some examples of the latter case:

- Topic 8 (“Should abortion be legal?”) gets an nDCG@5 score of 0 on both runs which include query expansion (*good-sweep-85* and *lucene-query-exp*). This is clearly linked to how this topic is expanded, as we have noticed that BERT consistently recognizes some

specific pairs of words as very similar. In this topic the terms “abortion” and “divorce”, although being far from having the same meaning, are being replaced in new queries interchangeably. This is almost certainly due to bias in BERT’s pre-training dataset, where these two words used to appear together very often.

- Also topic 10 (“Should any vaccines be required for children?”) manages to get an nDCG@5 of 0 on the same runs. Having a look at how query expansion worked on this topic, we think it may be due to the term *required* being replaced interchangeably with *mandatory* and *recommended*, whose meanings are similar but their lexical nuances really make the difference in this specific context.
- On the other hand, we observe that topics such as 12 (“Should birth control pills be available over the counter?”), 20 (“Is drinking milk healthy for humans?”) and 44 (“Should election day be a national holiday?”) perform uniformly bad among all of the selected runs. We suggest it could be the result of a misunderstanding of the expression *over the counter*, which is possibly not very common, but also of the odd specificity of these questions. For example, in topic 20 the presence or absence of the term *milk* changes the meaning of the topic completely, making it difficult to retrieve actually relevant documents.

5.2.2. Touché 2021 Topics

We ran our pipeline on the Touché 2021 topics made available by the new edition of the challenge. We measured the nDCG@5 scores of the same parameters combinations we used for the 2020 challenge, observing the same trend: the scores align with the *swordsman baseline* but we could not improve on it due to a 0.14% difference. Table 2 shows the nDCG@5 scores with the 2021 topics. Nevertheless, we were able to improve our *lucene baseline* by 2% thanks to the quality re-ranking module.

The 2021 topics appear to be tougher when compared to the previous year’s edition, as shown by the plot of Figure 3: we achieved a score of zero on 4 topics, which are “Should suicide be a criminal offense?”, “Should the press be subsidized?”, “Should we imprison fewer people?”, and “Is psychology a science?”. Even the employment of query expansion was not able to retrieve any relevant argument with respect to these queries. Finally, the number of topics on which we achieved a full score has decreased.

The following list presents how a few queries were expanded:

- Topic 51 (“Do we need sex education in schools?”) has the word “sex” replaced by “sexual”, “education” replaced by “school”, “classes”, “instruction”, “teachers” and “lessons”, and “schools” replaced by “classrooms” and “education”. Even if the words replacements seem to make sense, we achieved an nDCG@5 score of 0 for this topic using query expansion.
- Topic 53 (“Should blood donations be financially compensated?”) is expanded to “Should blood donations be financially supported?”, which appears to be a good expansion, and “Should blood sacrifices be financially compensated?”, which is clearly (negatively) affected by the BERT training bias.
- Topic 87 (“Are gas prices too high?”) is expanded to “Are gas prices too bad?”, “Are oil prices too low”, etc. leading to a higher nDCG@5 score when compared to some runs that do not make use of query expansion. Topic 87 along with topic 92 are the only two topics

