Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

BACHELORTHESIS

Exploring Large Language Models in Argument Retrieval for Content- and Demographic-Relevance

Robert Günzler

Field of Study: Informatik
Matriculation No.: 7495157
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Dr. Steffen Remus, Universität Hamburg

Language Technology
Department of Informatics
Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg
Hamburg, Germany

A thesis submitted for the degree of

Bachelor of Science (B. Sc.)

Printed on March 10, 2025

Exploring Large Language Models in Argument Retrieval for Content- and Demographic-Relevance

Bachelor's Thesis submitted by: Robert Günzler

Date of Submission: 10.03.2024

Supervisor(s):
Özge Sevgili Ergüven, Universität Hamburg
Dr. Irina Nikishina, Universität Hamburg
Dr. Steffen Remus, Universität Hamburg

Committee:
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Dr. Steffen Remus, Universität Hamburg

Universität Hamburg, Hamburg, Germany
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Language Technology

Affidavit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Bachelorstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

I hereby declare in lieu of an oath that I have written this thesis for the Bachelor's degree programme in Computer Science independently and have not used any aids other than those specified - in particular no Internet sources not named in the list of sources. All passages taken verbatim or in spirit from publications are labelled as such. I further certify that I have not previously submitted the thesis in another examination procedure.

10.03.2025

Date

Signature

(Robert Günzler)

Acknowledgements

I want to take this opportunity to thank my supervisors, Özge, Irina and Steffen, for the continued support throughout this entire project. I want to thank you for all the content-related guidance and all helpful suggestions you have given me, but also for all of the encouragement, and for staying patient with me throughout the process. I would also like to thank Prof. Dr. Chris Biemann for providing me with the opportunity to write my Bachelorthesis in such an exciting field and for the chance to even publish a short system description paper in the process.

Additionally I would like to thank my family, my friends and especially Rojee for not losing patience with me and supporting me throughout my academic journey so far.

Abstract

In an era where political discourse increasingly unfolds online - via social media platforms and other digital channels - the capacity to automatically detect and process argumentative language is becoming crucial. Argument Retrieval focuses on automatically identifying relevant information with an emphasis on argumentative discourse. Extending this concept, Perspective Argument Retrieval also incorporates socio-demographic aspects into the retrieval process, addressing the need to mitigate biases and prevent discrimination in automated analyses.

This thesis examines the use of large, general-purpose language models (LLMs) for enhancing Perspective Argument Retrieval. The proposed approach operates in a two-stage retrieval pipeline. In the first stage, conventional retrieval methods are employed alongside dataset-specific techniques to extract top candidate arguments. The subsequent reranking stage leverages LLMs with two seperate tasks. First, the aim is to improve the overall relevance of retrieved arguments by focusing the reranking on content relevance, before an approach employing LLMs in demographic-relevance prediction is investigated. Finally, an integrated pipeline is explored that combines both reranking aspects in the two-stage retrieval pipeline.

Within the research phase for this thesis, an initial version of the integrated approach was submitted to the Perspective Argument Retrieval Shared Task 2024, where overall a second place was achieved, which emphasizes the potential of the methods investigated.

Contents

List of Figures

List of Tables

1 Introduction

When exchanging information about something, humans commonly use natural language. In this case, communication consists of both proven facts and opinions, that are subjective to people. An objective of Natural Language Processing (NLP) is to understand this complex landscape of communication and to be able to learn from it. In this matter, it is important to be able to distinguish subjective statements and opinions from facts and to be able to differentiate all different existing opinions on a specific topic. These information can be valuable to analyze social trends or to determine the needs of potential customers. It is worth noting, that these are examples of areas, in which subjective statements and opinions dominate public discourse, which makes the ability to extract opinions even more valuable. The analysis of different perspectives and opinions is the subject of Opinion Mining (also referred to as Sentiment Analysis) (Birjali et al., 2021).

To further understand *why* people have certain opinions or make specific statements, considering the reasoning behind said opinions or statements is important. Arguments, that might be used in order to ground an opinion or an attempt of persuasion, need to be further investigated. According to (Wachsmuth et al., 2017), an argument can be seen as a conclusion (claim) combined with one or multiple premises (reasons). Automatic extraction of such arguments from one or multiple text documents is subject of Argument Mining (Lawrence and Reed, 2019).

In the process of opinion forming the large quantity of opinions and attitudes with according reasoning also remains an issue. In this regard it is sensible to regard a set of well justified perspectives that match the specified topic. This yields several smaller problems, the first one being the necessity to retrieve only the arguments that are relevant to a specific topic from a larger corpus of arguments. Additionally, it entails the need to distinguish convincing arguments from less convincing or faulty ones. To also facilitate unbiased opinion forming and decision making, it can be sensible to consider a diverse set of perspectives. Diversity in this case could indicate differences in terms of reasoning or in terms of the initial claim and the stance towards a question or topic. Furthermore, it can be reasonable to also analyze the person stating an opinion. For instance, by identifying the demographic or socio-cultural background of a statements author, it is possible to depict the perspectives and opinions of a specific demographic or sociocultural group, or to draw an even more diverse set of arguments for decision making and opinion forming. These, among other challenges, are tackled in research about Argument Retrieval (Bondarenko et al., 2022).

Due to the rise in capabilities of Large Language Models (LLMs) over the last few years, many new possibilities in NLP have arisen. It has already been shown that LLMs are able to contribute to excellent results on multiple different tasks from NLP research, such as Information Extraction (Ma et al., 2023) or Machine Translation (Jiao et al., 2023). In Argument Retrieval LLMs have also been utilized already to improve results (Sun et al., 2023). In this scenario, LLMs are typically invoked to re-rank a smaller subset of arguments, that has initially been retrieved from a large set of arguments using sparse or dense retrieval (Luan et al., 2021). This two-step approach helps to avoid large computational costs of inference with LLMs and also ensures the set of arguments does not exceed the LLM's context window.

The prediction of demographic features of a statement's author is part of the research on Author Profiling. Typical fields of use for Author Profiling are marketing and advertising, where the prediction of a target group in an economic sense is the motivation to do Author Profiling. Research in this area often focuses on social-media activities (HaCohen-Kerner, 2022; Ouni et al., 2023). Less prominent in research is performing Author Profiling for single arguments, specifically in relation to politics or society. Regarding this, besides demographic features like age or gender, sociocultural aspects like religious affiliation or political orientation are also interesting research targets, as they enable to perform a differentiated analysis of public political opinions and especially as they facilitate creating an overview over prominent goals and attitudes of different demographic or social groups. Cardaioli et al. (2020) and David et al. (2016) perform research on people's political orientation based on social-media activity on Twitter (now X) or Facebook.

Demographic and sociocultural features of political characters are the objects of investigation in the Perspective Argument Retrieval shared task (Falk et al., 2024), which was initiated in 2024 with the intention to examine the implicit knowledge contained in LLMs about different perspectives and demographic backgrounds.

1.1 Research Questions

To successfully integrate Argument Retrieval and Perspective Retrieval with the use of LLMs in a pipeline, several questions need to be answered. In the following, these questions, that shall all be answered in this thesis, are explained in this section.

1.1.1 Research Question 1

The development in recent years has resulted in an at least two-step process as state-of-the-art in Information Retrieval. During the first step, a small amount of relevant documents is extracted from a relatively large corpus of documents using sparse or dense retrieval. In the second step, the small set of extracted documents is then re-ranked using a more effective but also more computationally expensive attention-based model (Luan et al., 2021) LLMs are trained on vast quantities of text data and thus obtain a lot of implicit knowledge. A difference from smaller neural networks, which are trained or fine-tuned on a specific task, is that ideally using this knowledge can be challenging.

Thus, the first research question, that shall be answered, is:

How can LLMs be used to enhance the results of an argument retrieval process?

1.1.2 Research Question 2

On the subject of Author Profiling research is often focused on marketing-relevant information, like identifying a target group for marketing or advertising measures. Usually, these analyses are based on larger amounts of documents, that could for instance result from a user's social-media activity. Existing research on predicting political orientation based on text documents is also mainly based on social-media activities. Predicting demographic and sociocultural features from just a single given argument is a task, that has barely been explored yet. The little information given in such a short text document complicates predictions immensely.

The large amounts of data used to train LLMs raise the hopes that LLMs might contain implicit knowledge, that can be useful to predict demographic or sociocultural backgrounds.

Thus, the second research question is:

How can LLMs be used to implicitly predict the demographic or sociocultural perspective of a text document?

1.1.3 Research Question 3

To achieve a good outcome in a practical scenario of perspective argument retrieval, it is necessary to find a way to integrate retrieval methods in one pipeline.

Thus, the third research question is:

How can LLM predictions on argument relevance and perspective relevance effictively be combined to retrieve relevant arguments for a given question or topic with additionally given sociocultural or demographic aspects?

1.2 Thesis Overview

This chapter introduced the key research questions and set the stage for the thesis. In the following chapters, Chapter 2 provides the necessary background information, Chapter 3 reviews related work in the field, Chapter 4 explains the methodology used, Chapter 5 presents the evaluation of the experiments, and Chapter 6 concludes the study with a summary of findings and suggestions for future research.

2
Background

This section will introduce background knowledge to provide a foundation for the reader to understand the technologies and methods described in the following chapters of this thesis. Therefore, in the following the concepts of machine learning (ML) and natural language processing (NLP) will be described. In this context, the NLP tasks of information retrieval (IR), argument retrieval, and author profiling will be described and large language models (LLMs) will be discussed. Following that will be a description of the logistic regression as a statistical learning algorithm before finally different evaluation metrics are explained.

2.1 Machine Learning

Machine Learning (ML) is a research area of computer science focusing on algorithms that improve their performance through experience. Mitchell (1997) fittingly defines a learning program: “A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”. In other words; ML models are trained to perform a certain task using data (experience). The evaluation of an ML model then mostly focuses on its ability to generalize from the learned data to new, unseen data.

In ML three core paradigms are distinguished, namely *supervised learning*, *unsupervised learning* and *reinforcement learning* (Jordan and Mitchell, 2015). In supervised learning, models learn from labeled data, where each training example consists of one or multiple input features and the expected output feature or category. The goal unsupervised learning on the other hand is the discovery of patterns or structures in unlabeled data. In reinforcement learning a reward feedback system is used to train a model. The idea is that the model is rewarded when reaching a desired state or achieving a desirable goal so that it can learn, which actions lead to a desirable state by using a trial-and-error principle with the aim to maximize the rewards (Bishop, 2007).

ML is comprised of a wide range of algorithms, including e.g. decision trees, linear or logistic regression, or the k-nearest-neighbors algorithm. In recent years, *neural networks* (which are inspired by the widely connected neurons in the human brain) have gained a lot of attention. Neural networks can range from simple multilayer perceptrons that can be used to solve simple classification or regression tasks to large networks with many layers that are capable of learning complex representations for different input features. These deep neural networks are used in prominent models solving complex tasks in areas such as image recognition (Krizhevsky et al., 2012) and speech processing (Hinton et al., 2012).

