CLEARNESS: Coreference Resolution for Generating and Ranking Arguments Extracted from Debate Portals for Queries

Johannes Weidmann, Lorik Dumani and Ralf Schenkel

Trier University, Behringstraße 13, 54286 Trier, Germany

Abstract

Argumentation has always been used by humans to convince other people of certain viewpoints, e.g., to push through personal interests, or to resolve conflicts. The research field of computational argumentation deals with the extraction, analysis, retrieval, and generation of arguments in natural language texts. Modern argument search engines are able to generate appropriate arguments for controversial topics, very often based on posts taken from debate portals. However, an important issue is that posts in these portals are often quite long and incomprehensible. Apart from that, long posts in debate portals cannot be arguments by definition as not every text is argumentative.

In this paper, we dock on preliminary work about argument search engines and present CLEARNESS (**C**oreference reso**L**ution for g**E**nerating and r**A**nking a**R**gume**N**ts **E**xtracted from portal**S** for querie**S**), an approach to generate arguments in response to a query. The arguments we focus on consist of two essential elements: a claim, which is a point of view on a topic, and a premise, which provides the reasons or evidence backing up the claim. While previous works address issues like the generation of claims or the creation of abstract summaries of texts, we pursue a high-precision retrieval approach. We extract fine-grained premises from argumentative texts and modify these through coreference resolution to obtain an isolated text that is –although short– both coherent and completed.

We first build up a database by extracting arguments from the Args corpus of arguments from a number of popular debate portals. In these argumentative texts, we identify all coreferences and resolve them. Next, we examine classic and state-of-the-art approaches to rank arguments in response to a query. Additionally, we study the ranking behavior by utilizing query expansion. Lastly, we investigate the performance on relevance and coreference resolution.

Keywords

computational argumentation, coreference resolution, argument generation, argument retrieval

1. Introduction

Debates, be they verbal or non-verbal, have always been an integral part of human beings and continue to play a significant role in societies, cultures, and politics up to this day. They are essential to debate on a given topic or to make informed decisions after understanding them as they stimulate critical thinking and ease problem-solving. Additionally, new perspectives also encourage questioning one's own worldview and integrating new ideas into one's own opinion.

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany

s4joweid@uni-trier.de (J. Weidmann); dumani@uni-trier.de (L. Dumani); schenkel@uni-trier.de (R. Schenkel)

0009-0008-7095-4332 (J. Weidmann); 0000-0001-9567-1699 (L. Dumani); 0000-0001-5379-5191 (R. Schenkel)

CEUR Workshop Proceedings (CEUR-WS.org)

However, the prerequisite for participating in a debate is to research information on the topic of discussion. This is typically done through various media sources such as newspapers, websites, television, or social media.

Although nowadays there are almost limitless possibilities to inform and educate oneself, it is also becoming increasingly challenging to verify and filter relevant information. Social media has created a gigantic amount of information, thus most of it cannot be fact-checked. Fake news, the spread of manipulative false news, has become a global problem. For example, according to a statistic on fake news in the USA [1], during the Covid-19 pandemic, 80 % of people encountered fake news, but only 26 % were able to identify it as false. Another problem is that people are engaging with topics with less concentration. Research teams analyzed different media and examined how long a topic remained popular on Twitter, Reddit, and others. While in 2013, on average, a hashtag remained in the top 50 list for 17.5 hours, by 2016, it only stayed there for an average of 11.9 hours [2]. Interest in individual topics tends to decrease over time. Concurrently, the desire to constantly jump from one topic to another is increasing. The decline in attention span is exacerbated nowadays by short videos on YouTube, Instagram, or TikTok.

These findings are highly concerning and can potentially harm our culture of discourse in the future. Therefore, it is all the more necessary to support people in forming informed opinions during times of social media, accompanied by an influx of information and uncertainty about the truthfulness of facts. Therefore, we consider the field of *Computational Argumentation* (CA) to be extremely essential in assisting individuals in this process. CA is a subfield of NLP that, among others, deals with the extraction, analysis, and generation of arguments in natural language texts [3]. In particular, the findings of CA contribute to creating argument generation systems that support people in researching a topic or forming an opinion. *Argument search* involves gathering pertinent premises and claims related to a specific, often contentious topic. An *argument* [4] can be defined to consist of two components, a claim and a premise. The *claim* describes a controversial statement the arguer wants to persuade or dissuade the audience. A *premise* serves as evidence or clue to increase or decrease the acceptance of the claim. The polarity of a premise, i.e., whether it is supporting or opposing the claim, is called *stance*. An example of a claim could be "*Teachers should get tenure*", two supporting premises could be p_1 = "*Teacher tenure provides stability within schools*" and p_2 = "*It protects teachers' academic freedom*". The objective of an argument search engine is to furnish users with substantiated statements that aid in acquiring knowledge about their subject of interest and potentially facilitate their decision-making process [5].