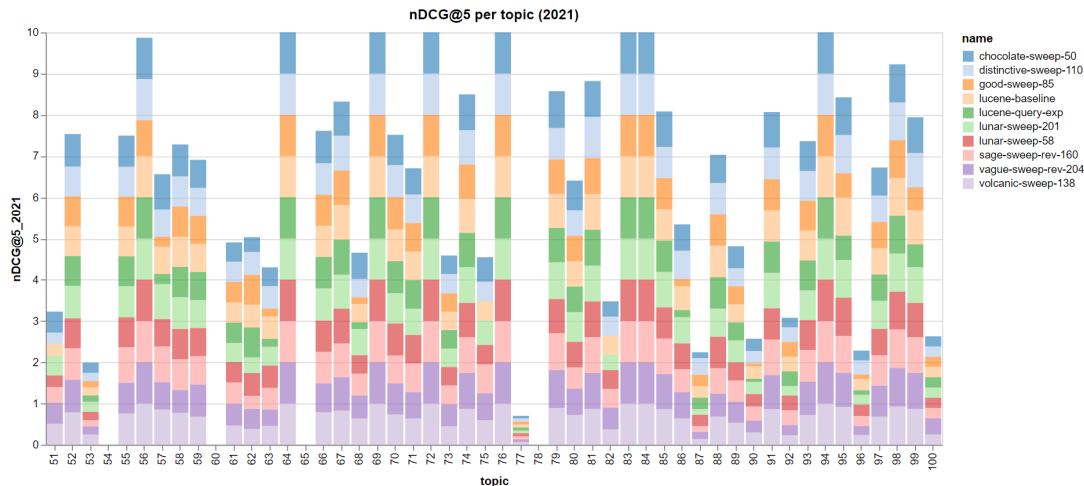


Figure 3: Bar plot of nDCG@5 scores for the 10 selected runs grouped by topic (Touché 2021 topics).

where query expansion was able to have a positive impact on the scores w.r.t. some of the runs not using it.

5.3. Statistical Significance Analysis

In order to determine if the differences between the 10 selected runs we reported in Table 2 are statistically significant, we performed two-way *ANalysis Of VAriance (ANOVA)* test on the Touché 2021 per-topic results. We employed the Matlab script *anova_test*, which is available in our repository on GitHub⁵ under the *statistical_analysis* section. As we can see in Figure 4, we can distinguish between two families of runs, which are those with and without query expansion (in red and blue respectively). We can also see that our *lucene baseline* cannot be considered significantly different from the other runs not applying query expansion, despite having a noticeably lower nDCG@5 score.

We also provide the ANOVA table (see Table 3), which confirms the previous observations: using the standard significance level $\alpha = 0.05$, the *p-value* associated with the runs is extremely low, meaning that the null hypothesis is rejected and so there is definitely a difference between runs of the two families. The relatively low value of *Mean Squares (MS)* associated with *Error* source means that most of the variance (represented by the horizontal *whiskers* in Figure 4) can be explained as the effect of changes in the parameters combinations of each run which in turn produce different behaviours when dealing with different topics.

6. Conclusions and Future Work

As noted in [26], the usage of state-of-the-art architectures for argument retrieval in the previous edition of the challenge only slightly improved the results obtained by the DirichletLM baseline

⁵<https://github.com/tommaso-green/yeagerists-clef-touche-2021>

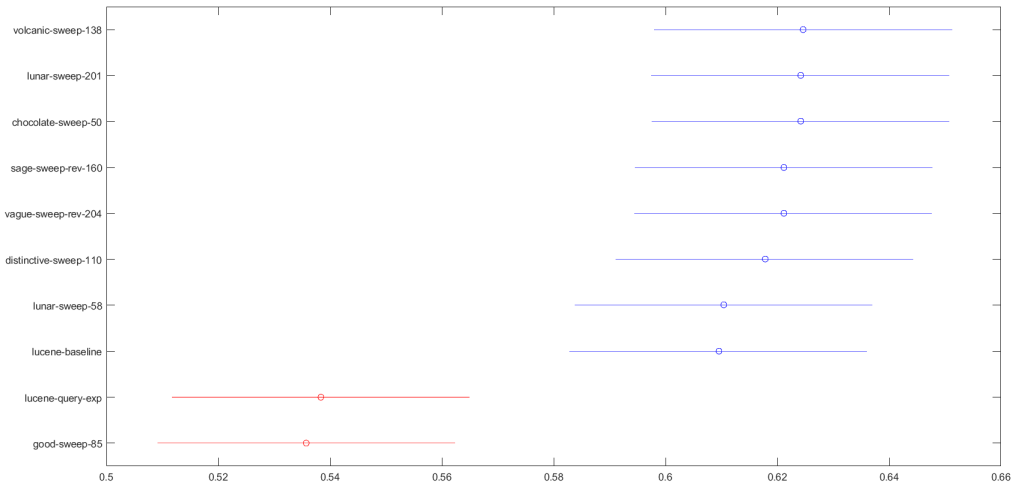


Figure 4: Results of two-way ANOVA significance test of the 10 selected runs on Touché 2021 topics.