Common applications of ML span across computer vision (e.g. object detection in images (Kaur and Singh, 2023)), natural language processing (e.g. machine translation (Lopez, 2008) or speech recognition (Prabhavalkar et al., 2024)), robotics (e.g. learning control policies (Kaelbling et al., 1996)), finance (e.g. credit scoring (Hand and Henley, 2007) or algorithmic trading (Czuba, 2023)), and many other domains. Generally, ML can be used in any domain that provides sufficient data to identify patterns and possibly use these patterns for predictive tasks.

2.2 Natural Language Processing

Natural Language Processing (NLP) is a research field combining linguistics with machine learning. Over the years, NLP has changed from only relying on linguistic features to also using statistical models and neural networks (Zhang and Shafiq, 2024). The scope of NLP ranges from low-level processing tasks, such as splitting a text into words, to high-level understanding tasks, such as comprehending the meaning or sentiment of an excerpt of text (Jurafsky and Martin, 2025). While there is a wide range of NLP tasks, some of the core NLP tasks include fundamental text processing steps, such as *tokenization* (the process of breaking text into smaller units such as words or subwords, (Jurafsky and Martin, 2025, Chapter 2)) or *part-of-speech* (assigning labels to words indicating their grammatical role in the sentence, (Jurafsky and Martin, 2025, Chapter 17)). Some of the more complex tasks regarded by NLP are sentiment analysis (determining the emotional tone or opinion expressed through a text (Pang and Lee, 2008)) as well as machine translation or language generation. Two more sub-tasks of NLP that play a crucial role in this thesis are Information Retrieval (IR), specifically Argument retrieval, as well as Author Profiling.

2.2.1 Information Retrieval

As the name suggests, IR tasks express the aim to retrieve relevant information concerning a given queue, such as a search query or a question, from a large corpus of data. A well-known example of an information retrieval process happens for each Google search query, where the words put in by the user are the search query for the system, which then tries to retrieve relevant web links from the large database of web pages.

Modern information retrieval (IR) often proceeds in at least two steps. First, a large collection of candidate documents is indexed. In this initial stage, the system estimates the relevance of each document to a given query. Two broad approaches are used for this estimation: *sparse retrieval methods* and *dense retrieval methods*.

Sparse retrieval methods analyze documents by comparing individual terms or words. In the simplest *bag-of-words* (BOW) model, a query and a document are compared solely based on the frequency of the words they contain. Many extensions exist where words are weighted differently based on their importance for the document's meaning.

However, simple methods like BM25 face challenges from polysemy (multiple meanings for a single word) and synonymy (different words with similar meaning) (Hambarde and Proença, 2023).

In contrast, *dense retrieval* methods first encode queries and documents into low-dimensional vector representations using a neural network. Similarity between a query and a document is then determined by the distance between their vector embeddings. Dual-encoder architectures are commonly used for this purpose: a single neural network is applied to both queries and documents during training, with the objective of minimizing the distance between semantically related pairs. Dense Passage Retrieval (Karpukhin et al., 2020) is a prominent example of this approach. Models like Sentence-BERT (see section 2.2.3) are built on dual-encoder architectures and are pre-trained on semantic textual similarity tasks, making their generated embeddings particularly suitable for dense retrieval.

In practice, a two-step pipeline has become standard to address the trade-off between computational efficiency and precision. In the first step, either sparse or dense methods are used to quickly narrow down the vast document collection to a smaller candidate set. This initial retrieval can be less precise, but it greatly reduces the number of documents that must be processed further. In the second step, a more computationally intensive re-ranking method is applied to this candidate set. The re-ranking step uses additional features or more advanced models to improve the final ordering of documents. This multi-step approach is particularly beneficial when dealing with very large datasets or long documents, which may be challenging for dense vector representations alone.

Argument retrieval extends the goals of *information retrieval* aiming not only to retrieve information relevant to a query, but focusing on extracting argumentative units - such as claims, premises, or conclusions - that are not only relevant based on their content but also contain reasoning or are of a persuasive nature (Wachsmuth et al., 2017; Habernal and Gurevych, 2017). Additional challenges in argument retrieval are for instance the consideration of different stances and, in the context of this thesis, the socio-demographic relevance of arguments. Modern approaches often use a two-step process: In the first step, documents are retrieved based on their content using standard IR techniques, and in the second step, more sophisticated methods are applied that focus on identifying less obvious features such as the stance (Dang et al., 2013).

2.2.2 Author Profiling

Another critical subtask in this work is **author profiling**, which involves inferring demographic characteristics (e.g., age or gender) from an author's writing style and content (Chen et al., 2024). Author profiling leverages both lexical features and advanced machine learning models to identify socio-linguistic patterns that correlate with demographic attributes. The demographic relevance prediction investigated in this thesis aims to identify arguments that match a given socio-demographic feature and is therefore an author profiling task that is intertwined with the argument retrieval task investigated.

A common approach to author profiling involves a two-stage pipeline. Initially, various feature scores (such as lexical items, syntactic patterns, and stylistic markers like punctuation or word usage) are extracted from the text. These features are then fed into a supervised classifier (like a logistic regression or a neural network model) that has been trained on a labeled dataset where the authors' demographic attributes are known (Chen et al., 2024; Rangel Pardo et al., 2015). This method aims to estimate demographic characteristics from an author's writing style and content. Thereby, it predicts attributes such as age or gender.

In contrast, the approach adopted in this thesis does not attempt to predict the author's demographic profile. Instead, the focus lies on predicting whether a given argument matches a specified socio-demographic property. This approach bypasses the two-step process of first inferring the author's profile and then matching it to the query. This direct prediction strategy faces challenges similar to those of the traditional author profiling approach, as both methods must handle noisy stylistic features and differences between data domains.

2.2.3 Large Language Models

Modern advances in NLP are largely driven by neural networks, and especially by the *Transformer architecture* introduced by Vaswani et al. (2017). Transformers replaced earlier sequential models (e.g., RNNs) by using self-attention mechanisms that allow them to capture long-range dependencies in text efficiently. Following this breakthrough, first *Large Language Models* (LLMs) were developed. An LLM is an extremely large neural network pretrained on massive corpora of data that can be fine-tuned for various NLP tasks (Brown et al., 2020).

LLMs not only generate coherent text but also serve as powerful feature extractors for retrieval tasks. In this work, two specific models are extremely important: *Sentence-BERT (SBERT)* and *Mixtral 8x7B*. SBERT(Reimers and Gurevych, 2019) modifies BERT by using a siamese network structure to produce high-quality sentence embeddings. These embeddings capture semantic similarity, enabling to perform fast, efficient retrieval by computing the cosine similarity between query and argument representations. In information retrieval these capabilities are extremely useful, since arguments may be similar to a query even if they use very different wording.

Mixtral-8x7b (Jiang et al., 2024) on the other hand is a generative model. It is built using a sparse Mixture-of-Experts (MoE) architecture. An MoE model is comprised of multiple smaller models (experts), where each expert is finetuned on a different set of tasks. Mixtral-8x7b is comprised of eight Mistral-7b models (Jiang et al., 2023) and possesses a gating mechanism, where during the inference only two of the eight submodels are active when generating each token. With this architecture Mixtral-8x7b can achieve the effective capacity of a much larger model while maintaining lower computational cost. In addition to that, the Mixtral model achieves great performances on tasks involving multilingual understanding (Jiang et al., 2024).

2.3 Logistic Regression

While advanced LLMs are central to the research conducted in this thesis, classical statistical methods remain indispensable especially for the integration of different approaches. *Logistic regression* is a foundational technique in ML used for binary classification tasks. The logistic regression is used to predict a binary feature based on a vector of input features. To model the probability based on the input features it uses a logistic function (sigmoid function) (Dubey et al., 2022).

The logistic regression is used for binary classification tasks. It predicts the probability of an outcome by computing a weighted sum of the input features, plus a term mitigating biases - this sum is known as the logit scores. To compute the precise probability, the logit score is passed through the sigmoid function, defined as $\sigma(z) = \frac{1}{1+e^{-z}}$. During training, the model's parametres are adjusted to minimize the cross-entropy loss (Jurafsky and Martin, 2025, Chapter 5).

If the internal weights learned by the logistic regression during training are used to compute a weighted sum manually, without using the bias-mitigating term and passing the result through the sigmoid function, the result can not be interpreted as a probability. However, for a ranking task, ordering the arguments on this weighted sum will result in the same results as ordering them on the actual logistic regression outputs would have. If the logistic regression is used to compute weights that are in turn used for a ranking task in the described fashion, this procedure can simplify and speed up the computation of said rankings.

2.4 Evaluation Metrics

When evaluating information retrieval tasks, a range of metrics can be employed to assess both the relevance and diversity of the retrieved results. In ranking tasks, these metrics are typically computed at specific cutoff points because a ranked list does not inherently provide a clear threshold for relevance. Instead, relevance must be determined by considering the top-k results that users are most likely to examine (Manning et al., 2008, Chapter 8).

2.4.1 Relevance Metrics

In the context of this thesis, relevance metrics are all metrics evaluating the arguments retrieved for a set of queries based on the relevance alone, without any regards to diversity of the predictions. For a perspective argument retrieval task as investigated in this thesis, arguments can be relevant or irrelevant based on their content, but also based on demographic aspects if demanded in the query. What arguments are considered relevant is defined separately for each experiment carried out in this thesis. This section will explain all three relevance metrics used to evaluate the effectiveness of different approaches throughout this thesis.

Precision

Precision is a performance metric predominantly used in the evaluation of classification tasks. In simple terms, precision tells us how many instances predicted as true are actually correct. Mathematically it is defined as $\frac{True\,Positives}{True\,Positives + False\,Positives}$. It is worth noting, that precision does not say anything about instances predicted negative, which may limit its informative value.

For the scenarios investigated in this thesis, the precision shows the ratio of how many arguments among the top-k are actually relevant. The precision metric not taking into account any arguments rank below the cutoff at k is not necessarily a problem, as in real-world scenarios usually only a handful of arguments, here represented by k, are useful. Precision is a fairly intuitive metric, as the formula is very simple, which simplifies the interpretation. An obvious downside of using precision to evaluate the ranking-based answers provided to each scenario in this thesis is, that all k arguments are taken into account with equal weights, which does not reward solutions, that produce a better order within each of the top-k rankings. By interpreting the precision for multiple different ks this becomes less of an issue.