An example of an argument search engine is Args [6, 7]. This platform provides the user with arguments on a topic divided in terms of their stance on a particular issue. Args and other platforms use posts from debate portals as arguments as their underlying dataset. As Args takes into account user-generated posts as arguments, the returned premises for user queries can become lengthy and linguistically unintelligible when considering individual sentences. Pronouns are often used, referring to previously mentioned objects in the sentence. If multiple objects exist, it can be difficult to determine which pronoun exactly refers to which object. Additionally, in long discussions, a comment may refer to a post that occurred much earlier, which may not be evident in the individual sentence. However, when considering the sentence isolated, the context might be missing for understanding. Due to the fact that in this work we consider standalone sentences as short premises, a challenge is that without additional

information, the meaning and sentence comprehension may be lost. As an example, imagine the two aforementioned premises p_1, p_2 would be written sequentially in one post but only the second sentence (p_2) would be retrieved as output isolated from p_1. Then, it is difficult to understand p_2 as it uses a pronoun ("*It*") that refers to a subject ("*Teacher tenure*") in p_1.

While existing works focus on topics such as motion-aware claim generation [8] or belief-based claim generation [9], we pursue a different goal in this study for generating premises. More precisely, given a query, we aim to generate short but relevant and coherent premises from existing valuable texts, employing a high-precision approach. To tackle the problem of current argument search engines with lengthy posts as premises, we seek to extract fine-grained premises from these posts using *coreference resolution*. A coreference is a concept that deals with the relationship between two or more expressions that refer to the same thing or person as shown in the above example. An expression can be a pronoun, a noun, or another type of word that refers to another noun in the text [10]. The goal is to extract short premises from texts and then resolve their coreferences to improve the retrieval with more understandable arguments when considering their sentences. For the remainder of the paper, we denote isolated sentences as premises. However, when we refer to longer premises, we call them posts. Additionally, we enhance the query (internal knowledge) by performing query expansion, which involves generating synonymous queries using ChatGPT.

2. Related Work

We now discuss various contemporary papers that have explored the field of argument generation. We specifically highlight diverse approaches and techniques, revealing the complexity and depth of existing work in this area. The purpose of this comprehensive literature review is to identify strengths and weaknesses in the current research landscape, while also establishing the originality and relevance of our paper. It underscores the evolution of the field and justifies the proposed study's exploration of argument generation with coreference resolution.

Alshomary et al. [9] focus on tailoring the generation of arguments according to the beliefs and convictions of the audience, which have not been considered in previous works. The study concludes that while there are limitations in modeling users' beliefs based on their stances, the results demonstrate the potential of encoding beliefs into argumentative texts. This lays the groundwork for future exploration of audience reach in argumentative discourse.

Al-Khatib et al. [11] explore how arguments can be generated and controlled using a knowledge graph. While previous research has already enhanced models like GPT-2 with knowledge, the use of an external knowledge graph is a new approach. Their findings demonstrate that their approach is capable of generating high-quality arguments by enriching the models with complex, interconnected knowledge.

Opitz et al. [12] consider new metrics for argument similarity that offer both high performance and interpretability. Previous approaches often lacked interpretable evidence or justifications for their evaluations, making it difficult to understand the features that determine argument similarity. The study suggests using Abstract Meaning Representation (AMR) graphs to represent arguments and demonstrates that new AMR graph metrics can provide explanations for argument similarity ratings. The AMR similarity statistics provided initial indications of what could

be considered a good conclusion, even without a reference comparison. They examined two hypotheses: (1) AMR semantic representation and graph metrics help in evaluating argument similarity, and (2) automatically derived conclusions can support or enhance the evaluation of argument similarity. Evidence was found for the former hypothesis but not for the latter.

Another approach is the use of argument generation frameworks. Hua et al. [13] present a framework called CANDELA for the automatic generation of counter-arguments, supported by a retrieval system and two decoders (text planning and content reflection). Unlike previous approaches, this approach is more precise as it considers language style, such as "*In theory, I agree with you*". CANDELA outperforms other methods like Seq2Seq, showing significantly better ROUGE, BLEU, and METEOR scores. In the evaluation conducted by human judges, who were asked to rate arguments on a scale of 1 (worst) to 5 (best) based on grammar, appropriateness, and content richness, CANDELA performed the best, delivering arguments with richer content and human-like responses.

