Leipzig University
Institute of Computer Science
Degree Programme Computer Science, B.Sc.

Axiomatic Re-ranking for Argument Search

Bachelor's Thesis

Marvin Vogel

1. Referee: Jun.-Prof. Dr. Martin Potthast

Submission date: September 14, 2023

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur under Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, 14. September 2023

. . . .
Marvin Vogel

Abstract

This thesis evaluates how axiomatic re-ranking can be used for argument retrieval. In particular, an initial ranking is taken, and the retrieval result is improved using four newly proposed axioms, namely QSenSim$_{mean}$, QSenSim$_{max}$, QArgSim$_{mean}$, and QArgSim$_{max}$. These axioms prefer a document if that document's sentences or argument units are more similar to the query and are implemented both using Sentence-BERT and Word2Vec. For evaluation, the documents of the args.me corpus are first ranked by DirichletLM for the Touché 2021 queries. Then, the top 5 of these documents for every query are re-ranked based on our new axioms. To determine whether an axiom achieves a significantly better result than the initial DirichletLM ranking, a t-test ($p < 0.05$, Bonferroni corrected) is done. We implement all axioms with Word2Vec and Sentence-BERT. Implemented with Sentence-BERT, all four axioms except for QSenSim$_{mean}$ achieve a significantly better result than the DirichletLM baseline. We also combine our new axioms with the axioms already implemented in ir_axioms. From the combinations that we test, QArgSim$_{max}$ + ArgUC + QTArg is the best axiom combination with an nDCG@5 of 0.673, which is also a significant improvement over the DirichletLM baseline. This shows that axiomatic re-ranking is a suitable retrieval method for finding good arguments.

Contents

Appendices

List of Tables

For tables displaying retrieval results (Chapter 4, Appendix B, Appendix C), a dagger† after an entry in the table indicates a statistically significant difference ($p < 0.05$, Bonferroni corrected) compared to the baseline highlighted in **bold**.

Chapter 1

Introduction

Motivation. People are often in situations where they have to make decisions based on various arguments speaking for or against a certain choice. Getting an overview of the different arguments concerning one topic is, however, difficult because these arguments are spread among a number of different sources like debate portals, scientific papers, or social media. Search engines specifically designed for argument retrieval solve this problem by providing a list of the best arguments for a given query. Finding good arguments, however, comes with some additional challenges that need to be taken into account. We can, for example, not just consider topic relevance as we would do for simple text retrieval but also need to take argument-specific features like the quality of an argument and its stance into account.

There are various ways to design a search engine for argument retrieval. One approach is to use axiomatic re-ranking, a technique used for standard text retrieval, and fine-tune it for argument retrieval. An axiom, in this context, describes a characteristic a document should fulfill, where a document is just unstructured text that, in the context of this thesis, might contain one or more arguments. Axioms usually take in two documents and then prefer the document that fulfills the described characteristic better. For example, Given a query and two documents of similar length, the Term Frequency Constraint 1 Axiom (TFC1) prefers the document that contains more query terms. It is also possible to combine these axioms by, for example, defining a new axiom that prefers the document preferred by the relative or absolute majority of a set of axioms. These axioms can be used for retrieval by using a two-step retrieval system. In the first step, all documents are ranked by an initial retrieval system. In the second step, the top k documents are re-ranked based on an axiom's preference. This method has the advantage that the initial retrieval system can be freely chosen. This means that axiomatic re-ranking can at least potentially improve any existing retrieval system. Other advantages are that existing retrieval systems can be adjusted for specific applications by using axioms designed for certain use cases and that retrieval systems using axiomatic re-ranking are easily explainable because the documents at the top of the ranking have the characteristics described by the axioms used. These advantages make axiomatic re-ranking a research field worth exploring.

Goals and Approach. This thesis aims to use the aforementioned advantage of axiomatic re-ranking and take an initial retrieval system designed for text retrieval and adjust it with axiomatic re-ranking for argument retrieval. More specifically, we want to evaluate if axioms can be used to retrieve good arguments and, if possible, to use an axiomatic approach to find good arguments. At first, we evaluate whether the already existing axioms proposed by other research are already capable of improving argument rankings significantly so we can see if these axioms are useful candidates for our argument retrieval system.

The main part of this thesis is about finding new axioms better suited for argument search. To achieve this, we change existing axioms to improve their retrieval results for argument search and propose four new axioms showing good argument retrieval capabilities. Two of these axioms compare the similarity of the sentences of documents and then prefer the document with a higher average similarity or the document with the most similar sentence. The other two axioms are similar to the aforementioned axioms except that they compare the similarity of a document's argument units with the query.

In the last part, we combine these newly proposed axioms with existing ones to improve retrieval results further. Combining axioms is possible by creating a new axiom that prefers the document that is preferred by the relative or absolute majority of the combined axioms.

Chapter 2

Related Work

The basis of argument retrieval is similar to any other retrieval task. Meaning that there is a user who is searching for specific information and is looking for documents relevant to this information in a corpus of documents. In particular, the user does this by putting a query into a retrieval system that returns a list of documents ranked by their relevance to the given query. This section will describe the specific characteristics of argument retrieval compared to general text retrieval that need to be considered when designing an argument retrieval system, as well as what methods and frameworks are used to implement and evaluate the argument retrieval techniques used in this thesis.

2.1 Argumentation

As described above, there are several specifics that need to be taken into account when the goal is to retrieve good arguments. General text retrieval systems usually rank the documents of their corpus by their relevancy to the given query. Applied to argument search, this approach, however, only takes into account how well an argument fits a given query and ignores whether an argument makes strong points and would be considered a strong argument. While the question “What is a good argument?” is a philosophical question discussed for millennia, every argument consists of similar parts. There is, however, not one singular approach to argumentation but a variety of considerably different co-existing approaches. A modern approach to argumentation was developed by van Eemeren and Grootendorst [2003], who do not see arguments as single units but as parts of a broader argumentation, where they define argumentation as follows:

Table 2.1: 15 quality dimension for assessing argument quality

Quality Dimension	Description
Cogency	Argument has (locally) acceptable, relevant, and sufficient premises.
Local acceptability	Premises worthy of being believed.
Local relevance	Premises support/attack conclusion.
Local sufficiency	Premises enough to draw conclusion.
Effectiveness	Argument persuades audience.
Credibility	Makes author worthy of credence.
Emotional appeal	Makes audience open to arguments.
Clarity	Avoids deviation from the issue, uses correct and unambiguous language.
Appropriateness	Language proportional to the issue, supports credibility and emotions
Arrangement	Argues in the right order.
Reasonableness	Argument is (globally) acceptable, relevant, and sufficient.
Global acceptability	Audience accepts use of argument.
Global relevance	Argument helps arrive at agreement.
Global sufficiency	Enough rebuttal of counterarguments.
Overall quality	Argumentation quality in total.

Argumentation *is a verbal, social, and rational activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint.*

By their definition, an argument (standpoint) is successful if all of the argument's propositions are accepted by all participants of the argumentation.

Wachsmuth et al. [2017b] describe arguments as a combination of a conclusion and one or multiple premises. Where a conclusion is a claim that can be accepted or rejected, and a premise is a factual statement supporting the claim. To assess the quality of an argument Wachsmuth et al. [2017a] looked at three different characteristics: The logical, rhetorical, and dialectical qualities of arguments. The logical quality of an argument describes the relation of the conclusion and the premises to each other, the rhetorical quality describes how convincing an argument is to an audience, and the dialectical quality describes the relations of an argument to other arguments of the same topic. To measure the quality of an argument, they propose 15 quality dimensions shown in Table 2.1. Arguments are then judged on a scale from 1 (low) to 3 (high) for each dimension. To assess the relevance of an argument, the two quality dimensions, Local relevance and Global relevance, can be used. Local relevance describes how well the premises of an argument support its conclusion, and global relevance describes how much an argument contributes to coming to an agreement. Another approach to measure argument relevance was used by Potthast et al. [2019]. They simply use the information retrieval notion of relevance, which means that retrieved information is relevant if it satisfies the needs of the user. The arguments are then judged on a scale from 1 (low) to 4 (high) based on how well they fit this criterion.

2.2 Re-ranking using Axiomatic Preferences

One approach for text retrieval is axiomatic re-ranking. As described by Hagen et al. [2016], the basic idea behind this approach is to first rank the documents of corpus to a given query with some initial retrieval system. Then, take the top k results of that ranking and re-rank these k documents based on the preference of an axiom. An obvious advantage of this approach is that any retrieval system can be used as the initial retrieval. In the context of axiomatic re-ranking Hagen et al. [2016] describes an axiom as a triple:

$$Axiom = (precondition, filter, conclusion), \tag{2.1}$$

where the precondition needs to be fulfilled for the axiom to have a preference, the filter is the actual preference function of the axiom, and the conclusion is the preference that the axiom has for two given documents.

This concept can be easily understood with an example, so let's look at the PROX2 axiom, which says: Prefer documents with earlier query term occurrences. If we express this axiom as a triple, we get:
Let $first(t, d)$ be the first occurrence of a term t in document d
Given a query $q = \{t\}$ with one term and two documents d_1 and d_2.

$$\begin{array}{lcl} \text{precondition} & = & t \in d_1 \wedge t \in d_2 \\ \text{filter} & = & first(t, d_1) < first(t, d_2) \\ \text{conclusion} & = & d_1 >_{PROX2} d_2 \end{array}$$

2.3 Bayesian Smoothing with Dirichlet Priors

The Dirichlet model is a widely used model for document retrieval. As described by Zhai and Lafferty [2001], the basic idea of this approach is to interpret every document as a language model and then rank the documents based on the probability that they generate the given query. This approach is simplified by Zhai and Lafferty [2001] to the probability of whether a single word generates a given document assuming:

$$p(q|d) = \prod_i p(q_i|d), \tag{2.2}$$

meaning that the product of the probability of all words of the query is equal to the probability of the whole query. The easiest way to calculate the probability of a word generating a document would now be to use the maximum likelihood estimate. Which means simply counting how often the word occurs in the document and dividing that by the number of all words in the document. The problem with this method is that it underrepresents words that are not contained in the document at all. To solve this problem, Zhai and Lafferty [2001] reduces the probability of words contained in the document and raises the probability of words not contained in the document. Words not contained in the document are assigned the probability that they are generating the entire corpus multiplied by a parameter controlling how much influence these words should have. Because the probability of words not contained in the document is increased, the probability of words contained in the document needs to be reduced. This can be done with various smoothing methods. The one used in this thesis is the Bayesian smoothing with Dirichlet priors method, where the probability of a word contained in the document is calculated as:

$$p(w|d) = \frac{c(w,d) + \mu p(w|C)}{\sum_w c(w,d) + \mu}, \tag{2.3}$$

where $c(w,d)$ is the number of occurrences of the word in the document, $p(w|C)$ is the probability that the word is generating the corpus, and μ is the smoothing parameter.

2.4 Cosine Similarity

Cosine similarity is a widely used metric for the similarity of two vectors. As described by Li and Han [2013], the idea behind this metric is that the cosine of the angle at which two vectors cross each other is used to measure how similar these vectors are. Given two N-dimensional non-zero vectors $\overrightarrow{v}$ and $\overrightarrow{w}$ the cosine similarity of these vectors is calculated as:

$$\cos(\overrightarrow{v}, \overrightarrow{w}) = \frac{\vec{v} \cdot \vec{w}}{||\vec{v}|| \cdot ||\vec{w}||} = \frac{\sum\limits_{i=1}^{N} v_i \cdot w_i}{\sqrt{\sum\limits_{i=1}^{N} v_i^2} \cdot \sqrt{\sum\limits_{i=1}^{N} w_i^2}}. \tag{2.4}$$

The cosine similarity takes a value between -1, meaning the vectors are exact opposites and therefore cross at an angle of 180°, and 1, meaning the vectors are the exact same, therefore crossing at an angle of 0°. A cosine similarity of 0 indicates that the vectors cross at a right angle.

2.5 Text Embeddings

Text embeddings describe a natural language processing technique where text (usually words) is represented as a vector. These vectors encode the characteristics like the meaning or semantic relations of the text they represent.

Word2Vec. The framework we use for word embeddings is Word2Vec, which is a technique for natural language processing proposed by Mikolov et al. [2013]. It uses a neural network to learn associations of words and represents every word it learned as a vector. This makes it possible to use word2vec to calculate the similarity of two words by calculating the cosine similarity between the two vectors of these words. In this thesis, the “word2vec-google-news-300” model is used.