Recall

Like the precision, the recall is a performance metric often used when evaluating classification tasks. Put simply, the recall expresses the fraction of all positive instances, that have actually been found. Mathematically it is defined as $\frac{True\,Positives}{True\,Positives + False\,Negatives}$. The recall thereby disregards the amount of false positive arguments completely. Therefore it is reasonable to look at both recall and precision at the same time, as both metrics focus on different kinds of errors, hence a trade-off between the two is usually necessary.

For the scenarios investigated in this thesis, the recall shows the fraction of all relevant arguments, that are predicted in the top-k arguments of a methods output. As in real-world scenarios usually only a handful of arguments will be relevant, and because in all scenarios investigated in this thesis, the maximum number of arguments taken into account is capped at max(k) = 20, any method with perfect precision results would be a perfect method for the task given. Hence, the recall value is useful only to help analyze possible opportunities for improvement in methods, that do not achieve perfect precision in all scenarios. The number of predicted positive arguments being fixed at k with no threshold involved in finding a cutoff point, the recall value for any method, is already determined given the precision. This makes a comparison of methods based on recall redundant if that already takes place based on precision. The recall values will still be used in evaluation to locate opportunities for further improving methods.

Normalized Discounted Cumulative Gain

Normalized Discounted Cumulative Gain (NDCG) is a performance metric commonly used to evaluate ranking tasks. The NDCG is based on DCG (Discounted Cumulative Gain). The DCG, in simple terms, is a weighted sum of relevance scores of the arguments in a ranking. Thereby the weights are based on the positions of each argument in the ranking, with lower-ranked arguments being discounted more. To determine the DCG, a gold relevance score determining the ground truth relevance of an instance, is required. Mathematically, it is defined as $\sum_{i=1}^{n} \frac{rel_i}{\log_2(i+1)}$, with i being each rank of argument, n being the overall number of arguments in question and rel_i being the ground truth relevance of the ith argument. After determining DCG, NDCG is defined as $\frac{DCG}{IDCG}$, where IDCG (Ideal Discounted Cumulative Discount) is the best possible DCG value for a scenario, determined by computing the DCG for a ranking of instances, that represents a descending order of relevance scores.

For the task scenarios investigated in this thesis, relevance scores are defined as 1 if an argument is relevant to the query and matches its demographic property, and 0 in all other cases. While the NDCG is a ranking-based metric and could be invoked to evaluate the entire ranking of arguments, it is still only computed for the top-k arguments respectively. This saves a lot of computing cost, and, as aforementioned, in most real-world scenarios, only a handful of arguments will be useful. While the NDCG for the scenarios investigated in this thesis is based on both the number of relevant arguments in the top-k and the order of arguments within the top-k, which makes it harder to compute and thus less intuitive, the metric suits well for the ranking scenarios in this thesis.

2.4.2 Fairness and Diversity Metrics

Aside from a relevance based evaluation it can be sensible to take the distribution of different stances or demographic aspects into account to not only maximize the relevance of retrieved arguments but also the fairness and diversity of said stances or demographic aspects within the predictions. Specific goals when targeting fair and diverse retrieval results are defined by (Castillo, 2019) as follows:

A fair ranking should possess:

1. satisfactory presence of items belonging to different groups 2. a persistent treatment of similar items 3. an appropriate representation of all items, especially those belonging to minority groups

Following Pathiyan Cherumanal et al. (2021), alpha-nDCG and KL-divergence are consulted for this purpose.

KL-divergence

The Kullback-Leibler-Divergence (*KL-divergence*) is a measure of the difference between two probability distributions. In the given perspective retrieval scenario, KL-divergence is used to compare the distribution of socio-demographic properties in the prediction with the distribution of those properties in the entire corpus. Thereby, a KL-divergence score is computed for each socio-demographic attribute individually. The outcomes are also aggregated, to have an easy and quick way of comparing predictions from a diversity perspective. As suggested by (Pathiyan Cherumanal et al., 2021), KL-divergence scores are computed by focusing on one characteristic at a time. Let f_t be the relative frequency of the target group in the prediction, and let $f_{\bar{t}}$ be the relative frequency of all other characteristics combined. Furthermore, let F_t be the relative frequency of the target group in the entire corpus, while $F_{\bar{t}}$ is the combined relative frequency of all other characteristics across the entire corpus. These frequencies will be used to compute the Normalized Discounted KL-Divergence for each characteristic per socio-demographic attribute, by defining the distribution for the prediction as $P = (f_t, f_{\bar{t}})$ and the distribution for the entire corpus as $Q = (F_t, F_{\bar{t}})$. This means, that, to compute the KL-divergence score for the entire prediction, the KL-divergence is computed for each characteristic for each socio-demographic attribute first. The results are then aggregated based on the corresponding socio-demographic attribute, before finally aggregating the scores across each socio-demographic attribute to one KL-divergence score for the entire prediction.

KL-divergence is defined as:

$$D_{KL}(P\|Q) = \sum_{x \in X} p(x) log \left(\frac{p(x)}{q(x)} \right)$$

Applied to the distributions given, the formula is:

$$D_{KL}(P\|Q) = f_t * log \left(\frac{f_t}{F_t} \right) + f_{\bar{t}} * log \left(\frac{f_{\bar{t}}}{F_{\bar{t}}} \right)$$

To increase the simplicity of evaluation, the KL-divergence is normalized. In this step, the KL-divergence across different cutoff points is also aggregated. Therefore the distribution for the precision is redefined as $P_k = (f_{t_k}, f_{\bar{t}_k})$, where f_{t_k} and $f_{\bar{t}_k}$ are the relative frequencies of the protected group and all other groups combined among the top-k results from the prediction respectively. The Normalized Discounted KL-divergence is computed as (Pathiyan Cherumanal et al., 2021):

$$rKL@k = \sum_{i=1}^{k} \frac{D_{KL}(P_k||Q)}{log_2(i+1)}$$

As mentioned above, this formula is used to calculate the KL-divergence score for each characteristic for each socio-demographic attribute given in the corpus. To aggregate the scores per attribute, the $rKL@k$ values for each characteristic are summed up and divided by the number of characteristics. To compute the overall KL-divergence score per prediction, the aggregated KL-divergence scores per attribute are summed up and divided by the total number of attributes.

alpha-NDCG

The alpha-nDCG score can be described as a normalized DCG score, that punishes multiple occurrences of the same socio-demographic characteristics, by discounting the relevance score of the corresponding argument. As introduced in (Clarke et al., 2008), alpha-nDCG is computed similarly to nDCG, the only difference being the discount factor of $(1-\alpha)^{r_{i,k-1}}$, where $r_{i,k-1}$ refers to a socio-demographic characteristic of the ith argument, marking the number of occurrences of this characteristics among the previous arguments. The entire definition of alpha-nDCG based on (Clarke et al., 2008) is:

$$\alpha\text{-}DCG = \sum_{i=1}^{n} \frac{rel_i}{\log_2(i+1)} * (1-\alpha)^{r_{i,k-1}}$$

Following Clarke et al. (2008), repetitive socio-demographic characteristics are punished with the square of the number of their respective occurrences.

In this thesis, a simpler version of alpha-nDCG is adopted, that applies a constant discount to recurrent socio-demographic characteristics. Specifically, the first time a socio-demographic perspective occurs, it is not discounted at all, while for every other occurrence, the discounted relevance score is defined as $r * (1-\alpha)$, where r is the relevance score for the respective query-argument pairing and alpha is fixed at $\alpha = 0.5$. Discounting recurring perspectives with a constant factor opposed to using a quadratic function does not punish several appearances of the same characteristic too harshly. This is sensible, as the composition of the argument corpus itself does not guarantee diversity across all socio-demographic properties, which might favor a strong prevalence of some characteristics in all predictions - in this case, using a quadratic discount, relevance scores might become unrecognizably small.

While using a suitable algorithm, alpha-nDCG can be computed for all characteristics of one socio-demographic attribute at once, it is still computed separately for each socio-demographic property given in the corpus. The alpha-nDCG score to evaluate one entire prediction is gained by aggregating the scores for each socio-demographic attribute and computing the arithmetic mean.

3
Related Work

In the previous section, the necessary background for this thesis was outlined. This section focuses on research in information retrieval and specifically argument retrieval. Since this thesis builds on the dataset from the Perspective Argument Retrieval shared task (Falk et al., 2024), it is sensible to examine submissions that achieved strong results on said shared task. Among these, Twente-BMS-NLP ranked first and GESIS-DSM ranked third, both consistently outperforming baseline models. Their approaches provide valuable insights into improving argument retrieval. Additionally, research leveraging large language models (LLMs) for retrieval tasks presents an alternative perspective on enhancing retrieval quality using LLMs. The following sections first explore LLM-based approaches, followed by methods explicitly developed for demographic-aware argument retrieval as investigated in this thesis.

3.1 LLMs in Information Retrieval

3.1.1 Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting

Qin et al. (2023) investigate the challenge of using large language models (LLMs) for document ranking, an area where traditional fine-tuned ranking models have historically outperformed off-the-shelf LLMs. They argue that existing pointwise and listwise ranking formulations do not align well with how LLMs process ranking tasks, leading to suboptimal performance. To address this, they introduce Pairwise Ranking Prompting (PRP), a method that reduces task complexity by presenting the model with two candidate documents at a time and asking it to determine which is more relevant. This approach allows LLMs to make relative comparisons rather than requiring absolute relevance scores, which are difficult to calibrate. Their experiments demonstrate that PRP achieves state-of-the-art ranking performance using moderate-sized open-source LLMs, even surpassing some black-box commercial models like GPT-4 on certain benchmarks. The study highlights the potential of efficient prompting strategies for improving LLM-based retrieval while maintaining scalability and cost-effectiveness.

3.1.2 Investigating Large Language Models as Re-Ranking Agents

Sun et al. (2023) explore the role of large language models (LLMs), particularly ChatGPT and GPT-4, as passage re-ranking agents in information retrieval (IR) systems. Traditionally, IR models have relied on manually supervised methods, which require extensive human effort and struggle with generalization. Sun et al. investigate whether LLMs can perform passage re-ranking without fine-tuning, purely through prompting.

Their experiments reveal that when properly instructed, ChatGPT and GPT-4 achieve competitive and even superior performance compared to state-of-the-art supervised re-ranking methods. They introduce an instructional permutation generation method, which prompts the LLMs to directly rank passages rather than assigning independent relevance scores. Additionally, they propose a sliding window strategy to overcome token length limitations, allowing LLMs to process and rank longer lists of passages more effectively.

Recognizing the high computational cost of deploying LLMs in real-world search systems, they also investigate a permutation distillation approach. This technique distills the ranking capabilities of ChatGPT into smaller, specialized models, reducing costs while maintaining strong performance. They find that a distilled 440M parameter model outperforms a 3B supervised model on the BEIR benchmark.