Schiller et al. [14] introduce a language model called Arg-CTRL, which can be used for argument generation and has the ability to fine-tune argument-related aspects such as topic, stance, and aspect. The Arg-CTRL model is trained using a Common Crawl Dump with 331 million documents and a Reddit Dump with 2.5 trillion documents as the data source. Generated arguments are evaluated by human judges for grammatical correctness and persuasiveness. The evaluation also demonstrates that the arguments generated by Arg-CTRL can compete qualitatively with human arguments, as measured by WA-scores. The results show that arguments generated using this approach are generally authentic and of high argumentative and grammatical quality. Refining Arg-CTRL with data from Common Crawl leads to a higher quality of generated arguments compared to using user discussions from Reddit comments.

In the mentioned studies, new arguments are generated through the use of knowledge graphs, frameworks, or specialized argument language models. However, in this study, we utilize existing arguments from the text corpus. We apply coreference resolution to these arguments, meaning that if there are coreferences within a sentence, they will be resolved. If no coreferences are present, we retain the sentence in its original form. Thus, in the case of resolved coreference, a sentence is obtained that is structurally similar to the original, but with references replaced by the intended entities. Therefore, our approach does not generate new arguments but it makes them more visible and adapts the invaluable user-generated opinions as it only modifies existing arguments through coreference resolution in order to enhance the coherence and persuasiveness of the sentence.

Another method to generate new arguments is to enrich the dataset with external information. Yu et al. [15] summarize methods and techniques in a review on how to generate text using additional knowledge. A widely used approach for text generation is the use of text generation models. Text generation can be enhanced by internal knowledge derived solely from the query. For example, the query can be expanded by extracting and incorporating its topic, enabling the system to produce more accurate outputs that align with the query. Additionally, individual keywords from a predefined vocabulary that describe the query, in summary, can help prevent the generation of universal outputs that do not precisely match the query. In this work, we facilitate argument generation by utilizing an external dataset consisting of 5,933 argumentative texts (external knowledge) that have already been labeled with topics and stances.

3. Dataset

In this section, we describe our dataset, its source, and its transformation into a usable format.

As our main dataset, we use the Args corpus [16], a large open-source dataset that consists of 387,606 arguments (i.e. posts) extracted from online debate portals.[1] Debate portals are websites specifically designed to organize and regulate debates. Users can post their opinions and viewpoints on controversial topics, as well as agree or disagree with others. Overall, Args consists of 59,637 debates with 200,099 posts with a pro stance and 187,507 with a con stance. More precisely, in the Args corpus an argument is stored as a claim (yielded from the debate title) and premises (yielded from the posts to that debate). The data is available in JSON format. The average count of arguments per claim is 5.5 while 90 % of the claims have 1 to 10 arguments. In general, the debates are rather short with about 62 % of debates having 6-10 arguments. The average count of arguments per debate amounts to 6.5. In our work, we process the Args corpus by splitting the posts or premises into sentences.

In order to ensure a robust system at the end, we pick 30 out of 50 topic titles provided in the XML file as queries from the CLEF lab Touché 2022. Touché is an initiative focused on argument retrieval.[2] Given a controversial topic, the goal of this task was to rank arguments by relevance, argument quality, and stance. However, this is not part of this study as we only investigate the effect of coreference resolution on the ranking in comparison with the original sentences. Including other ranking measures could skew the findings. With regard to the relevance of the arguments, we make use of the Qrels ("Query relevance judgments"), which contain relevance assessments for a set of queries and documents and are also provided by the lab. For easier handling, we developed separate parsers for each dataset to merge them into one dataset.

4. Methods

Our main goals are to evaluate the effect of different methods for coreference resolution and query expansion. Our retrieval pipeline is illustrated in Figure 1. The first step of the process is to collect the data from the data sources, in our case the processed dataset Args (see Section 3).

Next, we use query expansion to enhance the queries. This is expected to increase the recall in the (relevant) results, meaning a higher proportion of relevant premises will be found without impacting the ranking negatively as much as possible as we pursue a high-precision recall. For query expansion, we do not employ conventional approaches such as document collection analysis or pseudo-relevance feedback, which involves utilizing user ratings on the returned results. Instead, we utilize a zero-shot approach by making use of ChatGPT to obtain five similar queries to the original query.[3]

After identifying the relevant premises, we proceed with the text modification by applying coreference resolution to get small but coherent premises. We apply the algorithm to each retrieved text and resolve all co-references. As a result, we obtain all texts from our database with

[1] The sources of Args are debatewise.org, idebate.org, debatepedia.org, and debate.org

[2] The task name is "Argument Retrieval for Controversial Questions": https://touche.webis.de/clef22/touche22-web/argument-retrieval-for-controversial-questions.html

[3] For reproducing the experiments with ChatGPT, the extended questions can be downloaded from the following website: https://basilika.uni-trier.de/nextcloud/s/nzoJfd9t9HRDTHN