Word2Vec is primarily based on two models: the Continuous Bag-of-Words Model (CBOW) and the Continuous Skip-gram Mode (Skip-gram). CBOW describes a model that predicts a word from its surrounding context words, while Skip-gram describes the opposite, a model that predicts the context words from a given target word. Both of these models are used to adjust the weights of the hidden layer of a neural network, which is trained on a large corpus of text. After the training, the weights are used as vectors representing the words of the corpus. So, the result is a model that contains one vector for every word, where words that are semantically similar in the training corpus have assigned vectors that are close together in the vector space.

Sentence-BERT. For sentence embeddings, we use Sentence-BERT, a technique for sentence embeddings proposed by Reimers and Gurevych [2019]. It is based on a BERT, a language model used for natural language processing [Devlin et al., 2018]. BERT can be fine-tuned to compute the similarity between two sentences, but as both sentences have to be put into the model for this calculation, BERT is unsuitable for large numbers of sentence comparisons

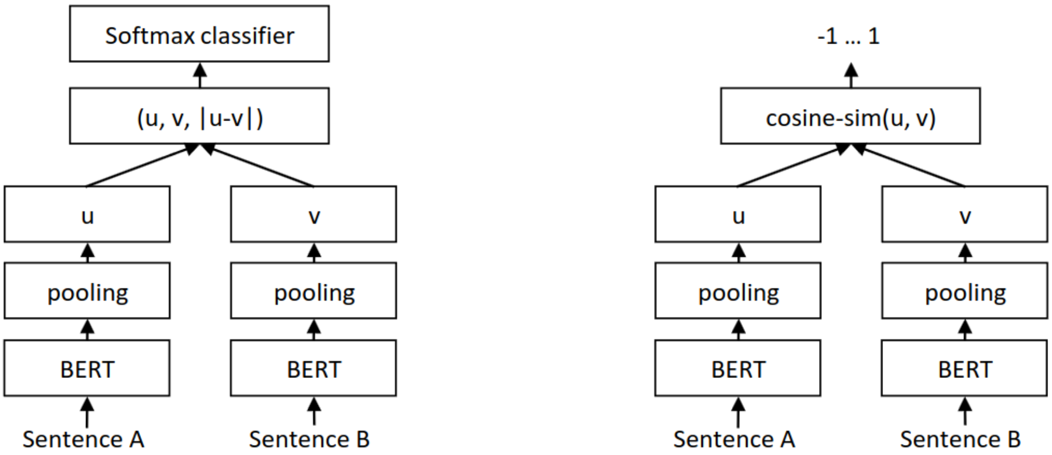


Figure 2.1: Architecture of Sentence-BERT [Reimers and Gurevych, 2019].

because of the computational overhead. Sentence-BERT solves this problem by converting a sentence into a vector in a way that the vector captures the semantic characteristics of that sentence. This makes it then possible to calculate the similarity between two sentences using, for example, cosine similarity.

To train Sentence-BERT, a Siamese network is used. This describes a type of neural network where two models with the same architecture and tied weights are trained at the same time. In the case of Sentence-BERT, two BART models are used. Then, a pooling method is added to the output of the BERT models to obtain fixed-size embeddings. By default, Sentence-BERT calculates the mean of all output vectors, but using the CLS-Token (a special token in the output of BERT containing sentence-level information) and calculating the max-over-time of all output vectors are pooling methods that are also possible. The outputs of the two pooling operations for the two BERT models are then compared depending on the dataset. If the dataset classifies the relation of two sentences into categories, a Softmax classifier is used, which calculates a probability for each relation category based on the output vectors of the two BERT models. The model then predicts the relation between the two sentences to be the category with the highest probability. If the dataset contains a numeric value for the similarity of the sentences, the cosine similarity of the output vectors is calculated. With this architecture, Sentence-BERT is then trained to generate similar vectors for similar sentences. An illustration of both versions of the Sentence-BERT training architecture can be seen in Figure 2.1.

2.6 Normalized Discounted Cumulative Gain

The normalized discounted cumulative gain is a widely used metric for measuring the performance of retrieval systems. As described by McSherry and Najork [2008], it is based on the discounted cumulative gain, which is calculated as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{judgement} - 1}{log_2(1+i)}. \tag{2.5}$$

where DCG@k means the discounted cumulative gain for the first five documents. It has to be noted that for using the DCG, all documents have to be given a judgment. This DCG is then normalized by dividing it by the ideal DCG (IDCG), which is equal to the DCG for a perfect ranking of all documents. This means that the nDCG is calculated as follows:

$$nDCG@k = \frac{DCG@k}{IDCG@k}. \tag{2.6}$$

2.7 Learning to Rank using LambdaMART

LambdaMART is a learning-to-rank algorithm. Learning to rank describes a supervised machine learning technique that aims to create the optimal order of a list of items. As proposed by Burges [2010], LambdaMART works in two steps. In the first step, the documents are ranked by their judgments, creating the perfect ordering, and the nDCG@5 is calculated (Note that any measure could be used here; this thesis, however, uses the nDCG@5). Then, each pair of documents is swapped, and the nDCG@5 is calculated again for every swap. The differences between the ideal nDCG@5 and the swapped nDCG@5 are summed up for every document (these sums are the lambdas used for LambdaMART). These lambdas now have the characteristic that highly relevant documents have big lambdas, while documents with low relevance have small lambdas.

In the second step of the algorithm, a regression tree is trained to predict the lambda of a document based on the previously chosen features of this document. If we now take a new dataset of documents and let the regression tree predict a lambda for every document, we can rank the documents in the dataset based on the lambdas. LambdaMart also makes it possible to analyze how much influence each feature used for the ranking had on the ranking position of each document by providing the feature importance for these features.

2.8 Statistical Testing

To determine whether our results are significantly better than the baseline, we use a paired t-test, which is a statistical test used for comparing the averages of two dependent samples. Its purpose is to find out whether the difference between the means arises from random change or because there is a significant difference between the samples. As described by Kaptein and van den Heuvel [2022], the basic idea behind this test is to define two hypotheses, the null hypothesis assumed to be true and the alternative hypothesis, which will be accepted if the null hypothesis is proven wrong. To determine if the alternative hypothesis should be accepted, we first calculate the difference between each pair of the sample. Now, instead of doing a two-sample test, a single-sample test can be done, where we test if the mean of all differences is equal to 0 to calculate the t-value as follows:

$$t_n = \frac{\bar{D}}{s_D/\sqrt{n}}, \tag{2.7}$$

where $\bar{D}$ is the average of the differences, s_D is the standard deviation of the differences, and n is the sample size (In the case of this thesis, the number of queries, which are tested per retrieval system). We then calculate the p-value as follows:

$$p = 2 \cdot (1 - F_t(|t_n|)), \tag{2.8}$$

where F_t is the t-distribution function with $n-1$ degrees of freedom and t_n is the observed value. If this p-value is below a certain threshold α ($\alpha = 0.05$ is used in this thesis), the null hypothesis can be rejected, and the alternative hypothesis is accepted, meaning that there is a significant difference between the two samples.

Because many comparisons are made in this thesis, this would, however, lead to the problem that the p-value would be smaller than α sometimes just by random chance. To solve this problem, the Bonferroni correction is used. As described by Ranstam [2016], the Bonferroni correction is simply calculated as:

$$\alpha' = \frac{\alpha}{k}, \tag{2.9}$$

where α' is the corrected threshold and k is the number of comparisons done at once. In the context of this thesis, the number of comparisons is the number of axioms or axiom combinations compared at once.

Chapter 3

Argument Retrieval using Axiomatic Re-ranking

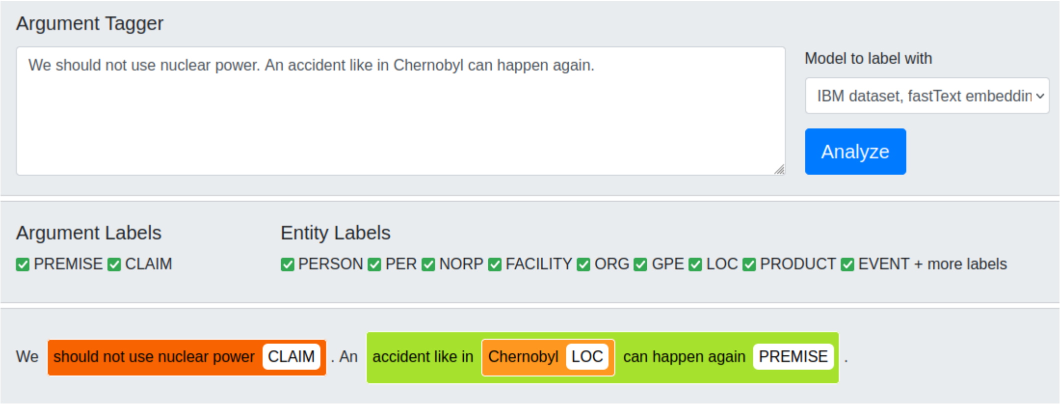
This chapter will describe the different methods we use to retrieve good arguments. Firstly, explanations of the used frameworks are given, followed by a description of retrieval ideas unsuited for argument retrieval. After that, the functionality of four newly proposed axioms is described. At the end, these new axioms are combined with the in ir_axioms implemented axioms to improve retrieval results further.

3.1 Used Frameworks

To test the different approaches of axiomatic re-ranking, various frameworks are used in this thesis. This section provides an overview of these frameworks and explains which functionalities they provide and how they are used to improve the retrieval of arguments.

ir_axioms is a Python framework designed by Bondarenko et al. [2022] and intended for retrieving documents using axiomatic re-ranking. It implements most of the already existing axioms and provides the ability to easily define new axioms as well as the possibility to re-rank existing rankings. Being well integrated with PyTerrier and Pyserini ir_axioms also makes it easy to analyze and compare the results of newly created rankings; for the purpose of this thesis, the integration with PyTerrier is used.

In ir_axioms, each axiom subclass has a preference function that takes the query and two documents as inputs and returns a number greater than 0 when the axiom prefers the first document, a number smaller than 0 when the axiom prefers the second document, and exactly 0 when the axiom does not



Argument Tagger

We should not use nuclear power. An accident like in Chernobyl can happen again.

Model to label with: IBM dataset, fastText embeddin

Argument Labels

☑ PREMISE ☑ CLAIM

Entity Labels

☑ PERSON ☑ PER ☑ NORP ☑ FACILITY ☑ ORG ☑ GPE ☑ LOC ☑ PRODUCT ☑ EVENT + more labels

We should not use nuclear power CLAIM . An accident like in Chernobyl LOC can happen again PREMISE .

Figure 3.1: Analyzing an argument with TARGER.

have a preference. The already implemented axioms return 1, -1, or 0 based on their preference. New Axioms can be created in two ways. The first is to concatenate already existing axioms with mathematical or logical operators, which combines the results of the preference functions of these axioms. Axioms can also be combined in a VoteAxiom, which prefers a document if a certain threshold of axioms prefers that document. The second way is to define a new axiom subclass, which makes it possible to define axioms with their own custom preferences function.

To change a ranking based on axiom preferences, ir_axioms provides the KwikSortReranker class. Firstly, an initial ranking is needed, of which we will take the top k documents. These top k documents are then re-ranked using the KwikSort algorithm, which orders the documents like the Quicksort algorithm using the axiom preferences as the ordering relation.

TARGER is a Python framework made by Chernodub et al. [2019]. Its purpose is to find arguments in texts. It provides the ability to tag argumentative units of a text with Dependency, FastText, or Glove embeddings trained on either the Essays or WebDiscourse data set. For the purpose of this thesis, the FastText embedding trained on the WebDiscourse data set is used because it achieves the best results. Words are tagged based on whether they are part of a claim, a premise, or part of neither. Through the SpaCy entity-tagger, TARGER also makes it possible to find the mentions of named entities (like locations, dates, or events) in a text.