3.2 Demographic-Aware Argument Retrieval

3.2.1 Twente-BMS-NLP: Combining Bi-Encoder and Cross-Encoder for Argument Retrieval

Zhang and Braun (2024) use a hybrid argument retrieval approach, combining a bi-encoder (paraphrase-multilingual-mpnet-base-v2) to retrieve the top 1000 candidates with a cross-encoder (ms-marco-MiniLM-L-12-v2) to re-rank the top 50. They find that monolingual models perform better, so they translate arguments into English before re-ranking.

For socio-demographic filtering, they apply explicit matching when labels are available and prediction when they are not. Predicted features use sentence embeddings, token length distributions, POS n-grams, and stop-word distributions. Some categories, like residence, are easier to predict due to data imbalance, while others, like political stance or education, are much harder. Gender-based filtering reduces performance, while filtering by important issues improves it.

Their findings show that hybrid models outperform bi-encoders alone, but demographic-aware filtering remains difficult. Inferred labels are often unreliable, and argument length limits feature extraction. Future improvements may require longer texts, better prediction models, or alternative diversity strategies beyond strict filtering.

3.2.2 GESIS-DSM: Socio-Cultural Differences in Argumentation

Maurer et al. (2024) explore whether socio-cultural differences in argumentation are primarily a matter of content or style. Their system follows a three-step pipeline: first, filtering arguments by explicit demographic labels when available; second, ranking arguments based on semantic similarity using Sentence-BERT; and third, selecting final arguments either through stylistic classification or re-ranking based on arguments generated by an LLM.

Their analysis finds little semantic differentiation between socio-cultural groups, as clustering Sentence-BERT embeddings does not align with demographic attributes. Instead, they identify measurable stylistic differences, such as variations in sentence complexity, vocabulary diversity, and pronoun usage. Their best-performing model classifies arguments based on these stylistic features using a random forest classifier.

Their results show that stylistic classification significantly improves performance over a Sentence-BERT baseline, particularly when demographic information must be inferred. However, re-ranking using LLM-generated arguments is less effective. The findings suggest that argument style plays a crucial role in distinguishing socio-cultural groups, though challenges remain in balancing accuracy and diversity in retrieval.

4 Methodology

Now that the research questions are defined and important background information are shown, the different methods and approaches tested in this thesis will be explained thoroughly. Firstly, the dataset functioning as a basis for the research will be introduced before the methodology for each research question established in section 1 is explained in detail. These descriptions include the experiments targeting the research question, the different feature scores that are derived from the data, the pipeline used to retrieve arguments, and the intended evaluation focusing on solving the respective research question.

4.1 Dataset

This thesis is grounded on a dataset of political questions and arguments, which is based on arguments from the Swiss voting recommendation platform Smartvote[1]. It contains political questions and arguments raised during the 2019 and the 2023 Swiss elections. The dataset was raised in the context of the Perspective Argument Retrieval Shared Task 2024 by Falk et al. (2024). The dataset is multilingual and contains German, French, and Italian languages.

The dataset consists of a corpus of arguments that only slightly varies for each set in the split and a set of queries that is distinctly different for each set in the split. The corpus contains 21 features in total, of which only three are relevant for further investigation. These features are the argument, the political topic, and the demographic profile of the argument's author. The remaining features are utility features, such as IDs or different representations of the same feature.

The query set contains only two features, that are relevant for further investigation, namely the query, a political question, and a social or demographic property that matches the socio-demographic profile of the authors of arguments relevant to this query.

1. www.smartvote.sh/

4.2 Research Question 1

To address the research question "*How can LLMs be used to enhance the result of an argument retrieval process?*", two experiments are conducted to evaluate the effectiveness of methods, where LLMs support argument retrieval. Both involve processing a set of queries against a set of political arguments from the aforementioned dataset.

Thereby, the first experiment includes both the socio-demographic property given for each query and the socio-demographic profile given for each argument. This enables a strict filtering process, retaining only arguments explicitly relevant to the specified socio-demographic property. As a result, the set of potentially relevant arguments is significantly reduced for each query. Further retrieval steps, effectively performed after this filtering, are based solely on semantic relevance but operate on a substantially narrowed corpus for each query.

The second experiment does not ask for any socio-demographic property with each query and does not include any socio-demographic information about the author of each argument. The scope of this task is therefore the prediction of all semantically relevant arguments from the entire corpus of arguments.

Both experiments aim to determine the extent to which LLMs can improve the relevance and quality of retrieved arguments compared to traditional retrieval methods in a standard argument retrieval scenario. Therefore, both experiments isolate the investigation of methods to predict the semantic relevance between arguments and queries without regard to any socio-demographic aspects. The two experiments are distinguished primarily by the scope of the dataset used. The first experiment only considers the subset of arguments that are relevant to a socio-demographic property given in each query. The second experiment on the other hand considers the entire corpus of arguments for each query. As both experiments aim to address the same research objective, the methodology across the two experiments remains largely consistent. However, analyzing the methods from these two different scopes enables an investigation of their strengths in varying scenarios.

The following sections detail the methodology used for the first experiment, followed by the methodology used for the second experiment, and conclude with an outline of the intended evaluation for both experiments.

4.2.1 Experiment 1

The intended pipeline for the first experiment follows a three-step process, the first step being the filtering based on socio-demographic matches between a query and the arguments as described above. The following two steps orient towards a standard two-step retrieval pipeline (as described in section 2.2.1) with an initial retrieval step followed by a reranking step. As LLM inference is costly in terms of computation, in this thesis, LLM usage is only investigated in the smaller reranking task. The entire intended pipeline is shown in graphic 4.1.

The approach chosen for the realization of each step of the pipeline is the calculation of different feature scores. These assign a relevance score to each query-argument pairing that is calculated based on the respective feature. With these feature scores, the three-step pipeline can easily be fulfilled. At the same time, the format allows for an easy comparison of different feature scores across all the experiments conducted in this thesis. The final pipeline for this experiment uses four different feature scores, namely

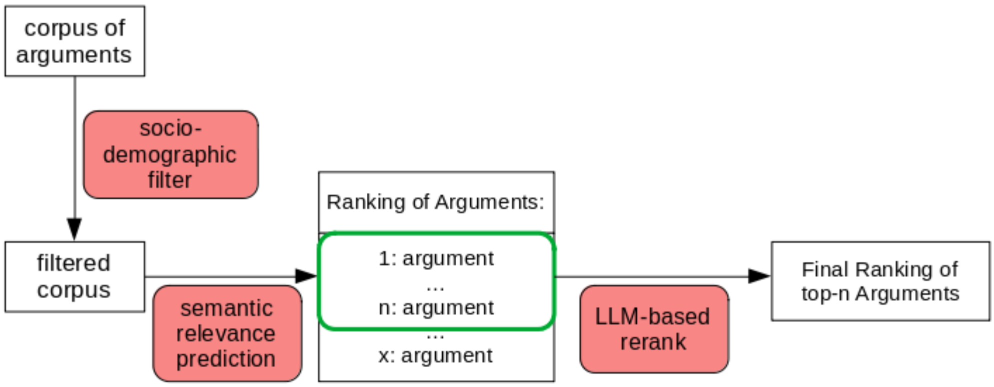


Figure 4.1: Intended pipeline for experiment 1

the *explicit demographic scores* for socio-demographic filtering, the *SBERT relevance scores* for an initial relevance prediction in the retrieval step, the topic scores to enhance the performance of the initial retrieval step, and the *LLM relevance scores* reranking the top-n retrieved arguments. The following sections provide an overview of these feature scores explaining all investigations involved with each score, where a special focus lies in the investigation of the LLM feature scores.

Explicit Demographic Scores

The explicit demographic scores match the explicitly given demographic profile for each argument's author with the demographic property asked in the query. For a given query, an explicit demographic score of 1 is assigned to each argument, whose author matches the property given in the query, while a score of 0 is assigned to every other argument. The explicit demographic scores could be used for strict filtering, where only arguments with a score of 1 are included in the predictions. However, they also match the format of predicted Demographic LLM scores that are introduced in section 4.3.1, which enables the creation of a uniform pipeline and also simplifies some evaluation steps, as the different demographic relevance scores are easily comparable.

SBERT Relevance Scores

Siamese BERT Networks, presented as sentence-BERT or SBERT models in (Reimers and Gurevych, 2019), have been shown to produce high-quality dense embeddings on a sentence level. In standard dense retrieval fashion, a relevance-based ranking, following the Probability Ranking Principle for a given query, can be derived, by embedding the query, along with all arguments in the corpus, using SBERT, by calculating the cosine similarity between the query embedding and each of the argument embeddings, and then ordering the arguments based on descending similarity (Luan et al., 2021; Karpukhin et al., 2020). By doing this for every given query, a prediction for an entire set of queries can be created.

As the dataset used in this thesis is multilingual, either the SBERT model needs to be trained on multilingual data, or another translation step is required. In this thesis, paraphrase-multilingual-mpnet-base-v2[2] is used to calculate the dense embedding vectors. The model is trained on a total of 53 languages, including French, Italian, and German, and is therefore suitable as an embedder for the given dataset.

Topic Scores

As mentioned earlier, the dataset serving as a basis for this thesis assigns a political topic to each argument. The topic scores are based on the assumption, that the queries can be classified into the same set of topics, that the entire set of arguments is also classified into. This would mean that all relevant arguments for one query are classified into the same topic. If a query's topic can be correctly identified, this enables filtering out all arguments that do not match that topic while performing the initial retrieval step. For the dataset this thesis is based on the assumption turns out to be true. In the following, five different approaches to calculate relevance scores based on the predicted topic for a query are further explained. Each of these assigns one score between 0 and 1 to each topic for a given query. For a given query, the scores for all topics add up to 1, hence the topic scores can be interpreted as an estimated probability for a topic to be relevant to the query. Ultimately, for every query-argument pair a topic score is calculated. For simplicity, the following descriptions will explain the topic score computations between a single given query and all arguments in the corpus. Generally, this step will simply be repeated for every single topic in the respective data split to calculate the topic scores for the entirety of the data.

Binary Topic Scores. Binary topic scores most accurately follow the ideas described above. In this thesis, two different methods are explored, and both seek to predict the most fitting topic for a query. One of these methods is based on a prediction using a Large Language Model, the other approach is based on the topic distribution among the most relevant arguments for a query.

To predict the topic of a given query using an LLM, a prompt is created that includes the query, a list of all topics in question, and a demand, to return the topic most fitting the query. The prompt used can be found in Appendix A.1.5. The list of topics is thereby the set of all topics that can be found in the corpus of arguments in no distinct order.

The prediction of a query's topic using the topic distribution among the most relevant arguments uses the SBERT cosine similarity scores to calculate an initial ranking of the arguments for the given query. The query's topic is then simply predicted as the one that appears most frequently among the top 50 arguments from that ranking.