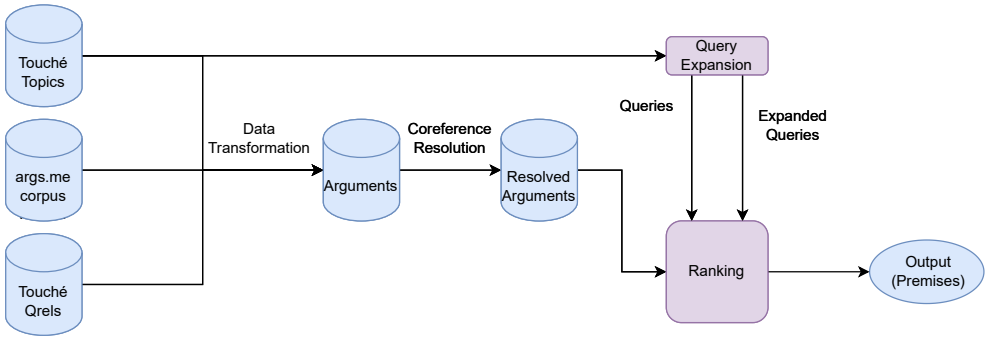

Figure 1: Pipeline for our argument search engine

resolved references. We then input the texts resolved through coreference resolution, along with the original and expanded queries, into our ranking system. The system provides us with relevant premises for each query –regardless of whether it is original or extended– on a controversial topic by sorting the premises based on scores between the query and premise, indicating their relevance, i.e., in the end, the user gets a fixed number of premises, sorted in descending order of estimated relevance. Within this section we compare different implementations for its components, deriving our final configuration based on these findings.

4.1. Coreference Resolution

We conducted preliminary tests with five different coreference libraries: neuralcoref, allennlp, coreferee, stanfordnlp, and fastcoref.[4]

To obtain the best possible result and identify as many coreferences as possible, we selected 10 argumentative texts with an average of 280 words and applied the mentioned coreference resolution algorithms to them. Then, we compared the texts and analyzed the number and quality of the identified coreferences. Each library provides us with a list of coreferences for a text, which consists of pairs of an object and a pronoun referring to the same entity in the real world. We used these coreferences as a reference point to evaluate the tools in terms of their results. The goal was to identify the best tool based on the quantity and quality of the identified references. Prior to running each tool on the texts, we manually defined a set of ground-truth coreferences known to exist in the texts. We have counted every coreference, even if it would not fit into the text when resolved later. Coreferences such as *we* and *us* are thus counted as coreferences because both parts refer to the same real entity. If a tool identifies a coreference that we have defined as true, it is considered a True Positive. Conversely, if the tool identifies a faulty coreference, it is considered a False Positive. False Negatives are those coreferences that the tool fails to detect, but they actually exist in the text. True Negatives are those that neither exist nor are identified by the tool. For each tool, we calculated the precision (the proportion of correctly identified coreferences by the tool out of the total coreferences detected by the tool), the recall (the proportion of correctly identified coreferences out of all

[4]https://github.com/huggingface/neuralcoref, https://github.com/allenai/allennlp, https://github.com/msg-systems/coreferee, https://stanfordnlp.github.io/CoreNLP/coref.html, https://pypi.org/project/fastcoref/

Table 1
Evaluation results of Coreference Resolution tools

Tool	Micro Avg. Precision	Macro Avg. Precision	Micro Avg. Recall	Macro Avg. Recall	Micro Avg. F1	Macro Avg. F1
Neuralcoref	0.6308	**0.7507**	0.3475	0.4320	0.4469	0.5351
AllenNLP	0.5760	0.5499	0.4308	0.4582	0.4854	0.4790
Coreferee	**0.6825**	0.6561	0.3739	0.3517	0.4735	0.4497
Fastcoref	0.6562	0.6706	**0.5887**	**0.6226**	**0.6162**	**0.6252**
StanfordNLP	0.4222	0.4345	0.3725	0.3324	0.3894	0.3612

ground-truth coreferences that exist in the text), and the F_1-score (harmonic mean of precision and recall). Table 1 shows the performances.

Precision, recall, and F1 scores serve as reliable benchmarks to determine which tool provides satisfactory results. However, during the compilation of all relevant coreferences, we adopted a detailed approach; that is, we included every coreference (all pairs of objects and pronouns that refer to the same entity) without considering whether the reference would be meaningful later. Pairs such as *'that'* and *'it'* are technically considered as coreferences, but add no value to the resolved text. This ultimately signifies that higher precision or recall does not causally indicate whether the tool delivers the best resolved text. Consequently, we meticulously examined all texts resolved by the tools, checking for sense, meaning, and sentence coherence. When evaluating the tools, we always kept our goal in mind. Since we follow a high-precision approach, we aim to obtain isolated premises in the final result. Therefore, a high precision is of great relevance to us, meaning that the coreferences should be correct and of high quality. Although Fastcoref has the best overall precision and recall scores, we hardly noticed any differences in actual results compared to neuralcoref. In the end, we decided to proceed with neuralcoref as the coreference library in our pipeline for several reasons. Neuralcoref uses a neural network model trained on extensive text data, which results in high accuracy in coreference resolution tasks. It has a clear and simple API that allows users to quickly and efficiently utilize the tool. This makes it possible to pass a text to the model and receive a list of coreferences in return. Neuralcoref has provided us with solid overall results that have convinced us in terms of both quality and quantity of coreferences. The tool was not overly greedy compared to other libraries like StanfordNLP which yielded too many unnecessary coreferences. Ultimately, the choice of library for the system is up to the individual. It depends on whether one prefers a high-precision or high-recall approach.