In the example shown in Figure 3.1, the text “We should not use nuclear power. An accident like in Chernobyl can happen again” is analyzed on the TARGER web page.[1] The model used in this example is the FastText embed-

[1] `https://demo.webis.de/targer`

ding trained on the IBM data set as the one trained on the WebDiscourse data set usually requires longer texts to tag claims and premises and was therefore not suited for a short example like this. TARGER identifies "should not use nuclear power" as a claim and "accident like in Chernobyl can happen again" as a premise supporting that claim. So, we can conclude that the argument is arguing against using nuclear power because of the danger of accidents, citing Chernobyl as an example. Here, the city of Chernobyl is also correctly identified as a location (LOC) by the SpaCy tagger used on the TARGER web page.

3.2 Query Sentence Similarity Axioms

The basic approach that we use to define new axioms that would prefer documents containing good arguments is to take an already existing axiom and change it slightly in the hope of improving its retrieval results. An axiom that looks like a promising candidate for this approach is the Semantic Term Matching Constraint 1 (STMC1) axiom proposed by Fang and Zhai [2006]. This axiom prefers the document whose terms are more similar to the query term and was invented to solve the problem that axioms for axiomatic reranking were solely based on the exact matching of terms. What this axiom does is to calculate the similarity between each pair of document terms and query terms for both documents and then prefer the document with the higher average similarity of the terms. So, the basic idea of this axiom is to compare similarities on the term level.

A possible adjustment for this axiom would now be not to compare the similarity of terms but to compare the similarity of whole sentences. So, we propose a new axiom defined as follows:

Axiom 1. *Given a query and two documents. Prefer the document whose sentences are more similar to the query.*

Axiom 1 shows the Query-Sentence-Similalarity axiom using the mean similarity (**QSenSim**$_{mean}$).

As seen in Algorithm 3.1, QSenSim$_{mean}$ calculates the similarity between every combination of a sentence from the document and the query for both documents and then prefers the document with the higher average similarity of the sentences and the query.

The similarities are calculated with two different embeddings: Word2Vec and Sentence-BERT. For the implementation with Word2Vec, we assume that the vector of a sentence can be expressed as the average of the vectors of

Algorithm 3.1: Mean Query-Sentence-Similarity Axiom

```
input : A Query $q$, and 2 Documents $d1$ and $d2$
output: The Preference of the Axiom
similarities = [] ;
for term $\in$ Terms($d1$) do
    similarities += Similarity(term, $q$);
end
doc1_similarity = Mean (similarities);
similarities = [] ;
for term $\in$ Terms($d2$) do
    similarities += Similarity(term, $q$);
end
doc2_similarity = Mean (similarities);
if doc1_similarity > doc2_similarity then return 1;
else if doc1_similarity < doc2_similarity then return -1;
else return 0;
```

the words of this sentence. If a word in the document is not contained in the Word2Vec model, we ignore this word; if all words in a sentence are not contained in the Word2Vec model, we ignore this sentence; and if all words in a document are not contained in the Word2Vec model, we assume that the axiom has no preference when this document is compared with any other document. So we first calculate the vector for each word of the sentence, then build the average of these vectors and do this for every sentence in a document. The second embedding is Sentence-BERT, where the vector for each sentence is calculated directly by the model. Then, the similarity between the sentence vectors and the query vector is calculated using cosine similarity for both models.

Another idea that is worth exploring is to not prefer the document with the higher average similarity between the query and the document's sentences but to prefer the document that has the most similar sentence to the query. So, we propose another axiom defined as follows:

Axiom 2. *Given a query and two documents. Prefer the document that has the sentence that is most similar to the query.*

Axiom 2 shows the Query-Sentence-Similalarity axiom using the maximum similarity ($\mathbf{QSenSim}_{max}$).

QSenSim$_{mean}$ is implemented as shown in Listing 3.1 with the difference that in lines two and six, the max of the similarities is taken instead of the mean. This axiom calculates the similarity between each sentence of the document and the query for both documents and then prefers the document with the higher maximum similarity of the sentences and the query.

3.3 Query Argument-Unit similarity Axioms

While the two new axioms proposed in the previous section are promising candidates for achieving a significantly better retrieval result, they are still general axioms that can be used for every text retrieval task and are not specifically designed for argument retrieval. If these axioms could be adjusted to retrieve good arguments instead of just text specifically, the retrieval results could be improved further. One way to specify these axioms for argument search is to not look for the similarity between every sentence and the query but to filter the sentences of a document for argumentative units and only look at the similarity between a document's argument units and the query. So, we propose two new axioms defined as follows:

Axiom 3. *Given a query and two documents. Prefer the document whose argumentative units are more similar to the query.*

Axiom 4. *Given a query and two documents. Prefer the document that has the argumentative unit that is the most similar to the query.*

Axiom 3 shows the Query-ArgumentUnit-Similalarity axiom using the mean similarity (**QArgSim**$_{mean}$).
Axiom 4 shows the Query-ArgumentUnit-Similalarity axiom using the max similarity (**QArgSim**$_{max}$).

Algorithm 3.2 shows the implementation of QArgSim$_{mean}$. QArgSim$_{max}$ is implemented in the same way, with the difference that in lines two and six, the max of the similarities is taken instead of the mean. These axioms filter the sentences of a document for argumentative units and calculate the similarity between each combination of argumentative units of the document and the query for both documents and then prefer the document with the higher average or maximum similarity of the argumentative units and the query. The argumentative units are filtered using TARGER. Each word in the sentence is tagged by TARGER based on whether it is a claim, a premise, or neither.

Algorithm 3.2: Mean Query-ArgumentUnit-Similarity Axiom

```
input  : A Query $q$, and 2 Documents $d1$ and $d2$
output: The Preference of the Axiom
similarities = [] ;
for argument_unit $\in$ Argument_Units($d1$) do
    similarities += Similarity(argument_unit, $q$);
end
doc1_similarity = Mean (similarities);
similarities = [] ;
for argument_unit $\in$ Argument_Units($d2$) do
    similarities += Similarity(argument_unit, $q$);
end
doc2_similarity = Mean (similarities);
if doc1_similarity > doc2_similarity then return 1;
else if doc1_similarity < doc2_similarity then return -1;
else return 0;
```

3.4 Combining Axioms to Further Improve Retrieval Results

There are already several axioms that have been proposed in previous works. Combining these axioms with the new axioms from this work is a promising idea to improve retrieval results further. To find axioms that are suitable candidates for this, we use a LambdaMART-Ranker. As features for the ranker, we use the axiom preferences, so the percentages of times an axiom's preference is greater than 0, smaller than 0, or equal to 0 for every axiom implemented in ir_axioms. We then train the ranker to improve the nDCG for the first five documents. Then, we sum up the importance of the three features of each axiom and rank them by their added-up feature importance. The idea behind this approach is that axioms that have a big influence on the improvement of the nDCG are axioms that perform well. Therefore, axioms with high feature importance should have a big influence on improving the retrieval.

As there are only four axioms specifically designed for argument retrieval already implemented in ir_axioms, we also evaluate a second approach to axiom combination, which is to simply combine every possible combination of these four axioms with our four new axioms.

To combine axioms, we simply add up the preferences of a set of axioms. This creates a new axiom that prefers the document preferred by a relative majority of the axioms in the aforementioned set. If both documents are preferred by exactly half of the axioms, then the sum of their preferences will be 0, so the new sum axiom has no preference.

Chapter 4

Evaluation

To evaluate the retrieval performance of our newly proposed axioms, we test them on the corpus of the args.me search engine. For this test, we first rank the arguments in the args.me corpus based on 50 topics from the Argument Retrieval for Controversial Questions task of the Webis Touche shared task 2021 using the dirichletLM retrieval model. From this ranking, we take the first five arguments for each topic and re-rank them based on the preferences of our four new axioms. Additionally, we evaluate the performance of the already existing axioms, compare them to our newly proposed axioms, and combine these axioms to improve the retrieval results further. To measure the quality of the retrieval rankings, we use the normalized discounted cumulative gain for the top 5 arguments.

4.1 Data

The args.me Corpus. The different retrieval approaches are evaluated on the Corpus of the args.me argument search engine developed by Ajjour et al. [2019]. It consists of almost 400,000 arguments collected from the four debate portals Debatewise, IDebate.org, Debatepedia, and Debate.org, as well as from discussions of the Canadian parliament. Most of the arguments (over 300,000) are from Debate.org, and they were crawled in the middle of 2019 using heuristics specifically designed for each debate portal.

Each argument in the corpus consists of the premises, which are the actual argument, the conclusion of the argument, its stance, and some additional metadata.

Touché at CLEF 2021 Topics. The Touchê task is a yearly event organized by the Webis group. In 2021, the event included the task “Argument Retrieval for Controversial Questions," whose goal was to "retrieve relevant and high-quality argumentative texts from the args.me corpus" [Bondarenko et al., 2021]. The retrieval systems of the participating teams were evaluated on 50 different queries. An example of a query and its top 3 arguments ranked by DirichletLM, as well as some additional statistics for the args.me corpus, can be seen in Appendix A. For each team, the top five arguments for each query were judged based on their relevance, where "0" means an argument is not relevant to the given query, "1" means the argument is relevant, "2" means the argument is highly relevant and "-2" means the argument is spam.

Due to slight differences in the implementations of the DirichletLM ranking system used in the Touché task and the DirichletLM ranking system used in this thesis, the initial rankings of two queries contain in total 5 unjudged documents (out of 250 documents in total). To solve this problem, we create the index so that it only includes documents judged for at least one query. After removing two additional documents from the index that were ranked in the top 5 by DirichletLM but were not judged for that query, we have an initial ranking where the top 5 documents for every query are all judged.

We also look at the results axioms can achieve on the Touché topics of 2020. For the evaluation, we filter the index in the same way because 97 documents were not judged and additionally remove the four topics "Is vaping with e-cigarettes safe", "Should performance-enhancing drugs be accepted in sports", "Is a two-state solution an acceptable solution to the Israeli-Palestinian conflict", and "Should euthanasia or physician-assisted suicide be legal" because most of the highly ranked arguments for these topics were not judged even with an index that only contains judged documents.

4.2 Evaluation of Existing Axioms

Over the past decades, a variety of axioms have been proposed, many of which are implemented in ir_axioms. First, we want to look at the results these axioms achieve in retrieving arguments on the Touché 2021 dataset. Note that the axioms REG, ANTI-REG, ASPECT-REG, STMC1, and STMC2 calculate semantic similarities between words and that there are two different implementations used for this in ir_axioms. Axioms whose names end with "-f" indicate that this axiom was implemented using FastText embeddings, while the ones without "-f" are implemented with WordNet synsets. Both versions of these axioms work exactly the same in all other regards.

Tables 4.1 and 4.2 show that none of the axioms implemented in ir_axioms are significantly better than the DirichletLM baseline. This is a somewhat disappointing result as it shows that none of the axioms (not even those specifi-

Table 4.1: Axioms implemented in ir_axioms evaluated for argument relevance. The DirichletLM baseline is highlighted in bold. (1)

Axiom	nDCG@5
LEN-DIV	0.647
RS-TF	0.647
ANTI-REG-f	0.645
DIV	0.644
QTArg	0.643
LB1	0.643
TFC1	0.640
REG	0.639
REG-f	0.639
M-AND	0.639
ArgUC	0.638
PROX2	0.638
RS-BM25	0.638
RS-PL2	0.637
AND	0.637
RS-TF-IDF	0.636
QTPArg	0.636
ANTI-REG	0.635
TF-LNC	0.635
LNC1	0.634
STMC2	0.634
STMC1-f	0.634
DirichletLM	**0.633**

Table 4.2: Axioms implemented in ir_axioms evaluated for argument relevance. The DirichletLM baseline is highlighted in bold. (2)

Axiom	nDCG@5
DirichletLM	**0.633**
RS-QL	0.633
TFC3	0.633
M-TDC	0.633
LEN-M-TDC	0.633
ASPECT-REG-f	0.633
ASPECT-REG	0.633
aSL	0.633
LEN-M-AND	0.633
LEN-AND	0.633
PROX3	0.632
PROX1	0.630
STMC1	0.630
STMC2-f	0.628
PROX5	0.623
PROX4	0.623

cally designed for argument search) deliver good results for argument retrieval. A reason for this might be that these axioms do not describe the characteristics of good arguments well enough. Also surprising is that the best argumentative axiom (QTArg) is only in fifth place, while LEN-DIV is the best axiom. There seems to be no specific reason for this, so this axiom might be on top by chance and not because it prefers good arguments. The same applies to PROX4, which is the axiom with the worst nDCG@5.