To form topic scores as described above for a given query, all arguments matching the query's predicted topic get a relevance score of 1. All arguments assigned to a topic that doesn't match the query's predicted topic get a relevance score of 0. These scores can then be used to effectively filter out all arguments not matching the topic but they can also be combined with other relevance scores for example by using a Logistic Regression as described in section 4.4.

Relative Topic Scores. The relative topic scores use the idea of predicting topics from the most relevant arguments for a given query to produce more differentiated relevance scores. This is done by taking one step away from the initial idea of predicting a given query's topic and filtering out all arguments that do not match. Instead, the

2. https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2/blob/main/README.md

aim is to develop topic scores that more accurately reflect the probability of each topic being relevant to the query, using a continuous range between 0 and 1.

Similar to the binary topic scores, the relative topic scores are based on the top 50 most relevant arguments to a given query based on SBERT relevance scores. The score for each topic is thereby determined as the number of occurrences of said topic among the top 50 arguments divided by 50. The main upside compared to binary topic scores is the possibility to not only mark the topic that is most likely to be relevant but instead differentiate topics, that are likely to be relevant, by also assigning topic scores other than zero to topics, that appear less often in the top ranking arguments based on SBERT.

Hyperbolical Weighted Topic Scores. The hyperbolical weighted scores try to predict more accurate probabilities for each topic being relevant to the given query, by considering not only the count of each topic among the top 50 most relevant arguments based on SBERT but also their respective positions within the ranking. Thereby, the topics of high-ranking arguments are considered to be more relevant for predicting the query's topic.

The hyperbolical weighted scores for one topic for a given query are based on the positions of the said topic in the ranking of arguments based on SBERT cosine similarities. Let $x_1, x_2, ..., x_m$ be the positions of a topic t in the ranking. Then, the hyperbolical weighted topic scores for topic t are computed as follows:

$$topic\_score(t) = \frac{\sum_{i=1}^{m} \frac{1}{x_i}}{\sum_{i=1}^{50} \frac{1}{i}}$$

Thereby, the term $\sum_{i=1}^{m} \frac{1}{x_i}$ adds the reciprocal ranks of all arguments that match the topic t, while the term $\sum_{i=1}^{50} \frac{1}{i}$ serves to normalize the scores.

Linear Weighted Topic Scores. Just like the hyperbolical weighted topic scores, the linear weighted topic scores use descending weights based on the positions of arguments in the ranking based on SBERT scores, when computing the score for the corresponding topics. As suggested in the name, the linear weighted topic scores use a linear function to determine the weight for each position in the ranking instead of a hyperbolical one.

To compute the linear weighted topic scores, let $x_1, x_2, ..., x_m$ be the positions of a topic t in the ranking of arguments based on SBERT cosine similarities. Then the scores are computed as follows:

$$topic\_score(t) = \frac{\sum_{i=1}^{m} \frac{50+1-x_i}{50}}{\sum_{i=1}^{50} \frac{50+1-i}{50}}$$

Here, the term $\sum_{i=1}^{m} \frac{50+1-x_i}{50}$ assigns a weight to each rank x_i reaching from 1 to 0.02 and descending linearly. The term $\sum_{i=1}^{50} \frac{50+1-i}{50}$ serves to normalize the scores, so that they all add up to 1, which simplifies comparing the different topic scores.

With this approach, the impact, that arguments have on the topic score decreases less drastically based on their position in the ranking, which could lead to more accurate estimates if one of the highest-ranked arguments doesn't match the expected topic.

Similarity Weighted Topic Scores. The similarity-weighted topic scores use the top 50 arguments predicted using SBERT cosine similarities while using these exact cosine similarities to weigh each corresponding topic when computing topic scores. To compute the similarity-weighted topic scores, let $x_1, x_2, ..., x_m$ be the positions of a topic in the ranking and let $S(x)$ be the similarity score of the argument in position x in the ranking. The scores are then defined as

$$topic\_score(t) = \frac{\sum_{i=1}^{m} S(x_i)}{\sum_{i=1}^{50} S(i)}$$

where the term $\sum_{i=1}^{m} S(x_i)$ sums up the similarity scores for all arguments matching topic t, while the term $\sum_{i=1}^{50} S(i)$ sums the similarity scores for all arguments in the top 50 to normalize the resulting topic scores.

LLM Relevance Scores

Large Language Models gain knowledge from being trained on large amounts of data. This knowledge can be used, to assess the relevance of arguments to a given query. Due to the high computational cost, the usage of LLMs during the initial retrieval step is not investigated further in this thesis. Instead, the investigation focuses on the usage of LLMs during the reranking step of the highest-ranking arguments from the initial retrieval step. Zhu et al. (2024) distinguish three different options - pointwise, pairwise, and listwise approaches - to use decoder-only models in a reranking step to enhance argument retrieval performance. Because pointwise methods, utilizing likelihoods as scores for the relevance between a query and an argument, are significantly outperformed by both pairwise and listwise approaches, they are not considered in this thesis. As pairwise approaches require a significantly higher number of prompts to rerank retrieved arguments for each query while yielding similarly promising results as listwise methods, these were also not investigated further. Instead, this thesis focuses solely on the adoption of listwise reranking methods based on LLMs to improve an initially retrieved ranking.

In the following, multiple methods are described, all of which prompt an LLM to (re-)order a set of arguments based on their relevance to a query. The prompt includes the query, the set of arguments, and the exact task for the model to perform. The different approaches vary in the composition of the prompt. Tested are various numbers of arguments in each query and several different tasks or task formulations for the LLM to perform. All of these prompting approaches are described in the following sections.

Task Mode Two different task modes are tested, both use an LLM to predict the relevance of arguments to a query, which differ in the style of answer expected. Other information retrieval systems that use LLMs for reranking, expect a ranking of the arguments as an answer (Sun et al., 2023). Such a ranking can be machine-readable, enabling an easy integration in the pipeline. It also fits the mode of the experiment, as all methods are evaluated based on their ranking ability. In this thesis, this ranking approach is investigated. Therefore, an LLM is prompted with the query, a list of the top-n highest-ranked arguments based on the initial retrieval step and the demand, to return a ranking of the arguments based on their relevance to the given query. The prompt can be found in Appendix A.1.1. The ranking returned by the LLM is interpreted as the final ranking of the top-n arguments, overruling all other feature scores used during the initial retrieval. This approach will later be referred to as the *direct reranking approach.*

While the list-based reranking approach is simple yet effective, it can not describe two arguments having the same relevance, e.g. in cases of uncertainty. At the same time, there is no measure for certainty, nor for the difference in relevance between different positions in the ranking. The feature scores used during the initial retrieval step can not only be used as a ranking by ordering arguments in a descending order based on the corresponding feature scores, they can also be used to assume the level of certainty within that ranking as the size of the gap between the feature scores of two arguments, where a bigger gap can be interpreted as higher certainty. This is especially relevant when combining multiple feature scores in one prediction because not all feature scores will yield the same ranking.

Trying to transfer these advantages to a reranking approach based on LLMs, an LLM scoring approach is tested. Here, an LLM is prompted with a query, a list of the top-n highest-ranked arguments based on the initial retrieval step and the demand, to return a score for each of the arguments, measuring the relevance between said argument and the given query. These scores can be used to predict a ranking immediately, and they offer more options to be combined with other feature scores from the retrieval step. The score-based approach will be referred to as the *score-based reranking approach*.

Window Size Large computational costs and the maximum context size of LLMs are limitations when it comes to performing a ranking task with an LLM, which is why in this thesis LLMs are only used to rerank the top-n highest-ranking arguments for each query, as opposed to ranking the entire corpus of arguments each time. To determine the ideal window of arguments for the reranking step, different window sizes are tested. These window sizes are 20, 30, 40, 50, and 100. With roughly 80 tokens per argument, all of these are well below the maximum context size of the LLM used throughout this thesis, mixtral8x7b (Jiang et al., 2024), which was trained on a context window of 32,000 tokens.

A larger context window naturally includes a bigger absolute number of relevant arguments and thus more room for improvement for an LLM, however, the longer prompts resulting from an increased context window might also be more complex for the LLM to process. To bypass this potential pitfall, RankGPT (Sun et al., 2023) suggest a sliding window strategy. This strategy allows to rerank a larger window of arguments by further separating it into smaller windows. Starting from the lowest ranks within the overall context window, the LLM ranks w arguments arguments based on their relevance to the query, where w is the sliding window size. The sliding window is then moved up in the overall context window by stepsize s. As long as the stepsize s is larger than the sliding window size w, this assures, that the top-ranking arguments from the lower context windows also appear in the subsequently higher windows. In this thesis, the sliding rerank with sliding window size $w = 50$ and step size $s = 20$ on an overall context window of $n = 100$ arguments is compared to the other context window sizes. The different context windows are tested on the *direct reranking approach*. The *score-based reranking approach* is always performed with $n = 50$.

Other differences in Prompting Apart from the task mode and the window size, some different formulations for otherwise similar prompts are tested. These are primarily examined on small subsets of queries with the aim of avoiding inaccuracies detected during the manual inspection of the data. Specifically, this refers to deviations from the expected response format.

4.2.2 Experiment 2

The second experiment does not include any socio-demographic information in the query, therefore it does not require any filtering of the corpus based on socio-demographic matching. Furthermore, any argument that semantically fits as an answer to the query is considered relevant. Therefore, the pipeline for this experiment closely resembles the pipeline for Experiment 1, only removing the filtering step based on demographic matching. The pipeline for Experiment 2 is shown in graphic 4.2.

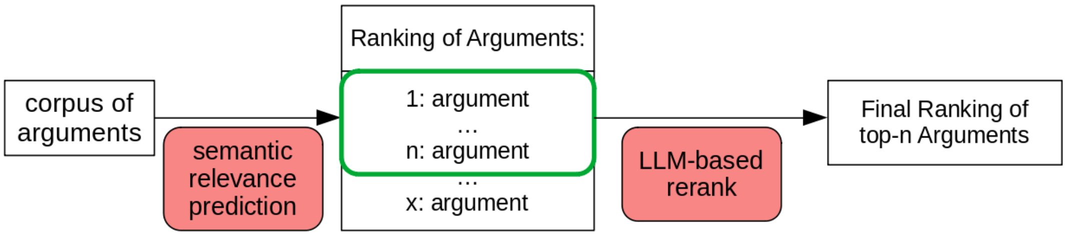


Figure 4.2: Intended Pipeline for Scenario 2

Just like the first experiment, the pipeline is realized with feature scores. For the relevance prediction in the initial retrieval step, SBERT relevance scores and topic scores are combined. The new ranking is calculated based on LLM relevance scores during the reranking step. Each feature score is computed using the method that demonstrated the best performance in *Experiment 1*.

SBERT similarity scores. The SBERT similarity scores for every query-argument pair are calculated as the cosine similarity between the query's SBERT embedding and the argument's SBERT embedding, the same way as they are calculated in *Experiment 1*.