4.2. Query Expansion

For the query expansion, we utilize the ChatGPT API provided by OpenAI. As the prompt sent to the API endpoint, we select:

```
This is a query: {query}. Return 5 similar queries based on
    this query in a JSON List with the key 'similar_queries'.
```

ChatGPT followed these instructions and processed our prompt by returning 5 similar queries based on our original query in a JSON list. The generated queries either replace keywords with synonyms or rephrase the query in the same sense. For example, with the prompt and the query *"Is vaping with e-cigarettes safe?"*, we obtain the following 5 expanded queries: *"What are*

the health risks of vaping?", •*"Is vaping less harmful than smoking?"*, •*"What chemicals are in e-cigarette vapor?"*, •*"Can vaping lead to addiction?"*, •*"What is the long-term impact of vaping on health?"*.

4.3. Ranking of Premises

Now we have all argumentative texts with resolved coreferences and all queries together, allowing us to start matching queries with sentences. We now analyze different methods to find relevant premises for a query. We utilize five methods for this task: Jaccard, BM25, BERT, TF-IDF, and ChatGPT. For each query (in total, we use 30), the system finds –as usual in argument retrieval– the most similar claims to the query. We can then consider their premises and determine the relevant texts. Each method takes as input the query (or a list of queries for query expansion), all relevant texts aggregated into one text (which we split into sentences), and the original texts to determine if a coreference has been resolved. We considered the following methods:

Jaccard. For the Jaccard coefficient, we calculate the Jaccard score between the query and each premise (sentence) in the texts. The score is computed based on the 4-grams of the query and premise.

BM25. For BM25, we input the relevant texts and the query (as a list of tokens).[5]

BERT. W.r.t. BERT, we make use of the "sentence-transformers" python library. More precisely, we use the model "all-MiniLM-L6-v2 model" that maps texts to 384-dimensional embeddings. In our implementation, we use it for the query and each premise in the text corpus. Then, we compute the cosine similarity which also represents the score for each premise to the query.

TF-IDF. For TF-IDF, we employ the TfidfVectorizer module of the open-source "scikit-learn" python library.[6] It applies TF-IDF to transform the raw text into a matrix of TF-IDF features. The module's internal cosine similarity function calculates scores for all query-premise pairs.

ChatGPT. The ChatGPT approach differs from the other methods in that we do not compute a score. Instead, we ask the chatbot to provide the most relevant sentences from the corpus regarding the query. To do this, we send the following prompt to the ChatGPT API endpoint:

```
This is a text: {text}. Return arguments that you can infer
    from this text matching to this query: {query}. You should
    only give me arguments that can be inferred from the text.
```

The result does not correspond exactly to the sentences from the corpus but rather to arguments that ChatGPT can infer based on its prior knowledge. Therefore, ChatGPT is unsuitable as a method for solving this task, despite its good performance. This is mainly due to the non-deterministic behavior of ChatGPT since it tries to answer in natural language and seems to pursue the paradigm that a related answer is better than no answer. Thus, it does not always reply properly to questions. Hence, we received arguments in the form of summaries, inferences based on prior knowledge, or even completely new ones which do not occur in the original texts at all. However, we encountered an issue with this method as the API only allows a limited number of tokens of 4,096 as a prompt. To circumvent this, we divided the text into sections

[5]We use the rank-bm25 python library: https://pypi.org/project/rank-bm25/

[6]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

of 300 words each, ensuring the token limit was never reached.[7] For each section, we sent the aforementioned prompt to the endpoint and saved the output. To obtain a comprehensive result, we then sent the following prompt for the results of all sections:

```
Give the five best arguments from these arguments that fit the
    most to this query: {query}.
```