4.3 Evaluation of Newly Proposed Axioms

We now want to evaluate the effectiveness of our new axioms implemented with Word2Vec and Sentence-BERT embeddings. As shown in Table 4.3, all versions of the Query-Sentence-Similarity axiom improve the nDCG@5 compared to the baseline. Still, only the axiom using Sentence-BERT and the maximum similarity can improve the nDCG@5 significantly. In contrast, both versions of the mean similarity and the maximum similarity axioms implemented with Word2Vec do not achieve a significant improvement. Looking at the two embeddings separately, it also stands out that the axioms using the Sentence-BERT embedding are better than the ones using the Word2Vec

Table 4.3: QSenSim axioms evaluated for argument relevance. The DirichletLM baseline is highlighted in bold.

Axiom	nDCG@5
QSenSim$_{max}$ (Sentence-BERT)	0.669†
QSenSim$_{mean}$ (Sentence-BERT)	0.658
QSenSim$_{mean}$ (Word2Vev)	0.638
QSenSim$_{max}$ (Word2Vev)	0.635
DirichletLM	**0.633**

Table 4.4: QArgSim axioms evaluated for argument relevance. The DirichletLM baseline is highlighted in bold.

Axiom	nDCG@5
QArgSim$_{max}$ (Sentence-BERT)	0.672†
QArgSim$_{mean}$ (Sentence-BERT)	0.661†
QArgSim$_{max}$ (Word2Vev)	0.650
QArgSim$_{mean}$ (Word2Vev)	0.646
DirichletLM	**0.633**

embedding. Another notable observation is that using the Sentence-BERT embedding the axiom using the maximum similarity achieves a better nDCG@5 than the axiom using the mean similarity. At the same time, it is the other way around with the axioms using the Word2Vec embedding.

Looking at Table 4.4, we can see that filtering the document content for argument units improves the nDCG@5 for all four axiom versions compared to the baseline as well as compared to their respective sentence similarity axioms. As with the Query-Sentence-Similarity-Axiom, the Query-ArgumentUnit-Similarity axiom also delivers better results using Sentence-BERT, with the differences that now the axioms using the maximum similarity are better both with the Sentence-BERT and Word2Vec embedding. This time, both Axioms implemented with Sentence-BERT improve the retrieval result significantly compared to the baseline, while the axioms implemented with Word2Vec do not improve the nDCG@5 significantly. The bad results of the axioms implemented with Word2Vec compared to the axioms implemented with Sentence-BERT might indicate that using the average of all word vectors of a sentence as that sentence's vector does not capture the meaning of that sentence well. To express a sentence as a vector a deeper understanding of sentence structures might be needed, which is provided by Sentence-BERT.

4.4 Combining Argumentative Axioms

All four of our newly proposed axioms using Sentence-BERT are better than all axioms already implemented in ir_axioms. The best of those being LEN-DIV with an nDCG@5 of 0.647. We now want to combine our new axioms with the numerous other axioms already implemented in ir_axioms. Because using Sentence-BERT as the embedding delivers better results than using Word2Vec, we will only evaluate our new axioms using Sentence-BERT from here on.

There are already four axioms that were proposed by Bondarenko et al. [2018] and Bondarenko et al. [2019] specifically designed for argument search, namely ArgUC, QTArg, QTPArg, and aSL. As these axioms are promising combination candidates for improving the retrieval results, we first evaluate the retrieval effectiveness of all possible combinations between each of our new axioms and the argumentative axioms implemented in ir_axioms.

Firstly, we want to evaluate the results of the combinations with the two Query-Sentence-Similarity axioms. Looking at Table 4.5, we can see that for both the QSenSim$_{mean}$ and the QSenSim$_{max}$ axiom, all possible combinations except for one decrease the nDCG@5. For the QSenSim$_{mean}$ axiom, the sum of QSenSim$_{mean}$, ArgUC, and QTArg improves the retrieval result, although not significantly. For the QSenSim$_{max}$ axiom, the sum of QSenSim$_{max}$ and aSL improves the retrieval result but also not significantly.

The retrieval results of the Query-ArgumentUnit-Similarity axioms in Table 4.6 are similar. For the QArgSim$_{mean}$ axiom, the best combination is again the sum of QArgSim$_{mean}$, ArgUC, and QTArg, but there are now also two other combinations that improve the retrieval result. These are the sum of QArgSim$_{mean}$, ArgUC, QTArg, and aSL, as well as the sum of QArgSim$_{mean}$, ArgUC, and QTPArg. All three improvements are not statistically significant. For the QArgSim$_{max}$ axiom, there is still only one combination that improves the retrieval result, which is the sum of QArgSim$_{max}$, ArgUC, and QTArg, while the sum of QArgSim$_{max}$ and aSL now results in a smaller nDCG@5. The improvement of QArgSim$_{max}$+ArgUC+QTArg is not statistically significant.

To check the results obtained by combining axioms, we also evaluate our axiom combinations on the Touché 2020 dataset. The results can be seen in Appendix C. As the axioms with the best results are now completely different, we can conclude that the combinations that delivered promising results on the Touché 2021 dataset did this by random change and not because they prefer better arguments.

We also evaluate these axiom combinations mentioned above using the VoteAxiom class provided by ir_axioms. The results can be seen in Appendix B as there were no notable differences to summing up the axioms' preferences.

Table 4.5: Combinations of QSenSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance. The QSenSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{mean}$+ArgUC+QTArg	0.659
QSenSim$_{mean}$	**0.658**
QSenSim$_{mean}$+ArgUC+QTArg+aSL	0.654
QSenSim$_{mean}$+aSL	0.653
QSenSim$_{mean}$+ArgUC+QTArg+QTPArg+aSL	0.651
QSenSim$_{mean}$+QTArg+aSL	0.651
QSenSim$_{mean}$+ArgUC+QTPArg	0.650
QSenSim$_{mean}$+QTArg+QTPArg	0.649
QSenSim$_{mean}$+QTArg	0.649
QSenSim$_{mean}$+ArgUC	0.648
QSenSim$_{mean}$+ArgUC+aSL	0.648
QSenSim$_{mean}$+QTPArg	0.646
QSenSim$_{mean}$+QTArg+QTPArg+aSL	0.646
QSenSim$_{mean}$+QTPArg+aSL	0.646
QSenSim$_{mean}$+ArgUC+QTPArg+aSL	0.645
QSenSim$_{mean}$+ArgUC+QTArg+QTPArg	0.645
DirichletLM	0.633
QSenSim$_{max}$+aSL	0.671
QSenSim$_{max}$	**0.669**
QSenSim$_{max}$+ArgUC+QTPArg	0.668
QSenSim$_{max}$+ArgUC+QTArg	0.668
QSenSim$_{max}$+ArgUC+QTPArg+aSL	0.668
QSenSim$_{max}$+ArgUC+QTArg+aSL	0.667
QSenSim$_{max}$+ArgUC	0.665
QSenSim$_{max}$+QTArg+aSL	0.664
QSenSim$_{max}$+QTPArg+aSL	0.664
QSenSim$_{max}$+QTArg	0.664
QSenSim$_{max}$+QTArg+QTPArg	0.663
QSenSim$_{max}$+ArgUC+QTArg+QTPArg+aSL	0.663
QSenSim$_{max}$+QTArg+QTPArg+aSL	0.662
QSenSim$_{max}$+ArgUC+QTArg+QTPArg	0.659
QSenSim$_{max}$+QTPArg	0.659
QSenSim$_{max}$+ArgUC+aSL	0.659
DirichletLM	0.633†

Table 4.6: Combinations of QArgSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance. The QArgSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QArgSim_{mean}+ArgUC+QTArg	0.668
QArgSim_{mean}+ArgUC+QTArg+aSL	0.664
QArgSim_{mean}+ArgUC+QTPArg	0.661
$\textbf{QArgSim}_{mean}$	**0.661**
QArgSim_{mean}+ArgUC+QTArg+QTPArg+aSL	0.659
QArgSim_{mean}+ArgUC+QTPArg+aSL	0.659
QArgSim_{mean}+aSL	0.659
QArgSim_{mean}+QTArg+aSL	0.659
QArgSim_{mean}+ArgUC+QTArg+QTPArg	0.658
QArgSim_{mean}+ArgUC+aSL	0.658
QArgSim_{mean}+QTArg+QTPArg+aSL	0.657
QArgSim_{mean}+ArgUC	0.656
QArgSim_{mean}+QTArg+QTPArg	0.655
QArgSim_{mean}+QTArg	0.654
QArgSim_{mean}+QTPArg+aSL	0.654
QArgSim_{mean}+QTPArg	0.647
DirichletLM	0.633
QArgSim_{max}+ArgUC+QTArg	0.673
$\textbf{QArgSim}_{max}$	**0.672**
QArgSim_{max}+ArgUC+QTArg+aSL	0.670
QArgSim_{max}+aSL	0.669
QArgSim_{max}+ArgUC+QTPArg	0.669
QArgSim_{max}+QTArg+aSL	0.668
QArgSim_{max}+ArgUC+QTPArg+aSL	0.664
QArgSim_{max}+ArgUC+QTArg+QTPArg+aSL	0.664
QArgSim_{max}+QTArg	0.664
QArgSim_{max}+ArgUC	0.663
QArgSim_{max}+QTArg+QTPArg+aSL	0.663
QArgSim_{max}+ArgUC+QTArg+QTPArg	0.662
QArgSim_{max}+QTArg+QTPArg	0.661
QArgSim_{max}+ArgUC+aSL	0.661
QArgSim_{max}+QTPArg+aSL	0.659
QArgSim_{max}+QTPArg	0.657
DirichletLM	$0.633^{\dagger}$

4.5 Finding Promising Axiom Combinations

Because there are not just these four axioms implemented in ir_axioms but also 33 other ones, it is an obvious idea to combine these other axioms with our newly proposed similarity axioms. Because a total of 37 axioms gives us over 137.000.000.000 possible axiom combinations for each of our four similarity axioms, testing every single combination is computationally unfeasible. To solve this problem, we try to find axioms that are promising candidates for improving the retrieval result when combined with our similarity axioms.

Our approach is to train a LambdaMART-ranker on the queries of the Touché 2021 dataset over 1000 iterations to improve the nDCG@5. We use the first 45 topics as our training dataset and the last five as our test dataset (Note that due to the LGBMRanker from LightGBM requiring all document judgments to be greater or equal to 0, we set all -2 judgments to 0). As features, we use the percentage of a given axiom being greater than 0 (meaning it prefers the first document), equal to 0 (meaning it has no preference), and smaller than 0 (meaning that it prefers the second document) for the first five documents of every query. This gives us 111 features in total. After the LambdaMART ranker is trained, we take the feature importance of each of our 111 features and sum up the three feature importances for every axiom. In Table 4.7, we can see all axioms ranked by their summed-up feature importances. Some axioms like PROX2, Prox 4, and TFC1 have a high influence on improving the nDCG@5, while seven axioms (TFC3, TF-LNC, M-TDC, LNC1, LEN-M-TDC, ASPECT-REG-f, and ASPECT-REG) have no influence at all.

We now define nine new axioms, A to J, which will consist of the sums of the axioms with the highest feature importance. We also add up all axioms implemented in ir_axioms with our new ones as a reference value. These new axioms are:

A := The axiom with the highest feature importance
B := Sum of the two axioms with the highest feature importance
C := Sum of the three axioms with the highest feature importance
D := Sum of the five axioms with the highest feature importance
E := Sum of the ten axioms with the highest feature importance
F := Sum of the 20 axioms with the highest feature importance
G := Sum of the axioms with a feature importance greater than zero
H := Sum of the axioms with the highest feature importance in their category[1]
I := Sum of all axioms

Looking at Table 4.8, we see that only a combination with axiom A (the

[1] Categories as in Appendix D

Table 4.7: Feature Importances of Axioms implemented in ir_axioms for improving the nDCG@5 using LambdaMART.