Topic scores. Concerning the topic scores, *Experiment 1* could not show a difference in performance between the relative topic scores, the hyperbolical weighted topic scores, and the SBERT weighted topic scores. For experiment 2 the SBERT weighted topic scores are used.

LLM scores. As for the LLM scores, the best configurations from *Experiment 1* are adopted in *Experiment 2*. Therefore, the LLM scores for this experiment are conducted for a fixed window of size 50. The general approach retrieves an *LLM relevance score* for each query-argument pair. The prompt used is shown in Appendix A.1.3.

4.2.3 Comparing Experiment 1 and Experiment 2

The *second experiment* regards all arguments as relevant to a query that semantically answer the query and generally considers the entire corpus of arguments during the retrieval of arguments for each query. Because the *first experiment* only considers those arguments as relevant to a query that both match semantically as an answer to the query and fit the socio-demographic property given in the query based on the socio-demographic profile of the argument's author, the set of relevant arguments for each query is generally much smaller. At the same time, the corpus of arguments can be strictly filtered based on socio-demographic fit, because both the socio-demographic profile for the author of each argument and the socio-demographic property desired in each query are explicitly given. Therefore, the corpus of arguments regarded for each individual query is significantly smaller as well, compared to the *second experiment.*

During the *first experiment*, different parameters are tested and evaluated across multiple test splits. The second experiment does not experiment with different parameters and is solely used to evaluate performance that can be achieved when applying the best parameters found through the first experiment at a different scope in the same dataset.

4.2.4 Intended Evaluation

The evaluation for the first research question focuses on the effectiveness of the reranking process within the information retrieval pipeline. Reranking is only performed on the top-50 initially retrieved arguments, hence the evaluation targets only this part of the ranking. This ensures that the evaluation is limited to the context, where reranking methods are applied.

The focus on reranking arises from the research objective, which is targeted at the usage of LLMs to enhance argument retrieval processes. Due to the high computational expenses necessary, LLMs are exclusively employed during the reranking step and not during the initial retrieval step. Therefore, the reranking process is the primary area of investigation.

The evaluation mainly focuses on comparing the quality of rankings before and after the reranking process. This comparison highlights the extent to which LLMs are able to distinguish relevant arguments from irrelevant arguments and enables the comparison of the LLM predictions to results based on the more common information retrieval methods used in the initial retrieval step. This is vital to determine whether the LLM-based reranking step successfully improves the initially retrieved ranking of arguments.

Another key aspect of the evaluation needs to be the importance of relative positions of arguments within the ranking, where higher ranking arguments are weighted stronger during the evaluation. This reflects the objectives of real-world information retrieval or argument retrieval scenarios, where the top-ranking arguments are usually much more likely to be considered, while lower-ranking results are oftentimes not perceived by users due to the mode in which retrieved arguments are presented (e.g. popular search engines like *google* or *bing* only show the top-10 search results on the first page). Consequently, metrics used in the evaluation should especially emphasize the quality of the predictions across the highest ranks within the top-50 subset of retrieved arguments.

While the reranking step is the primary focus of the evaluation, isolating the reranking step alone may not provide a complete understanding of the system's overall effectiveness. To fully evaluate the impact of the reranking process, it is essential to consider its role within the context of the entire information retrieval pipeline. This means assessing the effectiveness of the reranking methods in a two-step retrieval system across the entire system, rather than just the improvements across the subset of the top-50 retrieved arguments. Therefore, some metrics across the entire dataset that capture the performance of the two-step ranking system need to be calculated. These can also help with the comparability of results across different research objectives within this task, or even with totally different works on the same dataset.

To compare pre- and post-reranking results, the evaluation focuses on a detailed analysis of the top-50 arguments in both rankings. Within this subset, *recall* and *precision* are calculated and compared, to measure the effectiveness of the reranking in distinguishing relevant and irrelevant arguments. Both recall and precision are computed for each rank from 1 to 50, resulting in metrics such as *recall@1*, *recall@2*, ..., *recall@50* and *precision@1*, *precision@2*, ..., *precision@50*. These help with a detailed evaluation of the quality of the ranking results from the LLM-based reranking and also show the performance across the highest ranks within the top-50, which, as aforementioned, are especially relevant in most real-world applications of information retrieval systems.

Additional metrics are computed to evaluate the overall ranking quality to assess the combined effectiveness of the retrieval and reranking pipeline as a two-step ranking system across the entire dataset. The primary metric used for this purpose is *NDCG (Normalized Discounted Cumulative Gain)*, a metric often used when evaluating ranking system, since higher-ranking arguments are weighted higher when calculating the NDCG metric. Therefore, NDCG is useful to evaluate the quality of a ranking while especially focusing on the highest-ranking arguments within said ranking. In addition to NDCG, α-NDCG is calculated to account for diversity in relevance, and precision is computed to give more inside to the quality of the ranking in terms of relevance. All of these metrics are calculated at ranks 4, 8, 16 and 20, again because in most practical information retrieval tasks, only the highest ranking arguments are actually relevant.

During evaluation, all metrics are averaged across the entire split of the dataset.

4.3 Research Question 2

This subsection will address the research question "*How can LLMs be used to implicitly predict the demographic or sociocultural perspective with only little text input?*". To do so, one experiment is conducted, in which LLMs are used in a perspective argument retrieval scenario to enhance the prediction of relevant arguments by trying to predict, how well retrieved arguments match a given socio-demographic property.

This third experiment is based on the same aforementioned dataset. Similar to *Experiment 1* described in section 4.2.1, *Experiment 3* includes a socio-demographic in each query and considers only those arguments as relevant that both have an author matching the socio-demographic property and are relevant based on their content. Contrary to *Experiment 1*, the socio-demographic profiles of each argument's author are excluded from *Experiment 3*. In an initial retrieval step based on SBERT relevance scores and topic scores as described in section 4.2.1 for each query the top 50 most relevant arguments based on their content are retrieved. An LLM is then used to predict the socio-demographic relevance of these 50 arguments based on the expected socio-demographic property and the content of the arguments. The pipeline for this third experiment, which includes the socio-demographic relevance prediction, and the intended evaluation steps are described below.

4.3.1 Experiment 3

Experiment 3 serves to evaluate the capabilities of LLMs in predicting the socio-demographic attributes of a statement's author in an argument retrieval scenario. Because of the high computational efforts when using LLMs, the experiment revolves around the reranking step in a two-step argument retrieval pipeline. The initial retrieval step is identical to the retrieval step performed in *Experiment 2* which is described in section 4.2.2, which means it is based on the *SBERT relevance scores* and the *Topic scores* introduced in *Experiment 1* 4.2.1. The reranking step in *Experiment 3* is only based on the socio-demographic relevance of arguments predicted by LLMs. Similar to experiments 1 and 2, a score format is applied, which later enables experimentation with different ways to combine all feature scores available for each experiment. The *socio-demographic LLM rerank scores* used for the reranking step in *Experiment 3* are described in the following.

Socio-demographic LLM rerank scores

Being trained on large amounts of data, Large Language Models are often exposed to texts associated with socio-demographic information about their respective authors. Thereby the LLMs might gain knowledge that subsequently could be used to predict socio-demographic attributes of a statement's author based on the content of the statement itself.

While *LLM rerank scores* are based on the relevance between arguments and a given query, the *Socio-demographic LLM rerank scores* focus on the relevance between arguments and a socio-demographic property. Due to the similarities between the two scenarios, the *Socio-demographic LLM rerank scores* are calculated in a pattern similar to the calculation of *LLM rerank scores* described in section 4.2.1. This means that the *Socio-demographic LLM rerank scores* are also based on a listwise reranking approach, where a new ranking is determined for the top 50 highest-ranking arguments. The LLM is asked to return a relevance score for each argument that reflects how relevant the argument is concerning the given socio-demographic attribute. No variations concerning the prompt to determine *Socio-demographic LLM rerank scores* are tested in this experiment.

4.3.2 Intended Evaluation

Experiment 3 investigates a scenario, where arguments are to be retrieved that match a given query and also have an author matching with a given socio-demographic attribute. Because research question 2 aims to find ways to utilize Large Language Models for predicting socio-demographic aspects of a statement's author, the socio-demographic profile of each argument's author is excluded in *Experiment 3*. While *Experiment 3* aims to utilize LLMs for socio-demographic matching in an argument retrieval scenario, the primary focus of evaluation lies in the *Socio-demographic LLM rerank scores*, which represent the socio-demographic matching between a given socio-demographic property and each of the top-50 arguments and can be used to derive a ranking of the top-50 arguments for each query based on their socio-demographic matching alone.

Socio-demographic LLM rerank scores depict only the match between arguments and the given socio-demographic property, hence the evaluation should mainly focus on socio-demographic matching in the reranking step, as opposed to the overall relevance of arguments retrieved across the entire pipeline. Additionally, it is plausible that the matching abilities of LLMs differ depending on the social or demographic feature in question. This should be further investigated during evaluation as well.

Research question 2 primarily targets the prediction of socio-demographic aspects. However, since this entire thesis is situated in an argument retrieval context, the overall argument retrieval process should also be regarded during the evaluation, to assess possible benefits of *Socio-demographic LLM rerank scores* in a perspective argument retrieval scenario.

While evaluating the *Socio-demographic LLM rerank scores* solely based on demographics, all arguments whose author matches the given socio-demographic property are seen as relevant, with no regard to the content of the argument. This is done so bad predictions from the relevance-based retrieval step do not influence the evaluation. As this evaluation step focuses on the reranking, the same metrics used to evaluate the content-based reranking in *Research Question 1* are used in this step. This means that both recall and precision are computed at every rank from 1 to 50 both before and after the reranking step.

The queries are bundled based on the socio-demographic aspect given with each query to analyze the differences in performance concerning different social and demographic properties. Then, delta-NDCG is computed for every socio-demographic aspect at ranks 1, 4, 8, and 20. Delta-NDCG@k is the NDCG@k for the ranking based on the *Socio-demographic LLM rerank scores* minus the NDCG@k based on the ranking originally retrieved averaged across queries - or in this case all queries that ask for the same category of socio-demographic attributes. Delta-NDCG is used to measure the improvement of the reranked order compared to the original ranking. By comparing delta-NDCG values for different socio-demographic attributes, possible differences in the performance of the socio-demographic-based reranking can be found.

The combined performance of the initial content-based retrieval and the demographic-based reranking is assessed through some additional metrics. As this step does not single out the socio-demographic reranking but regards the entire perspective retrieval process, only those arguments are considered relevant, that match a query based on both the socio-demographics and the content. To evaluate the performance of the combined pipeline, NDCG is computed at ranks 4, 8, 16, and 20. The baseline methods used for comparison are an SBERT baseline only using the *SBERT relevance scores* to compute the final rankings, and the predictions from the initial retrieval step, which is the combination of *SBERT relevance scores* and *Topic scores*. The NDCG results are averaged across the entire split of the dataset. Additionally, NDCG is computed for each socio-demographic feature individually, to show possible differences in the performance of the overall pipeline depending on the socio-demographic attributes asked.