For each method and query, we take the best five arguments, i.e., for each of those with the highest scores. Each returned premise in the output (except for ChatGPT) is accompanied by the resolved coreferences, indicating whether a reference has been resolved (True) or not (False). W.r.t. the query expansion, we also output to which query the premise relates to. For the annotation process, we utilized a scale ranging from 0 to 2. A score of 0 implies that the premise provides no answer to the query and is thematically unrelated, 1 implies that the premise does not provide a precise answer but is thematically consistent with the query, and 2 suggests that the premise delivers an appropriate and thematically accurate response. Overall, we fed 30 queries into the system, applying query expansion to each. As a result, we had a total of 60 queries (30 standard queries + 30 expanded queries) input into the system. Note that the five expanded queries per query are viewed as one block, i.e., their premises are put into one list without duplicates. Then, we consider its n most relevant premises where the relevance is based on the score from the query to the premise. Since we tested five methods, each returning five premises, we had a total of 1,500 premises that needed to be annotated. This was accomplished manually. To evaluate the qualitative performance of each method, we calculated the Normalized Discounted Cumulative Gain (nDCG) score, the precision, and the mean reciprocal rank (MRR) for the normal query and the expanded queries as shown in Table 2.[8]

Table 2 reveals that ChatGPT delivers the best results in both disciplines. However, it was unable to provide the exact premises from the text and could not display resolved co-references. More precisely, it returned the five most relevant premises; rather, they are premises derived by ChatGPT based on the textual context. Therefore, each returned premise was relevant to the query because they were inferred from the entire context. As a result, ChatGPT exhibits consistent values of 1 for nDCG, precision, and MRR. Thus, we chose BERT as the module in our pipeline for our final system (i) because of the aforementioned issues and (ii) the premises returned are generally more flexible and not rigidly tied to the query in terms of identical words or structure. Note, that there are no significant deviations among the other methods that rely on a direct comparison between the query and premise in the text.

For query expansion, the results show that it delivers more relevant results for some methods but not for others. For instance, query expansion leads to slightly lower nDCG and precision values for Jaccard coefficient, BM25, and TF-IDF. Probably, since they are based on the bag-of-words models. However, for BERT, query expansion achieves higher nDCG and precision values. We strongly suspect that this is due to BERT's ability to "understand" the context of a sentence as its embeddings capture the context and not just individual terms as in the other

[7] This workaround successfully bypassed the token limit, but implicates a longer duration of the program since each request to the API takes about 10 seconds.

[8] We performed statistical significance calculations only for nDCG and MRR but not for Precision@{1,5} because these tests were performed only after acceptance and one hint from a reviewer.

Table 2
Evaluation results of query matching methods. Values with significant differences to ChatGPT are highlighted with a $*$. The precision is reported in two scenarios: one strict were we only consider high quality premises (score 2) and one lenient, which contains all premises with scores 1 or 2. The lower table shows the performance after utilizing Query Expansion (QE).

Method	(mean) nDCG@1	(mean) nDCG@5	(mean) precision@1 (strict)	(mean) precision@5 (strict)	(mean) precision@1 (lenient)	(mean) precision@5 (lenient)	(mean) Reciprocal Rank
Jaccard	0.73	0.91	0.53	0.69	0.9	0.95	0.71
BM25	0.80	0.91	0.7	0.63	0.9	0.89	0.81
BERT	0.73	0.92	0.53	0.72	0.93	0.96	0.73
TF-IDF	0.73	0.91	0.6	0.69	0.87	0.93	0.78
ChatGPT	**1.00***	**1.00***	**1.00**	**1.00**	**1.00**	**1.00**	**1.00***

Method	(mean) nDCG@1 (QE)	(mean) (nDCG@5) (QE)	(mean) precision@1 (strict) (QE)	(mean) precision@5 (strict) (QE)	(mean) precision@1 (lenient) (QE)	(mean) precision@5 (lenient) (QE)	(mean) Reciprocal Rank (QE)
Jaccard	0.68	0.88	0.59	0.63	0.83	0.91	0.71
BM25	0.67	0.89	0.43	0.61	0.87	0.89	0.66
BERT	0.78	0.92	0.6	0.72	0.97	0.97	0.77
TF-IDF	0.7	0.87	0.6	0.59	0.8	0.83	0.76
ChatGPT	**1.00***	**1.00***	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

methods. Therefore BERT can possibly find more relevant premises because of the reformulated queries. ChatGPT's performance could not be improved anyway. However, query expansion did not make it worse either. All nDCG precision, and MRR values remain at 1. We suppose this is because ChatGPT infers the premises through the queries. Since query expansion not only involves additional costs due to requests to ChatGPT, but also produced worse results for most methods except for BERT, where, however, the observed differences in relevance between a query and its expansion are not significant anyway, we did not use query expansion from the end-to-end evaluation in the next section.

5. Evaluation of Coreference Resolution

In this section, we examine the effect of coreference resolution in an end-to-end evaluation. First, we investigate the effect of coreference resolution on the regular output of our system. Second, we study how resolving the coreferences changes the understanding and structure of a premise.