	Axiom	Feature Importance
1	PROX2	30.705
2	PROX4	22.794
3	TFC1	20.326
4	ANTI-REG-f	18.979
5	STMC1	18.814
6	QTArg	18.787
7	STMC1-f	17.761
8	PROX1	17.668
9	REG-f	17.097
10	DIV	15.097
11	RS-TF	14.118
12	LB1	13.973
13	STMC2-f	12.920
14	RS-PL2	11.734
15	ArgUC	10.790
16	LEN-AND	10.081
17	RS-BM25	9.942
18	RS-QL	9.446
19	REG	7.762
20	STMC2	7.155
21	LEN-DIV	5.577
22	aSL	4.334
23	LEN-M-AND	3.918
24	RS-TF-IDF	3.858
25	QTPArg	3.490
26	PROX5	2.227
27	M-AND	1.673
28	AND	1.148
29	PROX3	1.122
30	ANTI-REG	0.320
31	TFC3	0
32	TF-LNC	0
33	M-TDC	0
34	LNC1	0
35	LEN-M-TDC	0
36	ASPECT-REG-f	0
37	ASPECT-REG	0

Table 4.8: Combinations of QSenSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance. The QSenSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{mean}$ + A	0.661
QSenSim$_{mean}$	**0.658**
QSenSim$_{mean}$ + C	0.657
QSenSim$_{mean}$ + F	0.657
QSenSim$_{mean}$ + B	0.656
QSenSim$_{mean}$ + H	0.654
QSenSim$_{mean}$ + G	0.652
QSenSim$_{mean}$ + I	0.652
QSenSim$_{mean}$ + D	0.651
QSenSim$_{mean}$ + E	0.647
DirichletLM	0.633
QSenSim$_{max}$ + A	0.671
QSenSim$_{max}$	**0.669**
QSenSim$_{max}$ + B	0.665
QSenSim$_{max}$ + C	0.660
QSenSim$_{max}$ + H	0.659
QSenSim$_{max}$ + D	0.657
QSenSim$_{max}$ + F	0.656
QSenSim$_{max}$ + I	0.653
QSenSim$_{max}$ + G	0.652
QSenSim$_{max}$ + E	0.648
DirichletLM	0.633†

Table 4.9: Combinations of QArgSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance. The QArgSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QArgSim$_{mean}$	**0.661**
QArgSim$_{mean}$ + A	0.660
QArgSim$_{mean}$ + H	0.659
QArgSim$_{mean}$ + B	0.658
QArgSim$_{mean}$ + F	0.656
QArgSim$_{mean}$ + C	0.656
QArgSim$_{mean}$ + D	0.655
QArgSim$_{mean}$ + I	0.651
QArgSim$_{mean}$ + G	0.651
QArgSim$_{mean}$ + E	0.650
DirichletLM	$0.633^{\dagger}$
QArgSim$_{max}$	**0.672**
QArgSim$_{max}$ + A	0.670
QArgSim$_{max}$ + B	0.661
QArgSim$_{max}$ + C	0.658
QArgSim$_{max}$ + H	0.658
QArgSim$_{max}$ + D	0.657
QArgSim$_{max}$ + E	0.655
QArgSim$_{max}$ + F	0.654
QArgSim$_{max}$ + I	0.651
QArgSim$_{max}$ + G	0.651
DirichletLM	$0.633^{\dagger}$

Table 4.10: STMC1 axiom implemented using different embedding and evaluated for argument relevance. The DirichletLM baseline is highlighted in bold.

Axiom	nDCG@5
FastText	0.634
DirichletLM	**0.633**
Sentence-BERT	0.631
WordNet	0.630
Word2Vec	0.626

axiom only consisting of PROX2) can improve the nDCG@5 compared to both Query-Sentence-Similarity (the Mean and the Max) axioms. As this improvement is not significantly better, we, therefore, have to assume this improvement does not appear on other datasets. Table 4.9 shows that for the Query-ArgumentUnit-Similarity similarity axioms, no improvement is possible using the axioms defined above.

4.6 Effectiveness of STMC1 using Different Embeddings

Because we use different embeddings in our new axioms than the ones used in the STMC1 axioms implemented in ir_axioms, we need to make sure that the improvement of the nDCG@5 does not just occur because we use a better embedding. To evaluate the retrieval results of Word2Vec and Sentence-BERT for term similarity instead of sentence similarity, we implement the STMC1 axiom with these embeddings like in Listing 4.1. We then compare them to the STMC1 axioms using FastText and WordNet embeddings, which are already implemented in ir_axioms in the same way. So, the only difference between all four axioms is the exact implementation of the similarity function used in lines 3 and 8.

As shown in Table 4.10, the STMC1 axioms using Word2Vec and Sentence-BERT achieve about as good of a nDCG@5 as the STMC1 axioms using FastText embeddings and WordNet synsets. In fact, the implementation using Sentence-BERT is worse than the implementation using FastText embedding and the baseline, while the implementation using Word2Vec is the worst of the four. We can, therefore, conclude that the improved ndCG@5 of our new Axioms described in Chapter 3 results from the change made to the axiom and not from the use of a different embedding. This means that axioms comparing the similarity of the sentences or argumentative units of a document to the query are better suited for argument search than axioms comparing the similarity of the terms of documents and the query.

Algorithm 4.1: Preference Function of STMC1

```
input  : A Query $q$, and 2 Documents $d1$ and $d2$
output: The Preference of the Axiom

similarities = [] ;
for term1 ∈ Terms(d1) do
    for term2 ∈ Terms(q) do
        similarities += Similarity(term1, term2);
    end
end
doc1_similarity = Mean (similarities);

similarities = [] ;
for term1 ∈ Terms(d2) do
    for term2 ∈ Terms(q) do
        similarities += Similarity(term1, term2);
    end
end
doc2_similarity = Mean (similarities);

if doc1_similarity > doc2_similarity then return 1;
else if doc1_similarity < doc2_similarity then return -1;
else return 0;
```

4.7 Influence of Axioms on Argument Quality

Until now, we only looked at how axiomatic re-ranking influences the relevance of arguments, but because the Touché 2021 dataset also contains judgments for the quality of arguments, we also want to evaluate our new axioms for argument quality.

Looking at Table 4.11, we can see that all four of our new axioms improve the argument quality, although none of them improves the quality significantly. Another difference is that the QSenSim axioms have a better nDCG@5 than the QArgSim axioms. This could, however, just be a coincidence as the nDCG@5 difference between the axioms is small. The QSenSim axioms still deliver better results than the best axiom implemented in ir_axioms (which is REG-f with an nDCG@5 of 0.821), while the QArgSim axioms have a smaller nDCG@5, but the differences are minimal. Because a way to significantly

Table 4.11: QSenSim and QArgSim axioms evaluated for Argument Quality. The DirichletLM baseline is highlighted in bold.

Axiom	nDCG@5
QSenSim$_{mean}$	0.823
QSenSim$_{max}$	0.822
QArgSim$_{max}$	0.820
QArgSim$_{mean}$	0.818
DirichletLM	**0.808**

improve the argument quality with our new axioms is not obvious, and the improvement of the quality is not the focus of this work, we do not explore this idea further.

Chapter 5

Conclusion and Future Work

This chapter provides a summary of the previous chapters. In particular, it concludes whether it is possible to use axioms for argument retrieval and explains the level of improvement axioms are able to achieve. It also describes how well the axioms proposed in Chapter 3 perform in the experiments done in Chapter 4. The second part gives an outlook on possible future work on axiomatic re-ranking in the context of argument search.

5.1 Conclusion

In conclusion, we can say that axiomatic re-ranking is indeed usable for argument retrieval with good results. The axioms proposed in Chapter 3 are able to achieve a significantly better retrieval result than the DirichletLM baseline and can even be further improved by combining them with other axioms. As an evaluation for the newly proposed axioms, we can conclude that the axioms using Sentence-BERT to calculate the similarity, that use the maximum similarity, and that filter a document's text for arguments units deliver better results than the ones using Word2Vec, the mean similarity, and don't filter for argumentative units. Combining these new axioms with the other axioms implemented in ir_axioms can further improve retrieval results. Still, the axioms have to be carefully chosen and need to be weighted in case of long additions, as in most cases, axiom combinations actually worsen the retrieval result. This might be caused by giving axioms with a lower nDCG@5 than the new axioms too much influence on the axiom preference in simple addition. The best axiom combination we found is QArgSim$_{max}$ + ArgUC + QTArg with an nDCG@5 of 0.673, showing that axiom combinations can potentially improve the retrieval result.

5.2 Future Work

We demonstrate that axiomatic re-ranking can improve the results of retrieval systems for argument search. There are, however, some more research fields worth exploring that are not part of this thesis. Considering that we only re-ranked the first five arguments for every query, the first is the influence of the number of re-ranked axioms on the retrieval results. Another one is the combination of axioms. In particular, it can be further evaluated which axioms should be chosen for combination and how these axioms can be weighted. A third idea worth exploring is to take a closer look at how axioms can improve the quality of arguments. This includes both the overall quality as well as an evaluation of whether axioms can be used to improve single quality metrics like the ones proposed by Wachsmuth et al. [2017a]

We already tried two further retrieval systems but discarded those because of bad retrieval results. The first is query expansion, where we use Word2Vec to add semantically similar words to the query. The second system is to define a new axiom that counts the named entities of an argument using the SpaCy entity tagger.

Bibliography

Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Data Acquisition for Argument Search: The args.me corpus. In Christoph Benzmüller and Heiner Stuckenschmidt, editors, *42nd German Conference on Artificial Intelligence (KI 2019)*, pages 48–59, Berlin Heidelberg New York, September 2019. Springer. doi: 10.1007/978-3-030-30179-8_4.

Alexander Bondarenko, Matthias Hagen, Michael Völske, Benno Stein, Alexander Panchenko, and Chris Biemann. Webis at TREC 2018: Common core track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018*, volume 500-331 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2018. URL `https://trec.nist.gov/pubs/trec27/papers/Webis-CC.pdf`.

Alexander Bondarenko, Maik Fröbe, Vaibhav Kasturia, Matthias Hagen, Michael Völske, and Benno Stein 0001. Webis at trec 2019: Decision track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2019. URL `https://trec.nist.gov/pubs/trec28/papers/Webis.D.pdf`.

Alexander Bondarenko, Lukas Gienapp, Maik Fröbe, Meriem Beloucif, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2021: Argument Retrieval. In K. Selçuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 12th International Conference of the CLEF Association (CLEF 2021)*, volume 12880 of *Lecture Notes in Computer Science*, pages 450–467, Berlin Heidelberg New York, September 2021. Springer. doi: 10.1007/978-3-030-85251-1_28. URL https://link.springer.com/chapter/10.1007/978-3-030-85251-1_28.

Alexander Bondarenko, Maik Fröbe, Jan Heinrich Reimer, Benno Stein, Michael Völske, and Matthias Hagen. Axiomatic Retrieval Experimentation with ir_axioms. In *45th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2022)*. ACM, July 2022.

Christopher J. C. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11, 01 2010. URL https://api.semanticscholar.org/CorpusID:397316.

Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. TARGER: Neural Argument Mining at Your Fingertips. In Martha R. Costa-jussà and Enrique Alfonseca, editors, *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 195–200. Association for Computational Linguistics, July 2019. URL https://www.aclweb.org/anthology/P19-3031.

Ronan Cummins and Colm O'Riordan. A constraint to automatically regulate document-length normalisation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 2443–2446, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311564. URL https://doi.org/10.1145/2396761.2398662.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 115–122, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933697. doi: 10.1145/1148170.1148193. URL https://doi.org/10.1145/1148170.1148193.

Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 49–56, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138814. URL https://doi.org/10.1145/1008992.1009004.

Hui Fang, Tao Tao, and Chengxiang Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2), apr 2011. ISSN 1046-8188. URL https://doi.org/10.1145/1961209.1961210.

Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 381–390, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584874. URL https://doi.org/10.1145/1526709.1526761.

Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 721–730, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340731. URL https://doi.org/10.1145/2983323.2983704.

M. Kaptein and E. van den Heuvel. *Statistics for Data Scientists: An Introduction to Probability, Statistics, and Data Analysis*. Undergraduate Topics in Computer Science, UTiCS. Springer International Publishing, 2022. ISBN 9783030105327. URL https://books.google.de/books?id=e4K8zwEACAAJ.

Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In Hujun Yin, Ke Tang, Yang Gao, Frank Klawonn, Minho Lee, Thomas Weise, Bin Li, and Xin Yao, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 611–618, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 7–16, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307178. URL https://doi.org/10.1145/2063576.2063584.