4.4 Research Question 3

Research question 3 aims to combine the findings from the investigation for *Research question 1* and *Research question 2* in an argument retrieval pipeline taking into account both content relevance and socio-demographic relevance. To do so, *Experiments 1*, *2*, and *3* are further looked upon to investigate different variations of the standard two-step retrieval-reranking pipeline.

The standard two-step information retrieval pipeline consists of a retrieval step, serving to find the top-n documents for a given query, and a ranking step, ordering the top-n retrieved arguments from most to least relevant concerning the given query (Dang et al., 2013). The two-stage retrieval approach is based on the idea that low-cost and high-cost methods exist, where low-cost methods can provide a sufficiently high recall to perform an initial retrieval step, and high-cost methods can always provide a better final ranking to refine the results. With this approach, however, the information about relevance from methods used in the initial retrieval might be overlooked in the final ranking. The approach investigated in this section serves as an alternative to the standard two-step pipeline, aiming to include knowledge from all retrieval and ranking methods in the final ranking by combining all of the feature scores introduced.

To do so, a logistic regression model is trained on a train-split of the data, using all available feature scores to predict the relevance of a query-argument pair. Depending on the experiment in question, the logistic regression model will have three to four input features. A query-argument pair is labeled 1 if the argument is considered relevant to the query and 0 if it isn't. For *experiment 2*, all arguments are considered relevant, that answer the query based on their content alone. For *experiments 1* and *3*, only those arguments are considered relevant, that both match the query based on their content and are authored by someone fitting the demographic attribute provided in the query. After training the logistic regression, it can be used to derive weights for each of the feature scores. For simplicity, the internal logit scores of the regression are used as weights for a weighted sum rather than performing the inference. This simplification is less accurate when trying to predict whether an argument is relevant to a query. However, as all experiments involve ranking tasks and are only evaluated based on the final ranking, using the logit scores to compute a weighted sum will lead to the same order and thus the same results as performing the logistic regression to predict the relevance for a query-argument pair. To compute the final ranking for one experiment, all feature scores considered in the experiment are multiplied with the corresponding internal weight from the logistic regression and then added up. For each query, the arguments are then ordered in descending order based on the calculated sum.

4.4.1 Revised Pipeline for Experiment 1

Experiment 1 aims to retrieve arguments relevant to a query based on their content that also match a socio-demographic attribute given in the query based on the demographic profile given explicitly for the argument's author. Binary *explicit demographic scores* are computed for every query-argument pair to determine the socio-demographic matching. The relevance based on content for each pair is initially predicted with *SBERT similarity scores* and *topic scores*. These three scores are used to create an initial ranking of the arguments for each query, which is in turn used, to determine the top 50 most relevant arguments for each query to include them in the LLM prompt. The LLM then returns the *LLM relevance scores* 4.3.

Using the train-split of the data, a logistic regression is trained. As an input, it regards the *explicit demographic score*, the *SBERT similarity score*, the *topic score*, and the *LLM relevance score* for a query argument pair. The expected output is 0 if either the argument does not match the query based on the content or if it does not match the socio-demographic attribute given. The expected outcome is 1 only if it matches based on both content and the demographic feature.

After the logistic regression is trained, the internal weights for each of the feature scores are retrieved and used as weights to compute a weighted sum of all feature scores. To predict the best arguments for any other data split, all of the four feature scores mentioned above are combined as a weighted sum using the logit scores as weights. For each query, the arguments are sorted in descending order based on the weighted sum 4.4).

4.4.2 Revised Pipeline for Experiment 2

Experiment 2 does not include a socio-demographic property in the query and therefore aims to retrieve arguments focusing solely on content-based relevance. It therefore does not include any kind of demographic scores. Instead, *SBERT similarity scores* are computed and *Topic scores* are used to compute an initial ranking, from which the top 50 arguments for each query are scored by an LLM, resulting in the *LLM relevance scores*.

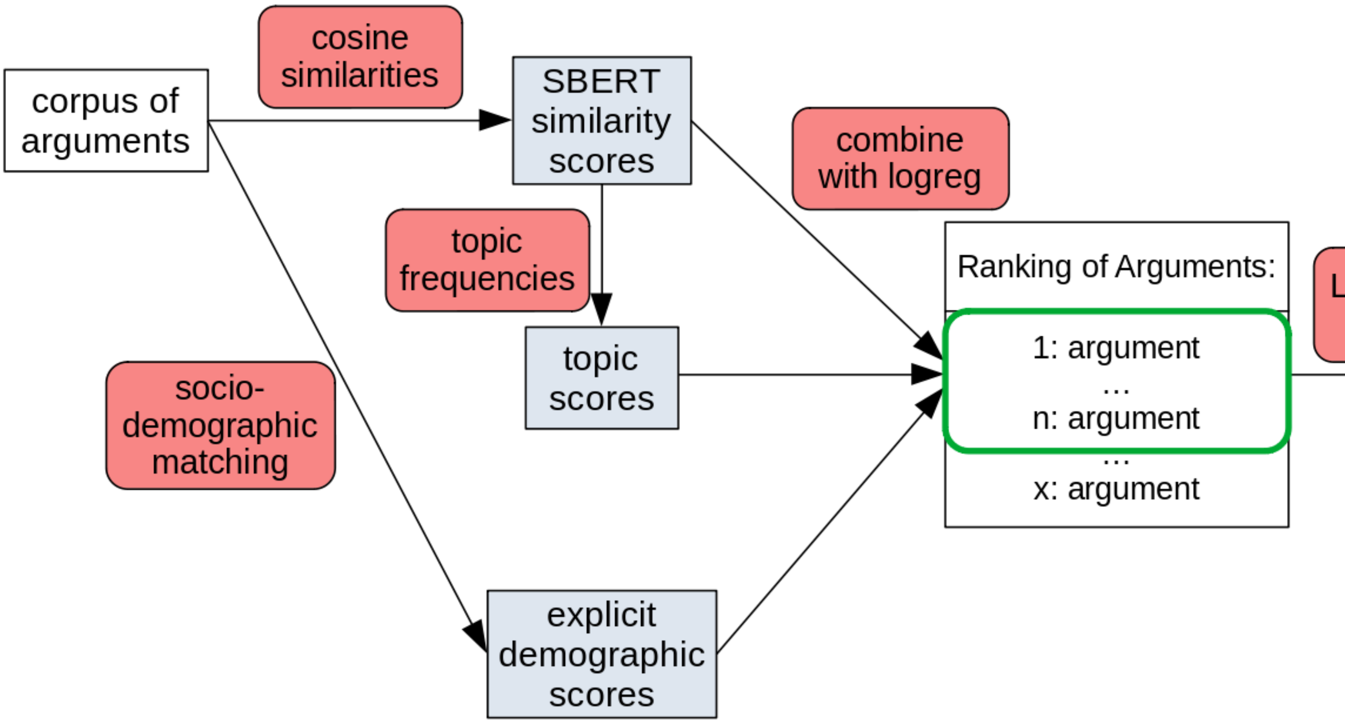


Figure 4.3: Revised Pipeline step 1

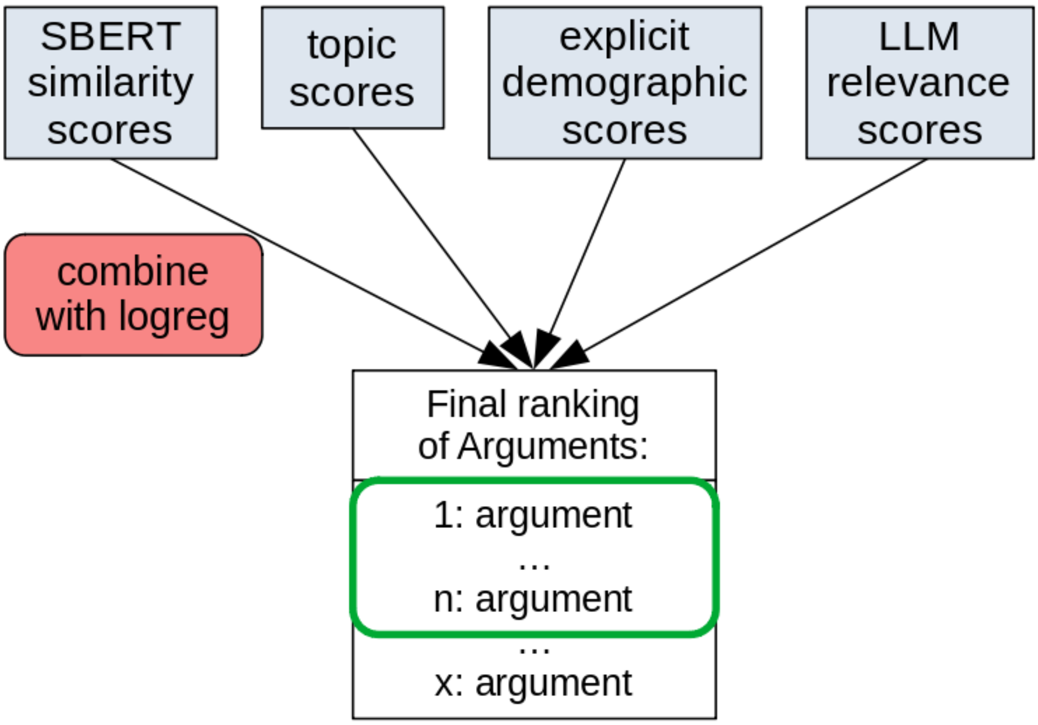


Figure 4.4: Revised Pipeline step 2

Thus, the logistic regression trained on the train-split of the dataset takes only three inputs for each query-argument pair, namely the *SBERT scores*, the *topic score*, and the *LLM relevance score*. The expected output for the logistic regression training is 1 if the argument matches the query based on the content and 0 if it does not. The argument's author's demographic profile is not considered in any way.

After training is complete, the internal logit scores of the regressions can be used as weights to compute a weighted sum of the feature scores mentioned, when predicting relevant arguments for another data split. When creating a final prediction of relevant arguments, the arguments are ordered descendingly based on this weighted sum for each query.

4.4.3 Revised Pipeline for Experiment 3

Experiment 3 aims to retrieve arguments that match a query based on their content and the socio-demographic property included in the query based on the demographic profile of their author, without directly looking at said demographic profile. Therefore, the content-relevance-based *SBERT similarity scores* and *topic scores* are computed, to create an initial ranking of the arguments, which is then used to determine the top 50 arguments for each query. These arguments are included in two different LLM prompts - one of which aims to score the arguments based on how well they match the query based on their content (see Appendix A.1.3), resulting in the *LLM relevance scores*, while the other one aims to score the arguments based on how well their author matches the given socio-demographic property (see Appendix A.1.4), resulting in the *LLM demographic scores*.