5.1. Effect of Coreference Resolution on Argument Retrieval

We input the 30 queries on our dataset (see Section 3) and apply BERT (see Section 4) to find the 10 most relevant premises for each query, yielding 300 premises in total, 23 % (71 premises) of which have a resolved coreference. Although this is not a particularly high proportion of the total output, it shows that good coreference resolution is relevant for these premises.

W.r.t. the effect of coreference resolution on the rank of a single premise, we examined the equivalent original unresolved sentence for each resolved premise and checked what rank it actually holds. We found that on average, coreference resolution improves the rank of a premise by 16.7 positions. In the worst case, the rank of the resolved premise drops by four positions, and in the best case, it improves by 377 positions. The median of all rank differences is 0. Note

Table 3
Overall relevance of premises with resolved and unresolved texts, * = p < 0.05

	nDCG scores (mean)									
Corpus	**@1**	**@2**	**@3**	**@4**	**@5**	**@6**	**@7**	**@8**	**@9**	**@10**
unresolved	0.78	0.80	0.83	0.84	0.83	0.85	0.86	0.88	0.90	0.94
resolved	**0.87**	**0.87**	**0.89**	**0.89**	**0.89**	**0.90**	**0.90***	**0.91***	**0.93***	**0.96**

that whether the effect is positive or negative depends strongly on the sentence structure and the quality of the coreference resolution tool. For example, in the best case, a sentence like "*It will reduce crime, as showed in the real world*" is ranked higher after resolving the pronoun "*It*" to "*more gun laws*" as it perfectly fits the query. As a negative example, the original sentence "*I do not think that schools should require their students to wear a school uniform*" becomes "*I do not think that schools should require schools students to wear a school uniform*" by resolving the coreference *their* to *schools*, making the sentence sound less natural and introducing redundancy, resulting in a lower rank.

Now, we examine the overall relevance of the premises (i) with the texts containing resolved coreferences, and (ii) with the original texts. Thus, we run the system on both the original and the resolved corpus and extract the top 10 relevant premises from each version. We then annotate the relevance score for each result using the 0-1-2 scale, as done in the annotations. This allows us to assess the overall relevance using nDCG as shown in Table 3. The results show that coreference resolution increases the overall relevance of the returned premises. For each $n, 1 \leq n \leq 10$, the variant with the resolved corpus demonstrates higher values. A two-sided paired t-test with Bonferroni correction shows significant differences for $n \in \{7, 8, 9\}$.

5.2. Effect of Coreference Resolution on Isolated Premises

In the first part of the evaluation, we measured the impact of coreference resolution for the whole argument retrieval pipeline. In the second part of the evaluation, we now focus on the effect of coreference resolution on individual premises. We simulate the scenario in which all premises in the output contain resolved coreferences, retrieving with BERT the 10 most relevant premises that contain a resolved coreference for all 30 queries. In total, we obtained 300 premises to examine more closely. More precisely, we investigate two questions: (i) what is the effect of coreference resolution on the understanding of a premise, and (ii) what exactly the success of the coreference resolution depends on.

We employ a scale ranging from 0 to 2 for both investigations. A score of 0 indicates a resolved reference that was incorrectly identified. An example could be the resolution in the sentence "*We should be making it easier for these people to become citizens, not harder*" by replacing *these people* with *they*. A score of 1 suggests that the resolved reference does not provide additional assistance, but neither does it alter the original meaning. An example is the sentence "*Teacher tenure creates complacency because teachers know they are unlikely to lose their jobs*", where the coreference *their* could be replaced by *teachers*. A score of 2 implies that the resolved coreference substantially enhances the premise's value, even to the point of restoring its original meaning only through resolution. For instance, in the premise "*My case is against it, it should not be*

illegal, it should be legal" through the resolution of *it* by *abortion*.

Note that 24 % of the premises (Score 0) contain resolved coreferences that disrupt the structure and meaning of the original sentence. Manual inspection showed that this is mostly due to an error in the coreference resolution tool or finding a correct reference that does not make sense in the context of the sentence. 61 % of the premises do not improve the understanding of the sentence but also do not disrupt its structure or meaning. These are cases where a correct coreference is identified and resolved, but it does not provide any additional information for the user's comprehension because the pronoun alone is already sufficient to convey the meaning. Finally, in 15 % of all premises, coreference resolution significantly enhances the understanding of a premise. By resolving the coreferences, the premise regains its meaning when considered individually. These are mostly premises where the pronoun is used but the object is missing. Replacing the pronoun with the object brings the actual statement back to the surface.

This does not generally imply that coreference resolution is ineffective for modifying premises. The results are highly dependent on the sentence structure of the premise. In nearly all instances with scores 0 and 1, the coreference cluster is resolved within the same sentence, meaning the object and pronoun are present in the original premise. In the instances with a score of 2, the original premises nearly always lack the object and only the pronoun is included.