Table 3

Two-way ANOVA table with significance level α set to 0.05

Source	SS	DoF	MS	F_{stat}	p -value (Prob > F_{stat})
Columns (Runs)	0.5516	9	0.0613	8.6504	5.4886e-12
Rows (Topics)	47.2754	49	0.9648	136.1783	3.7870e-235
Error	3.1244	441	0.0071	0.0	0.0
Total	50.9514	499	0.0	0.0	0.0

and our results seem to point in that direction, achieving a slight improvement over the baseline with an nDCG@5 score of 0.8279. Regarding the 2021 edition of the challenge, the new topics appear tougher, leading to a lower average nDCG@5 score. Nevertheless, the improvement of our approach is more significant when compared to our *lucene baseline*.

We believe that the difficulty of meaningfully improving the baseline score is related to the complexity of the task, but nevertheless, we deem that deep-learning-based approaches could provide interesting insights and improvements over traditional baselines.

First of all, it would be interesting to use transformer-based models also for the retrieval part: in particular, similarly to what was done last year in [27], it could be interesting to design a vector space IR model based on BERT, which could produce document embeddings fine-tuned for this task. For example, BERT models tailored to produce embeddings of long spans of text such as SBERT [28] could be deployed on this task. This integration would be orthogonal to our work and substitute the DirichletLM retrieval model.

As far as query expansion is concerned, while developing the approach described in section 3.2, we had the opportunity to try *Back-translation*, which consists of using a multilingual translation model to translate a sentence into a foreign language and back again into the original one. We tested it using the TextBlob library⁶, which for translation task relies on

⁶<https://textblob.readthedocs.io/en/dev/>

Google Translate API, so we could not use it for the final submission. Since it proved to be a really promising way of exploiting pre-trained transformer-based models to generate differently phrased sentences, while preserving most of the original meaning, we are eager to further investigate this technique.

Regarding argument quality prediction, it could be interesting to study the performance of our model in a fine-tuning approach instead of the adaptation one we used. In addition to this, it could be interesting to make the re-ranker predict the three types of quality (logical, dialectical and rhetorical) to see whether it would be possible for it to capture these three aspects independently. The re-ranking of results could then be performed based only on one of these qualities or on the overall quality score. Clearly, this adds a substantial layer of difficulty for the language model but could yield interesting details on how it internally represents these sophisticated features of language.

References

- [1] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2021: Argument Retrieval, in: D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), *Advances in Information Retrieval. 43rd European Conference on IR Research (ECIR 2021)*, volume 12036 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2021, pp. 574–582. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_67. doi:10.1007/978-3-030-72240-1_67.
- [2] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2021: Argument Retrieval, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), *Working Notes Papers of the CLEF 2021 Evaluation Labs*, CEUR Workshop Proceedings, 2021.
- [3] H. Wachsmuth, M. Potthast, K. Al-Khatib, Y. Ajjour, J. Puschmann, J. Qu, J. Dorsch, V. Morari, J. Bevendorff, B. Stein, Building an argument search engine for the web, in: *Proceedings of the 4th Workshop on Argument Mining*, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 49–59. URL: <https://www.aclweb.org/anthology/W17-5106>. doi:10.18653/v1/W17-5106.
- [4] Y. Ajjour, H. Wachsmuth, D. Kiesel, P. Riehmman, F. Fan, G. Castiglia, R. Adejoh, B. Fröhlich, B. Stein, Visualization of the topic space of argument search results in args.me, in: E. Blanco, W. Lu (Eds.), *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018) - System Demonstrations*, Association for Computational Linguistics, 2018, pp. 60–65. URL: <http://aclweb.org/anthology/D18-2011>.
- [5] R. Levy, B. Bogin, S. Gretz, R. Aharonov, N. Slonim, Towards an argumentative content search engine using weak supervision, in: *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2066–2081. URL: <https://www.aclweb.org/anthology/C18-1176>.
- [6] C. Stab, J. Daxenberger, C. Stahlhut, T. Miller, B. Schiller, C. Tauchmann, S. Eger, I. Gurevych, ArgumenText: Searching for arguments in heterogeneous sources, in: *Proceedings of the*