Frank McSherry and Marc Najork. Computing information retrieval performance measures efficiently in the presence of tied scores. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *Advances in Information Retrieval*, pages 414–421, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-78646-7.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

Martin Potthast, Lukas Gienapp, Florian Euchner, Nick Heilenkötter, Nico Weidmann, Henning Wachsmuth, Benno Stein, and Matthias Hagen. Argument Search: Assessing Argument Relevance. In *42nd International ACM Conference on Research and Development in Information Retrieval (SIGIR 2019)*. ACM, July 2019. doi: 10.1145/3331184.3331327. URL `http://doi.acm.org/10.1145/3331184.3331327`.

Jones Ranstam. Multiple p-values and bonferroni correction. *Osteoarthritis and cartilage*, 24(5):763–764, 2016.

Jan Heinrich Reimer, Johannes Huck, and Alexander Bondarenko. Grimjack at Touché 2022: Axiomatic Re-ranking and Query Reformulation. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *Working Notes Papers of the CLEF 2022 Evaluation Labs*, volume 3180 of *CEUR Workshop Proceedings*, September 2022. URL `http://ceur-ws.org/Vol-3180/paper-260.pdf`.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Shuming Shi, Ji-Rong Wen, Qing Yu, Ruihua Song, and Wei-Ying Ma. Gravitation-based model for information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 488–495, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930345. URL `https://doi.org/10.1145/1076034.1076117`.

Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 295–302, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277794. URL `https://doi.org/10.1145/1277741.1277794`.

Frans H. van Eemeren and Rob Grootendorst. *A Systematic Theory of Argumentation: The pragma-dialectical approach.* Cambridge University Press, 2003. doi: 10.1017/CBO9780511616389.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. Computational Argumentation Quality Assessment in Natural Language. In

Phil Blunsom, Alexander Koller, and Mirella Lapata, editors, *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 176–187, April 2017a. URL `http://aclweb.org/anthology/E17-1017`.

Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an Argument Search Engine for the Web. In Kevin Ashley, Claire Cardie, Nancy Green, Iryna Gurevych, Ivan Habernal, Diane Litman, Georgios Petasis, Chris Reed, Noam Slonim, and Vern Walker, editors, *4th Workshop on Argument Mining (ArgMining 2017) at EMNLP*, pages 49–59. Association for Computational Linguistics, September 2017b. URL `https://www.aclweb.org/anthology/W17-5106`.

Hao Wu and Hui Fang. Relation based term weighting regularization. In Ricardo Baeza-Yates, Arjen P. de Vries, Hugo Zaragoza, B. Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors, *Advances in Information Retrieval*, pages 109–120, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 334–342, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133316. doi: 10.1145/383952.384019. URL `https://doi.org/10.1145/383952.384019`.

Wei Zheng and Hui Fang. Query aspect based term weighting regularization in information retrieval. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan Rüger, and Keith van Rijsbergen, editors, *Advances in Information Retrieval*, pages 344–356, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12275-0.

Appendix A

Query and Argument Examples

Table A.1: Statistics for the args.me Corpus

Statistic	Number of Words
mean	273.9
standard deviation	370.8
shortest document	0
longest document	15720
25th percentile	26
50th percentile	111
75th percentile	362

Argument units identified by TARGER are highlited in **bold**.

Query: how should nuclear waste be stored

Document ID: S78068d61-Afaf71f39
Relevance: 1
Quality: 1

Refutation to the Pros case It seems as though my opponent seems to worry about the nuclear power being weaponized, and its byproducts. He implies weaponized as he diverges into nuclear weapons, and as I stated in the definitions this debate is about electricity. R1: Weaponization We are not arguing whether or not it is justified to have nuclear weapons, so I assume you mean they may be weaponized therefore need to be banned. This argument is fairly faulty. The technology to actually make uranium enriched enough to be used in weapons is a process that has nothing to do with nuclear power. **[1] Further more the plants use techniques with the uranium basically rendering it useless for weaponize techniques.** Nuclear fuel it enriched from 3-5%. [2] You need 90% enriched uranium for nuclear weapons. [3] Linking Nuclear weapons to nuclear power is absurd. [1] Also you claim their defense is less at risk. New Zealand is not like the US, they are peaceful and out of the way. [4] **Their peace may not be their nuclear abolishment, rather they do not patrol the middle east.** R2: WasteMy opponents main case is the waste. He claims the waste is unstoppable and radioactive. Lets refute first his views on the storage. **Many plants store them in huge metal vaults (under security big time) filled with water, this keeps the radiation inside. [5] The waste actually overtime becomes non hazardous.** [6] The EPA and the NRC constantly regulate and check the disposal to make sure the storage is safe, unlike you claim. [6] Further more my opponent claims a byproduct so "large". This is actually false. The Nuclear industry actually emits less byproducts then other industries. [7] The only waste problem that you cite that makes any sense is what the waste is. Also the nuclear industry takes FULL responsibility for their waste is any problem occurs. (like they have to pay for it). [8] Why does this matter? Because they now have a financial reason to keep the storage system safe. Nuclear actually emits less waste then coal. [8, 9] **(this ties** in next). **Methods for Safer underground storage sites are being researched.** [8]Then my opponent claims the byproducts from nuclear are the most hazardous. Before we do this, let me cite things about our buddy coal. Well our buddy coal has more radioactive waste then Nuclear. [10] Who would have though that? Well sorry coal, we are not buddies anymore. Now, as I have proven Nuclear waste is stored with great regulation hindering it safe. Now, what about the byproducts from, lets say solar? **Some solar cells have many nasty byproducts as well. They use many toxic materials.** [11] Now, how is this worse then nuclear? Nuclear Stores the waste in safe facilities, whereas solar cannot store the waste. So in a comparison your argument is false as coal is already more radioactive, and solar has toxic byproducts that are not even stored. Further more (economically speaking) solar has a terrible type of problem: China. China imports most of the solar cells in the US. [12]Also objections to the nuclear power/weapon see above.Defense of my case Jobs My opponent claims this is false. As I stated the nuclear industry woudl make 3,000 jobs or more per plant. The Solar **industry can only make 10,000 new jobs total.** [13] In this case Nuclear is better then this green energy. For wind, 11% of the jobs are building the turbine, 5% for maintenance. [14] Why is this significant? Because this means 15% of the jobs wind creates are temporary jobs. The Nuclear business has 56,000 people working for it. banning this energy means 50,000 people lose jobs. [15] Cost of nuclear Power My opponent claims it costs 109$ per killowatt hour. I disagree:"Since 1987 the cost of producing electricity from has decreased from 3.63 cents per KWHr to 1.68 cents per KWHr in 2004 and plant availability has increased from 67% to over 90%." [16] 2$ is far off from what you claim! Nuclear Power is cheaper then all of the fossil fuel types, and competes with coal, a very cheap Fossil Fuel. [17] Also this:"The nation's 103 nuclear reactors were the lowest cost electricity producers of any source of expandable, baseload electricity in 2002." [18] Safety "I agree completely that nuclear is just as safe as other energy. " My opponentHe basically conceded, and is only saying he is worried of a melt down. The NRC has many rules and regulations so it would be harder for melt-down situations to occur. [19] "the design and operation of nuclear power plants aims to minimise the likelihood of accidents, and avoid major human consequences when they occur. " [20]My opponent forgets the only accident recently is fukishima, which could be easily prevented with newer reactors. Chernobyl for example was due to a flawed design. **Newer designs and regulations prevent these instances.** As these accidents are rare, and when the happen are easily controlled (3 mile island) [21], and therefore your argument is mainly a consesion as these instances **are rare and are easily controllable.** The **workers in these plants have rigorous training and can** prevent and be able to stop the incident occurs, and **control** it if prevention measures fail. [22] Green energies My opponent first goes after the emissions. My opponent concedes they Emmit's less emissions than solar. His counter is the construction. If It emits less then it will eventually even out and bet solar. There are actually methods that make nuclear powers uranium last forever by getting uranium from the sea. [23] So your worry is over... Also as I stated last round, these reactors work on thorium too, therefore will last longer there too. Worry on your part over. *Low on room*ConclusionMy opponent has not fulfilled his BOP (he has it all, 1st round) and in my opinion has no reasons to do with the energy forever. http://www.nei.org... [1]http://www.nei.org... [2]http://nuclearfiles.org... [3]http://www.washingtontimes.com... [4]http://www.nei.org... [5]http://www. nrc.gov... [6]http://www.nrc.gov... [7]http://world-nuclear.org... [8]http://www.iaea.org... [9]http://www. scientificamerican.com... [10]http://en.wikipedia.org... [11]http://en.wikipedia.org... [12] http://grist.org... [13]http://www.bls.gov... [14]http://www.bls.gov... [15] http://nuclearinfo.net... [16]http://web.mit.edu... [17]http://www.nei.org... [18] http://www.nrc.gov... [19] http://www.world-nuclear.org... [20]http://www.world-nuclear.org... [21]http://www.nei.org... [22]http://en.wikipedia .org...[23]

Document ID: Sb3cf5511-Abeeb2b59
Relevance: 2
Quality: 1

I affirm Resolved: On balance, the benefits of nuclear power outweigh the risks. To win this debate, I only need to show that, in general principles (on balance), the benefits associated with nuclear power as a whole outweigh the risks of the technology. For example, Three Mile Island is an atypical example of nuclear power as a whole, and is therefore not an accurate way to frame this debate. I will now first address my opponent's case before introducing my own.1. Construction Costs:Listing two or three examples is not an accurate representation of the nuclear industry as a whole. The average **nuclear power plant is in** fact **extremely beneficial** to **local economies, resulting in a net economic gain that far exceeds construction costs.** The numbers speak for themselves: the average nuclear plant generates 400-700 permanent jobs, $20 million in total state and local tax revenue, $75 in federal tax payments, and nearly $430 million in local economic output annually. [(1) http://flv.texasgulfcoastonline.com...]Considering that the life of a typical nuclear reactor is 30-40 years, total community economic output ($430,000,000 x 30, $430,000,000 x 40) amounts to $12.9-17.2 billion. This **exceeds the already atypical $8 billion in start-up** costs cited by my opponent. [(2) http://www.iaea.org...]Therefore, the typical economic benefits of a nuclear power plant are greater than the economic investment.2. Health EffectsThis seems to be the structure of Con's argument:P1: Exposure to a lot of radiation is detrimental one's health.P2: There's a lot of radiation at nuclear power plants.C: Therefore, working at a nuclear **power plant is detrimental to one's health.While nuclear power plants obviously have higher radiation levels than surrounding environments, my opponent only cites health detriments to extreme amounts of radiation exposure, without regard to the amount actually present at power plants.From the New York Times, March 14, 2011 [(3) http://www.nytimes.com...]:"[I]n the United States the usual radiation exposure limit for nuclear power plant workers is 50 millisieverts, or 5 rem, per year (compared with the 0.3 rem that the Environmental Protection Agency says most people get from normal background radiation).** When there is an emergency, the limit can be raised to 25 rem, which is still far below the level at which people would show symptoms or get sick."**Radiation exposure is further mitigated by the fact that in areas with the most potential for exposure, workers wear full-body suits and/or take shifts to reduce the intensity of absorption. Concerns for surrounding communities are also alleviated in that emissions by nuclear plants into environmental surroundings are insignificant compared to natural levels already present [(4) http://www.heritage.org...].If my opponent really wants to get into human health and its relation to nuclear power plants, it is worth noting that most plant workers are unionized and generally receive health insurance and other benefits, as well as exercise on the job [(5) http://centralny.ynn.com...].3.** MeltdownsCon cites only two meltdown scenarios (Chernobyl and Three Mile Island) to justify the inherent risks of nuclear power. Today, such concerns are unfounded. Modern nuclear reactors are free from the design flaws that caused the Chernobyl plant to explode, and reactors now have multiple containment cores to prevent the escape of nuclear material even if an accident does occur [(6) http://discovermagazine.com...].The Three Mile island meltdown was also contained, and no detrimental health effects have been reported. Today's water reactors, which slow neuron emissions to keep reaction rates steady, make major meltdowns extremely improbable [6]. Regarding storage, my opponent admits that storing nuclear waste in steel and concrete casks is optimal, but that Congress has not acted on such proposals. This has nothing to do with the risks of nuclear waste, which can in fact be stored safely, but rather the ineptitude of Congress itself. This point should be dropped from the round.When stored in steel and concrete casks, uranium waste can be stored for at least a century, and can often be recycled to meet future energy needs., In fact, an amount the volume of one's fingertip posseses the energy potential of 1,780 pounds of coal [6]. Such uranium is useful, efficient, and can be stored safely.4. TerrorismContrary to my opponent's claims, modern plants are built to survive a terrorist attack. Plants are built to withstand the impact of a passenger airplane [1]. Further, if pilots are caught loitering over nuclear sites, the FAA has a policy of detaining and interrogating those responsible [4].Con claims that terrorists have an incentive to attack nuclear power plants, but terrorists also have an incentive to attack oil rigs, atomic testing sites, and the White House. As Jack Spencer of the Heritage Foundation explains, "A successful terrorist attack against a nuclear power plant could have severe consequences, as would attacks on schools, chemical plants, or ports. However, fear of a terrorist attack is not a sufficient reason to deny society access to any of these critical assets" [4]. At the point where my opponent never quantifies the likelihood of a terrorist attack on a nuclear facility, he is only operating on a "what-if" scenario without gauging either the intent of terrorist groups or the likelihood that they could even conduct a successful attack.In my case I will provide turn this point, showing that nuclear power actually reduces terrorist activity, thereby shifting this voting issue over to Pro side.Now for my case...1. Nuclear Energy is the Best Way to Reduce Carbon Dioxide EmissionsTotal emissions over the life of a nuclear plant (including uranium mining, shipping, construction, etc.) are about the same as hydroelectric plants and wind farms, and emit less carbon dioxide than solar plants. The **efficiency of nuclear power (the energy potential** of 1,780 pounds of coal in **an area the size of a fingertip) plus the overall positive economic impact of power plants makes nuclear energy the best renewable energy option for the United States.2.** Nuclear Energy Reduces TerrorismNuclear energy reduces U.S. dependence on oil from nations that continue to enable terrorist regimes. According to the Wall Street Journal, "One of the biggest dangers to our security is from oil nations providing support to anti-U.S. terrorist groups. The **faster we can move away from carbon-based energy, the faster we take away** that funding source" [(7) http://online.wsj.com...]**.** Nuclear energy **reduces** U.S. **oil dependence on nations such as Saudi Arabia and Russia, the latter of which uses Iran to funnel weapons directly to Hezbollah [(8) http://www.cfr.org...].Nuclear power also reduces nuclear proliferation by terrorists.** Atomic warheads are excellent as reactor fuel, and currently amount to 15% of **world** nuclear fuel. **Expanding** the **use** of nuclear energy and the resulting increased demand for reactor fuel will divert warheads away from terrorist groups, reducing the threat of nuclear terrorism against the U.S. [(9) http://www.spiked-online.com...]. ConclusionNuclear energy is a safe, reliable and efficient way for the United States to **become energy independent while reducing greenhouse gas emissions and mitigating terrorist threats**. The harms proposed by my opponent are insignificant if not nonexistant. On balance, the benefits of nuclear power outweigh the risks. The resolution is affirmed.