6. Conclusion and Future Work

We presented a pipeline for an argument search engine that returns fine-grained premises from long posts in debate portals using coreference resolution. We have centrally investigated the effect of coreference resolution on the overall output of the search engine and, individually, on the understanding of a premise. Additionally, we examined the effect of query expansion and which method is best suited to match a premise to a query. We consider neuralcoref the best tool for resolving coreferences and BERT the best method for finding a relevant premise to a query. The evaluation of the returned premises revealed that coreference resolution has a positive effect on both the general relevance and the understanding of the premises. More relevant premises are found for a query, which, when considered individually, tend to be more understandable than those without resolved coreference.

Note that in our evaluation which can be seen as a first step, about 24 % of the premises deteriorated rather than improved as a result of the resolution, while it had no effect on about 61 % and contributed to an improvement in only 15 %. This is mainly because resolving is not useful everywhere. Thus, in the future, we plan to check the sentences for suitability before resolving the coreferences. For example, we might try to suppress cases with coreference resolution within the same sentence. Further, we could evaluate the performance when coreference is only applied to premises that lack the object, and only the pronoun is included.

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the projects ReCAP and ReCAP-II, Grant Number 375342983 - 2018-2024, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

[1] A. Watson, Statistic on Fake News, https://www.statista.com/topics/3251/fake-news/topicOverview, 2023. Accessed: 2023-07-30.

[2] P. Lorenz-Spreen, B. Mønsted, P. Hövel, S. Lehmann, Accelerating dynamics of collective attention, Nature Communications 10 (2019). doi:10.1038/s41467-019-09311-w.

[3] A. Lauscher, H. Wachsmuth, I. Gurevych, G. Glavas, Scientia potentia est - on the role of knowledge in computational argumentation, CoRR abs/2107.00281 (2021). URL: https://arxiv.org/abs/2107.00281. arXiv:2107.00281.

[4] F. Macagno, D. Walton, C. Reed, Argumentation Schemes, 2018, pp. 517–574.

[5] E. Durmus, Towards understanding persuasion in computational argumentation, 2021.

[6] 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://aclanthology.org/W17-5106. doi:10.18653/v1/W17-5106.

[7] H. Wachsmuth, Args - Argument Search, https://www.args.me/index.html, 2023. Accessed: 2023-08-28.

[8] D. Suhartono, A. Gema, S. Winton, T. David, M. Fanany, A. Arymurthy, Sequence-to-sequence learning for motion-aware claim generation, International Journal of Computing 19 (2020) 620–628. doi:10.47839/ijc.19.4.1997.

[9] M. Alshomary, W. Chen, T. Gurcke, H. Wachsmuth, Belief-based generation of argumentative claims, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, Association for Computational Linguistics, 2021, pp. 224–233. URL: https://doi.org/10.18653/v1/2021.eacl-main.17. doi:10.18653/v1/2021.eacl-main.17.

[10] R. Sukthanker, S. Poria, E. Cambria, R. Thirunavukarasu, Anaphora and coreference resolution: A review, CoRR (2018). URL: http://arxiv.org/abs/1805.11824. arXiv:1805.11824.

[11] K. Al Khatib, L. Trautner, H. Wachsmuth, Y. Hou, B. Stein, Employing argumentation knowledge graphs for neural argument generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4744–4754. URL: https://aclanthology.org/2021.acl-long.366. doi:10.18653/v1/2021.acl-long.366.

[12] J. Opitz, P. Heinisch, P. Wiesenbach, P. Cimiano, A. Frank, Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation, in: Proceedings of the 8th Workshop on Argument Mining, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 24–35. URL: https://aclanthology.org/2021.argmining-1.3. doi:10.18653/v1/2021.argmining-1.3.

[13] X. Hua, Z. Hu, L. Wang, Argument generation with retrieval, planning, and realization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2661–2672. URL: https://aclanthology.org/P19-1255. doi:10.18653/v1/P19-1255.

[14] B. Schiller, J. Daxenberger, I. Gurevych, Aspect-controlled neural argument generation, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 380–396. URL: https://aclanthology.org/2021.naacl-main.34. doi:10.18653/v1/2021.naacl-main.34.

[15] W. Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H. Ji, M. Jiang, A survey of knowledge-enhanced text generation, CoRR abs/2010.04389 (2020). URL: https://arxiv.org/abs/2010.04389. arXiv:2010.04389.

[16] Y. Ajjour, H. Wachsmuth, J. Kiesel, M. Potthast, M. Hagen, B. Stein, Data Acquisition for Argument Search: The args.me Corpus, 2019, pp. 48–59. doi:10.1007/978-3-030-30179-8_4.