2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 21–25. URL: <https://www.aclweb.org/anthology/N18-5005>. doi:10.18653/v1/N18-5005.

- [7] H. K. Azad, A. Deepak, Query expansion techniques for information retrieval: A survey, *Information Processing & Management* 56 (2019) 1698–1735. URL: <https://www.sciencedirect.com/science/article/pii/S0306457318305466>. doi:<https://doi.org/10.1016/j.ipm.2019.05.009>.
- [8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface’s transformers: State-of-the-art natural language processing, 2020. arXiv:1910.03771.
- [9] D. Victor, Neuralqa: A usable library for question answering (contextual query expansion + bert) on large datasets, 2020.
- [10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.
- [11] W. Zhou, T. Ge, K. Xu, F. Wei, M. Zhou, BERT-based lexical substitution, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3368–3373. URL: <https://www.aclweb.org/anthology/P19-1328>. doi:10.18653/v1/P19-1328.
- [12] H. Wachsmuth, N. Naderi, Y. Hou, Y. Bilu, V. Prabhakaran, T. A. Thijm, G. Hirst, B. Stein, Computational argumentation quality assessment in natural language, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 176–187. URL: <https://www.aclweb.org/anthology/E17-1017>.
- [13] S. Gretz, R. Friedman, E. Cohen-Karlik, A. Toledo, D. Lahav, R. Aharonov, N. Slonim, A large-scale dataset for argument quality ranking: Construction and analysis., in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7805–7813.
- [14] Y. Ajjour, H. Wachsmuth, J. Kiesel, M. Potthast, M. Hagen, B. Stein, Data Acquisition for Argument Search: The args.me corpus, in: C. Benz Müller, H. Stuckenschmidt (Eds.), 42nd German Conference on Artificial Intelligence (KI 2019), Springer, Berlin Heidelberg New York, 2019, pp. 48–59. doi:10.1007/978-3-030-30179-8_4.
- [15] M. E. Peters, S. Ruder, N. A. Smith, To tune or not to tune? adapting pretrained representations to diverse tasks, in: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), 2019, pp. 7–14.
- [16] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.
- [17] L. Gienapp, B. Stein, M. Hagen, M. Potthast, Webis Argument Quality Corpus 2020 (Webis-ArgQuality-20), 2020. URL: <https://doi.org/10.5281/zenodo.3780049>. doi:10.5281/zenodo.3780049.
- [18] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

- [19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, in: ICLR 2020 : Eighth International Conference on Learning Representations, 2020.
- [21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [22] D. P. Kingma, J. L. Ba, Adam: A Method for Stochastic Optimization, in: ICLR 2015 : International Conference on Learning Representations 2015, 2015.
- [23] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: Advances in Neural Information Processing Systems, volume 30, 2017, pp. 971–980.
- [24] L. Biewald, Experiment tracking with weights and biases, 2020. URL: <https://www.wandb.com/>, software available from wandb.com.
- [25] M. Bundesmann, L. Christ, M. Richter, Creating an argument search engine for online debates, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_182.pdf.
- [26] A. Bondarenko, M. Fröbe, M. Beloucif, L. Gienapp, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2020: Argument Retrieval, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes Papers of the CLEF 2020 Evaluation Labs, volume 2696 of *CEUR Workshop Proceedings*, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [27] C. Akiki, M. Potthast, Exploring argument retrieval with transformers, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_241.pdf.
- [28] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3980–3990.