Document ID: S89d1bea-Ab89d1614
Relevance: 2
Quality: 2

Underground nuclear **waste** storage means that nuclear waste is stored at least 300m underground. [I1] The harm of a leak 300m underground is significantly limited, if the area has been chosen correctly then there should be no water sources nearby to contaminate. If this is the case, then a leak's harm would be limited to the layers of sediment nearby which would be unaffected by radiation. By comparison **a leak outside** might lead to animals nearby suffering from contamination. Further nuclear waste might reach water sources should there be a leak above ground, if it is raining heavily when the leak happens for example. Further, the other options available, such as above ground storage present a potentially greater danger, should something go wrong. This **is because it is much easier for nuclear waste to leak radiation into the air.** This is problematic because even a hint of radiation may well cause people to panic owing to the **damaging** and heavily publicised consequences **of previous nuclear safety crises**. As such, underground storage is safer both directly and indirectly.[1] As well as this, underground storage also prevents nuclear waste or nuclear radiation from reaching other states and as such, results in greater safety across borders.[2] Further, storing all nuclear waste underground means that countries can concentrate their research and training efforts on responding to subterranean containment failures. Focus and specialisation of this type is much more likely to avert a serious release of **nuclear material** from **an underground** facility **than** the **broad** and **general approach that will be fostered by diverse and distinct above-ground storage** solutions**.** [1] "Europe eyes underground nuclear waste repositories." Infowars Ireland. 20/02/2010 http://infowars.org/2010/02/20/europe-eyes-underground-nuclear-waste-repositories/ [2] "EU Debates Permanent Storage For Nuclear Waste." 04/11/2010 AboutMyPlanet. http://www.aboutmyplanet.com/environ- ment/eu-debates-perma nent-storage-for-nuclear-waste/ [I1]I am not sure how to replace this section. “Leakage” of radioactive material into the air is a minimal danger. The contributor may be referring to the ejection of irradiated dust and other particulates that has occurred when nuclear power stations have suffered explosive containment failures, but this is not comparable to the types of containment failures that might happen in facilities used to store spent nuclear fuel rods and medical waste. One of **the more** substantial risks presented by underground storage is release of nuclear material into a water source.

Appendix B

Combining Axiom Using the VoteAxiom

Table B.1: Combinations of QSenSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance. The axioms are combined using the VoteAxiom. The QSenSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{mean}$	**0.658**
QSenSim$_{mean}$, aSL	0.653
QSenSim$_{mean}$, QTArg	0.651
QSenSim$_{mean}$, ArgUC	0.649
QSenSim$_{mean}$, QTPArg	0.648
QSenSim$_{mean}$, ArgUC, QTArg	0.645
QSenSim$_{mean}$, QTArg, aSL	0.645
QSenSim$_{mean}$, QTArg, QTPArg, aSL	0.642
QSenSim$_{mean}$, ArgUC, QTPArg	0.642
QSenSim$_{mean}$, ArgUC, QTArg, QTPArg, aSL	0.642
QSenSim$_{mean}$, ArgUC, QTArg, aSL	0.641
QSenSim$_{mean}$, QTArg, QTPArg	0.641
QSenSim$_{mean}$, ArgUC, QTArg, QTPArg	0.641
QSenSim$_{mean}$, ArgUC, aSL	0.640
QSenSim$_{mean}$, QTPArg, aSL	0.640
QSenSim$_{mean}$, ArgUC, QTPArg, aSL	0.640
DirichletLM	0.633
QSenSim$_{max}$, aSL	0.671
QSenSim$_{max}$	**0.669**
QSenSim$_{max}$, QTArg	0.666
QSenSim$_{max}$, ArgUC	0.665
QSenSim$_{max}$, QTPArg	0.659
QSenSim$_{max}$, QTArg, aSL	0.647
QSenSim$_{max}$, ArgUC, QTPArg, aSL	0.646
QSenSim$_{max}$, ArgUC, QTArg	0.644
QSenSim$_{max}$, ArgUC, QTArg, aSL	0.644
QSenSim$_{max}$, ArgUC, QTArg, QTPArg, aSL	0.644
QSenSim$_{max}$, ArgUC, QTPArg	0.643
QSenSim$_{max}$, QTArg, QTPArg, aSL	0.643
QSenSim$_{max}$, QTArg, QTPArg	0.642
QSenSim$_{max}$, QTPArg, aSL	0.641
QSenSim$_{max}$, ArgUC, QTArg, QTPArg	0.639
QSenSim$_{max}$, ArgUC, aSL	0.638†
DirichletLM	0.633†

Table B.2: Combinations of QArgSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance. The axioms are combined using the VoteAxiom. The QArgSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QArgSim$_{mean}$	**0.661**
QArgSim$_{mean}$, aSL	0.660
QArgSim$_{mean}$, ArgUC	0.656
QArgSim$_{mean}$, QTArg	0.654
QArgSim$_{mean}$, ArgUC, QTArg	0.650
QArgSim$_{mean}$, QTPArg	0.647
QArgSim$_{mean}$, ArgUC, QTArg, aSL	0.647
QArgSim$_{mean}$, ArgUC, QTPArg	0.647
QArgSim$_{mean}$, ArgUC, QTArg, QTPArg, aSL	0.646
QArgSim$_{mean}$, QTArg, aSL	0.646
QArgSim$_{mean}$, QTArg, QTPArg, aSL	0.645
QArgSim$_{mean}$, ArgUC, QTPArg, aSL	0.645
QArgSim$_{mean}$, QTArg, QTPArg	0.643
QArgSim$_{mean}$, ArgUC, QTArg, QTPArg	0.642
QArgSim$_{mean}$, ArgUC, aSL	0.641
QArgSim$_{mean}$, QTPArg, aSL	0.640
DirichletLM	0.633
QArgSim$_{max}$	**0.672**
QArgSim$_{max}$, aSL	0.669
QArgSim$_{max}$, QTArg	0.664
QArgSim$_{max}$, ArgUC	0.663
QArgSim$_{max}$, QTPArg	0.657
QArgSim$_{max}$, ArgUC, QTArg	0.654
QArgSim$_{max}$, ArgUC, QTPArg	0.649
QArgSim$_{max}$, QTArg, aSL	0.648
QArgSim$_{max}$, ArgUC, QTArg, QTPArg, aSL	0.647
QArgSim$_{max}$, ArgUC, QTPArg, aSL	0.646
QArgSim$_{max}$, ArgUC, QTArg, aSL	0.646
QArgSim$_{max}$, QTArg, QTPArg, aSL	0.645
QArgSim$_{max}$, QTArg, QTPArg	0.644
QArgSim$_{max}$, ArgUC, QTArg, QTPArg	0.643
QArgSim$_{max}$, QTPArg, aSL	0.641†
QArgSim$_{max}$, ArgUC, aSL	0.641†
DirichletLM	0.633†

Table B.3: Combinations of QSenSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance. The axioms are combined using the VoteAxiom. The QSenSim baseline is highlighted in bold. QSenSim is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{mean}$, A	0.662
QSenSim$_{mean}$	**0.658**
QSenSim$_{mean}$, H	0.651
QSenSim$_{mean}$, C	0.645
QSenSim$_{mean}$, D	0.644
QSenSim$_{mean}$, F	0.636
QSenSim$_{mean}$, E	0.635
QSenSim$_{mean}$, G	0.634
DirichletLM	0.633
QSenSim$_{mean}$, I	0.633
QSenSim$_{mean}$, B	0.633
QSenSim$_{max}$, A	0.670
QSenSim$_{max}$	**0.669**
QSenSim$_{max}$, H	0.659
QSenSim$_{max}$, D	0.655
QSenSim$_{max}$, C	0.641†
QSenSim$_{max}$, E	0.639†
QSenSim$_{max}$, F	0.636†
QSenSim$_{max}$, B	0.635†
QSenSim$_{max}$, G	0.634†
DirichletLM	0.633†
QSenSim$_{max}$, I	0.633†

Table B.4: Combinations of QArgSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance. The axioms are combined using the VoteAxiom. The QSenSim baseline is highlighted in bold. QArgSim is shown for comparison.

Axiom Combination	nDCG@5
$\textbf{QArgSim}_{mean}$	**0.661**
QArgSim_{mean}, A	0.661
QArgSim_{mean}, H	0.656
QArgSim_{mean}, D	0.652
QArgSim_{mean}, E	0.641
QArgSim_{mean}, C	0.640
QArgSim_{mean}, F	0.638
QArgSim_{mean}, B	0.635
DirichletLM	$0.633^{\dagger}$
QArgSim_{mean}, I	$0.633^{\dagger}$
QArgSim_{mean}, G	$0.633^{\dagger}$
$\textbf{QArgSim}_{max}$	**0.672**
QArgSim_{max}, A	0.670
QArgSim_{max}, H	0.656
QArgSim_{max}, D	$0.645^{\dagger}$
QArgSim_{max}, C	$0.641^{\dagger}$
QArgSim_{max}, E	$0.640^{\dagger}$
QArgSim_{max}, F	$0.638^{\dagger}$
DirichletLM	$0.633^{\dagger}$
QArgSim_{max}, I	$0.633^{\dagger}$
QArgSim_{max}, G	$0.633^{\dagger}$
QArgSim_{max}, B	$0.633^{\dagger}$

Appendix C

Axiom Combinations Evaluated on the Touché 2020 Topics

Table C.1: Combinations of QSenSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance on the Touché 2020 dataset. The QSenSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{meam}$+aSL	0.832
QSenSim$_{meam}$+QTArg+aSL	0.831
QSenSim$_{meam}$+QTPArg+aSL	0.829
QSenSim$_{meam}$+QTPArg	0.828
QSenSim$_{meam}$+ArgUC+QTPArg+aSL	0.828
QSenSim$_{meam}$+ArgUC	0.828
QSenSim$_{meam}$	**0.828**
QSenSim$_{meam}$+ArgUC+aSL	0.827
QSenSim$_{meam}$+QTArg	0.827
QSenSim$_{meam}$+ArgUC+QTPArg	0.826
QSenSim$_{meam}$+QTArg+QTPArg	0.825
QSenSim$_{meam}$+QTArg+QTPArg+aSL	0.825
QSenSim$_{meam}$+ArgUC+QTArg	0.824
QSenSim$_{meam}$+ArgUC+QTArg+QTPArg+aSL	0.824
QSenSim$_{meam}$+ArgUC+QTArg+aSL	0.823
QSenSim$_{meam}$+ArgUC+QTArg+QTPArg	0.823
DirichletLM	0.810
QSenSim$_{max}$+QTPArg+aSL	0.835
QSenSim$_{max}$+QTArg+QTPArg+aSL	0.833
QSenSim$_{max}$+aSL	0.833
QSenSim$_{max}$+QTPArg	0.833
QSenSim$_{max}$+ArgUC+QTPArg	0.831
QSenSim$_{max}$+ArgUC+QTPArg+aSL	0.831
QSenSim$_{max}$	**0.830**
QSenSim$_{max}$+QTArg+QTPArg	0.829
QSenSim$_{max}$+ArgUC	0.829
QSenSim$_{max}$+QTArg+aSL	0.828
QSenSim$_{max}$+ArgUC+QTArg+QTPArg	0.828
QSenSim$_{max}$+ArgUC+QTArg+QTPArg+aSL	0.827
QSenSim$_{max}$+ArgUC+aSL	0.827
QSenSim$_{max}$+QTArg	0.826
QSenSim$_{max}$+ArgUC+QTArg+aSL	0.824
QSenSim$_{max}$+ArgUC+QTArg	0.821
DirichletLM	0.810

Table C.2: Combinations of QArgSim axioms with argumentative axioms implemented in ir_axioms evaluated for argument relevance on the Touché 2020 dataset. The QArgSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QArgSim$_{mean}$+aSL	0.823
QArgSim$_{mean}$+QTArg+aSL	0.823
QArgSim$_{mean}$+ArgUC+aSL	0.823
QArgSim$_{mean}$+ArgUC	0.822
QArgSim$_{mean}$+ArgUC+QTArg	0.821
QArgSim$_{mean}$+ArgUC+QTArg+QTPArg+aSL	0.821
QArgSim$_{mean}$+ArgUC+QTArg+QTPArg	0.821
QArgSim$_{mean}$+ArgUC+QTArg+aSL	0.820
QArgSim$_{mean}$	**0.820**
QArgSim$_{mean}$+ArgUC+QTPArg+aSL	0.820
QArgSim$_{mean}$+QTArg	0.819
QArgSim$_{mean}$+QTArg+QTPArg+aSL	0.819
QArgSim$_{mean}$+QTArg+QTPArg	0.818
QArgSim$_{mean}$+QTPArg+aSL	0.817
QArgSim$_{mean}$+ArgUC+QTPArg	0.817
QArgSim$_{mean}$+QTPArg	0.816
DirichletLM	0.810
QArgSim$_{max}$+QTArg+aSL	0.828
QArgSim$_{max}$+QTPArg	0.827
QArgSim$_{max}$+QTArg	0.827
QArgSim$_{max}$+ArgUC	0.827
QArgSim$_{max}$+aSL	0.827
QArgSim$_{max}$+QTPArg+aSL	0.826
QArgSim$_{max}$+ArgUC+QTArg+QTPArg	0.826
QArgSim$_{max}$+QTArg+QTPArg	0.825
QArgSim$_{max}$+ArgUC+QTArg	0.825
QArgSim$_{max}$+ArgUC+aSL	0.825
QArgSim$_{max}$+QTArg+QTPArg+aSL	0.825
textbfQArgSim$_{max}$	**0.824**
QArgSim$_{max}$+ArgUC+QTPArg	0.824
QArgSim$_{max}$+ArgUC+QTArg+QTPArg+aSL	0.823
QArgSim$_{max}$+ArgUC+QTArg+aSL	0.822
QArgSim$_{max}$+ArgUC+QTPArg+aSL	0.822
DirichletLM	0.810

Table C.3: Combinations of QSenSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance on the Touché 2020 dataset. The QSenSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QSenSim$_{mean}$	**0.828**
QSenSim$_{mean}$ + A	0.821
QSenSim$_{mean}$ + B	0.821
QSenSim$_{mean}$ + C	0.819
QSenSim$_{mean}$ + H	0.819
QSenSim$_{mean}$ + G	0.818
QSenSim$_{mean}$ + I	0.818
QSenSim$_{mean}$ + D	0.816
QSenSim$_{mean}$ + F	0.814
DirichletLM	0.810
QSenSim$_{mean}$ + E	0.808
QSenSim$_{max}$	**0.830**
QSenSim$_{max}$ + A	0.825
QSenSim$_{max}$ + B	0.824
QSenSim$_{max}$ + H	0.824
QSenSim$_{max}$ + C	0.820
QSenSim$_{max}$ + G	0.818
QSenSim$_{max}$ + F	0.817
QSenSim$_{max}$ + I	0.817
QSenSim$_{max}$ + D	0.814
DirichletLM	0.810
QSenSim$_{max}$ + E	0.804

Table C.4: Combinations of QArgSim axioms with axioms that have the highest feature importance for improving the nDCG@5 with LambdaMART evaluated for argument relevance on the Touché 2020 dataset. The QArgSim baseline is highlighted in bold. DirichletLM is shown for comparison.

Axiom Combination	nDCG@5
QArgSim$_{mean}$	**0.820**
QArgSim$_{mean}$ + I	0.817
QArgSim$_{mean}$ + F	0.816
QArgSim$_{mean}$ + G	0.815
QArgSim$_{mean}$ + H	0.814
QArgSim$_{mean}$ + A	0.814
QArgSim$_{mean}$ + B	0.813
QArgSim$_{mean}$ + E	0.813
QArgSim$_{mean}$ + C	0.812
QArgSim$_{mean}$ + D	0.812
DirichletLM	0.810
QArgSim$_{max}$ + A	0.825
QArgSim$_{max}$	**0.824**
QArgSim$_{max}$ + C	0.824
QArgSim$_{max}$ + B	0.820
QArgSim$_{max}$ + F	0.817
QArgSim$_{max}$ + I	0.816
QArgSim$_{max}$ + H	0.816
QArgSim$_{max}$ + G	0.816
QArgSim$_{max}$ + E	0.813
QArgSim$_{max}$ + D	0.811
DirichletLM	0.810

Appendix D

Axioms

Table D.1: List of existing Axioms (1)

Axiom	Description	Source
Term Frequency		
TFC1	Prefer the document with more query terms.	Fang et al. [2004]
TFC2	Reward the increase of a small term frequency more than the increase of a big one.	Fang et al. [2004]
TFC3	Prefer the document with more different query terms.	Fang et al. [2011]
TDC	Prefer the document that contains terms with a high inverse document frequency.	Fang et al. [2004]
M-TDC	Like TDC but with changes in the definition to prevent unintuitive preferences.	Shi et al. [2005]
LEN-M-TDC	Like M-TDC but assumes that both documents have the same length.	Bondarenko et al. [2022]
Document Length		
LNC1	Penalize longer documents for non-query terms.	Fang et al. [2004]
LNC2	Do not prefer shorter documents when both documents contain the same percentage of query terms.	Fang et al. [2004]
TF-LNC	Rewarding additional query terms more than document length is penalized.	Fang et al. [2004]
QLNC	Penalize longer documents for not containing query terms.	Cummins and O'Riordan [2012]
Lower-Bound Term Frequency		
LB1	Prefer a long document with a non-zero term frequency over a short document with a zero term frequency.	Lv and Zhai [2011]
LB2	First occurrences of query terms are more important than repeated occurrences.	Lv and Zhai [2011]

Table D.2: List of existing Axioms (2)

Axiom	Description	Source
Query Aspects		
REG	Prefer the document that covers more query aspects. (Prefer the document that contains the query term least similar to all other query terms.)	Zheng and Fang [2010]
ANTI-REG	Prefer the document that contains the query term most similar to all other query terms.	Bondarenko et al. [2022]
ASPECT-REG	Prefer the document that contains query terms of more query aspects.	Bondarenko et al. [2022]
AND	Prefer the document that contains all query terms.	Wu and Fang [2012]
LEN-AND	Prefer the document that contains all query terms if both documents have the same length.	Bondarenko et al. [2022]
M-AND	Prefer the document that contains the larger subset of query terms.	Bondarenko et al. [2022]
LEN-M-AND	Prefer the document that contains a larger subset of query terms if both documents have the same length.	Bondarenko et al. [2022]
DIV	Prefer the document whose term set is less similar to the query term set.	Gollapudi and Sharma [2009]
LEN-DIV	Prefer the document whose term set is less similar to the query term set if both documents have the same length.	Bondarenko et al. [2022]
RSIM	Prefer one document from each document similarity cluster while having no preference for documents from different clusters.	Hagen et al. [2016]
Semantic Similarity		
STMC1	Prefer the document with terms more semantically similar to the query terms.	Fang and Zhai [2006]
STMC2	Occurrences of a term should be rewarded more than occurrences of semantically similar terms.	Fang and Zhai [2006]
STMC3	Do not reward the occurrence of more different query terms if the terms are equally important.	Fang and Zhai [2006]
Term Proximity		
PROX1	Prefer the document with a shorter average distance between query term pairs.	Hagen et al. [2016]
PROX2	Prefer the document where the first occurrence of each query term is closer to the beginning of the document on average.	Hagen et al. [2016]
PROX3	Prefer the document that contains the query as a phrase closer to the beginning of the document.	Hagen et al. [2016]
PROX4	Prefer the document that contains fewer non-query terms in the closest grouping of all query terms.	Hagen et al. [2016]
PROX5	Prefer the document that has the smallest text span containing all query terms for every occurrence of every query term on average.	Hagen et al. [2016]
QPHRA	Prefer the document containing all highlighted query phrases.	Hagen et al. [2016]
PHC	The retrieval score of the document with the smaller query proximity distance should be increased stronger.	Tao and Zhai [2007]
CCC	The proximity distance function should fall off fast for small distances and be almost constant for large distances.	Tao and Zhai [2007]

Table D.3: List of existing Axioms (3)

Axiom	**Description**	**Source**
Argumentativeness		
ArgUC	Prefer the document with more argumentative units.	Bondarenko et al. [2018]
QTArg	Prefer the document with more query terms in argumentative units.	Bondarenko et al. [2018]
QTPArg	Prefer the document where the first query term in an argumentative unit is closer to the beginning of the document.	Bondarenko et al. [2018]
aSL	Prefer the document with an average sentence length between 12 and 20 words.	Bondarenko et al. [2019]
MEArg	Prefer the document with more medical entities in argumentative units.	Bondarenko et al. [2019]
CompArg	Prefer the document with more comparative objects in argumentative units.	Reimer et al. [2022]
CompPArg	Prefer the document where comparative objects occur earlier in argumentative units.	Reimer et al. [2022]
ArgQ	Prefer the document with a higher argument quality.	Reimer et al. [2022]
QSenSim$_{mean}$	Prefer the document whose sentences are more similar to the query.	Section 3.2, Axiom 1
QSenSim$_{max}$	Prefer the document that has the sentence that is most similar to the query.	Section 3.2, Axiom 2
QArgSim$_{mean}$	Prefer the document whose argumentative units are more similar to the query.	Section 3.3, Axiom 3
QArgSim$_{max}$	Prefer the document that has the argumentative unit that is the most similar to the query.	Section 3.3, Axiom 4
Retrieval Score		
RS-{Model}	Prefer documents that are assigned a higher score by the given retrieval model.	Bondarenko et al. [2022]
Other		
ORIG	Prefer the document ranked higher in the original ranking.	Hagen et al. [2016]
ORACLE	Prefer the document ranked higher by human judgments.	Hagen et al. [2016]
NOP	Never prefer any document.	Bondarenko et al. [2022]
RANDOM	Prefer documents randomly.	Bondarenko et al. [2022]