The logistic regression is therefore trained with four input features for a query-argument pair, namely the *SBERT similarity score*, the *topic score*, the *LLM relevance score*, and the *LLM demographic score*. The expected output for the logistic regression is the same one as it is for *experiment 1*.

The regressions' internal logit scores are used as the weights. To predict the relevant arguments for a different data split, the weights are used to compute a weighted sum of all four feature scores. The weighted sum is in turn used to order the arguments descending for each query to form the final ranking of arguments for said query.

5
Evaluation

In the previous chapter the methods investigated for each of the research questions were explained. This chapter will describe the experimental setup for the implementation of each method, including the dataset and the evaluation metrics. Then, the outcomes of the experiments will be investigated, by firstly regarding the distribution of the feature scores, and, thereafter, evaluating the results for each experiment and each research question based on the evaluation metrics. Finally the participation in the Perspective Argument Retrieval shared task (Falk et al., 2024) that is linked to this thesis is introduced.

5.1 Experimental Setup

5.1.1 Task Description

The research conducted in the context of this thesis is based on a dataset of political arguments, collected in the context of the two swiss elections from 2019 and 2023. The data was originally collected by the voting recommendation platform SmartVote. The combined dataset consists of roughly 47k arguments made by 3.8k politicians concerning 247 political questions and issues (Falk et al., 2024). The raw data from the SmartVote platform was pre-processed, the queries were annotated with socio-demographic attributes, and the data was split into a train set, a development (dev) set and three different test sets by the authors of the Perspective Argument Retrieval Shared Task 2024 (Falk et al., 2024). The aim of this shared task is the investigation of different socio-demographic perspectives in an argument retrieval setting. Therefore, the shared task introduces three scenarios, that are equivalent to the three experiments described in chapter 4: *Scenario 1*, the *no perspectivism* scenario is equivalent to *experiment 2* investigated in this thesis, while *Scenario 2*, the *explicit perspectivism* scenario is equivalent to *experiment 1* investigated in this thesis, and *Scenario 3*, the *implicit perspectivism* scenario is equivalent to *experiment 3* investigated in this thesis. The submission by team *sövereign* (Günzler et al., 2024) resulted from the research process for this thesis and placed second in the final ranking of the shared task (Falk et al., 2024), the details are described in section 5.2.5.

As aforementioned, the dataset used is made up of political issues and arguments collected during the swiss elections 2019 and 2023. The dataset is multilingual, containing German, French and Italian language. The roughly 47k arguments and 247 political questions and issues are divided into 5 different data splits. The train split of the data contains roughly 21k arguments and 105 political questions, the dev split contains roughly 5k arguments and 30 political questions, and the first test set contains roughly 6k arguments and 45 political questions. These three sets are all based on data from the 2019 swiss election alone. For each of the sets, every political question is present three times, once in every French, Italian and German respectively. Therefore the train set technically only contains 35 unique questions, but has each of those questions translated into two other languages - the same goes for the dev set and the first test set. Test set 2 contains 40 political queries and 12758 political arguments, while test set 3 contains 27 political queries and 2349 political arguments. These two sets contain monolingual queries in German alone. The second test set is based on the swiss election from 2023, while the third test set is based on an annotation study, where readers are presented 20 arguments for each query and select those arguments that they perceive as relevant (Falk et al., 2024).

For the perspective scenarios 2 and 3 (in this thesis *experiments 1* and *3*), the set of political queries is significantly larger, as it contains every query multiple times, annotated with a different socio-demographic attribute each time. The number of political arguments for each split of the data is the size of the set of political arguments that are labeled relevant to any of the queries from that set. While each argument can be assigned to one of the data splits distinctly, the arguments are not provided in distinct sets for each data split. Instead, for each of the three test sets, a different corpus is provided. Every corpus contains the political arguments that belong to the train split of the data, those that belong to the dev split of the data, and those that belong to the current test set. However, the corpus for test set 1 does not contain any arguments that are relevant to a query from test set 2, nor is it the other way around.

During the research process for this thesis, the data was used exactly how it was provided by the authors of the Perspective Argument retrieval shared task. This is partially due to the planned participation in the shared task that required strictly following the shared task rules. After the shared task participation the datasets where not changed in any way, so that comparability with results from the shared task would still be assured.

Graphic 5.1 (Günzler et al., 2024) shows an example for a perspective query and some political arguments. Aside from the political issue the query also contains a socio-demographic property. Each argument includes a profile of socio-demographic properties, a topic and a stance. The graphic also elucidates in which case an argument is relevant to a query. Arguments highlighted in green are considered relevant, because they match based on the content and are authored by someone matching the socio-demographic attribute given in the query. Arguments highlighted in red are considered irrelevant, because they do not match the socio-demographic attribute given in the query. The argument highlighted in orange does match based on it's socio-demographic profile, however it is not considered relevant based on it's content and is therefore also not considered as relevant to the query during the evaluation process.

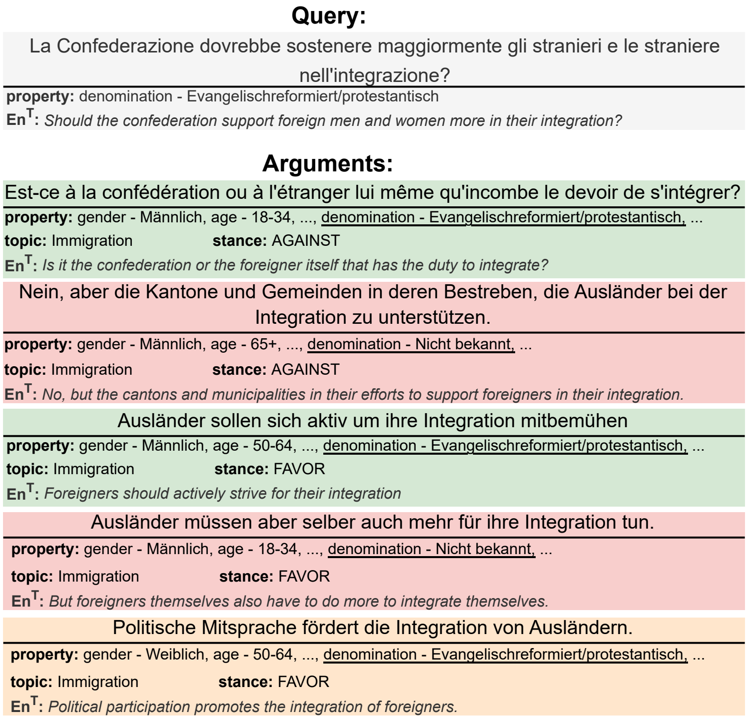


Figure 5.1: Example for multilingual query and arguments. Relevant arguments are marked in green, irrelevant arguments are marked in red (if they don't match the demographic feature) or orange (if they don't match based on content).

5.1.2 Implementation Details

To compute the different feature scores described in chapter 4, different libraries and tools are used. The SBERT similarity scores are computed using the SBERT sentence embedder by Reimers and Gurevych (2019) (see also section 2.2.3). Both the LLM content relevance scores and the LLM demographic relevance scores are generated by the Mixtral8x7b model by MistralAI (Jiang et al., 2024). Two different methods of accessing the Mixtral8x7b model are used during different experiment within this thesis. For some experiments, the model is run locally using Ollama[1] to be able to access the model simply via a local API. In this case, the model is run using standard settings except for the context size, which is set to 8192 tokens. During other experiments, Mixtral8x7b is used through the HuggingChat API[2]. Unfortunately, for these experiments none of the internal settings are visible, which makes attempts to recreate these scores more difficult.

As Large Language Models are probabilistic models, there is a chance of the model behaving in an unexpected way. This can lead to errors when raising and processing the LLM relevance scores. As the scores are processed automatically in this thesis, an error in the answer format of the LLM is problematic, if it makes the LLM's output not machine-readable. The HuggingChat API is also inconsistent in other ways, sometimes returning error codes due to the network connection or due to overload. If any error is detected, the request that lead to the failure is repeated a number of times. If one request fails repeatedly however, the corresponding query will be assigned relevance score of 0 for every argument, meaning the the corresponding query will be ranked

1. https://ollama.com
2. https://github.com/Soulter/hugging-chat-api

only based on the other feature scores. A similar procedure is used for incomplete answers, where the LLM returns relevance scores in a machine readable format but does not include all of the arguments that it asked to include. In this case, the missing arguments are assigned a score of 0.

Among other things, this thesis investigates different topic scores and different ways of getting LLM relevance scores. Therefore, different strategies of computing topic scores and LLM scores are immediately compared in the following sections. Most of these comparisons are performed either on the dev split of the data or even on a subset of the dev split. The findings from these small experiments are then used in the attempt of improving results on other splits of the data, mainly the three test sets that are investigated. As mentioned in section 5.1.1, both the train and the dev set as well as the first test set are based on data from the Swiss election 2019, while the second data is based on Swiss elections from 2023 and the third test set is based on readers' annotations and therefore contains a different perspective of arguments. The differences in origin and nature of the data between the dev set and especially the second and third test set could impair the performance when using methods on the test sets that have proven to be effective on the dev set.

5.1.3 Evaluation Metrics

Information retrieval is often evaluated using recall and precision. Recall measures how many of the actually relevant arguments are found by the system, while precision looks at how many of the retrieved arguments are actually relevant. Both metrics are useful because they give a basic idea of how well a system retrieves relevant information.

However, in the perspective argument retrieval shared task, the system does not return a simple set of relevant arguments but instead ranks them. A key difference is that there is no built-in cutoff point that separates relevant from irrelevant arguments. This means that recall and precision, which are normally used for retrieval tasks, can only be calculated at predefined cutoff points.

Because of this, precision isn't ideal as an overall evaluation metric for the system. Precision measures how many retrieved arguments are actually relevant, but in this case, the system does not mark any arguments as *retrieved* or *not retrieved* — it just ranks them. A cutoff would artificially label the arguments above it as positive and those below it as negative, but since the system itself doesn't define a cutoff, precision doesn't really fit as a main metric. That said, precision at the same cutoff points can still be useful when comparing different approaches within the same experiment because it is simple and easy to understand.

The same applies to recall. Normally, recall is calculated as the number of retrieved relevant arguments divided by the total number of relevant arguments. But here, the system doesn't decide what is retrieved — it just ranks everything. So recall can only be calculated at certain cutoff points. In this thesis, recall is mainly useful for evaluating the initial retrieval step, which is based on SBERT similarity and topic scores. It's not really helpful for evaluating the LLM scores when looking at the entire dataset because the LLM only ranks a small subset of arguments anyway.

Since socio-demographic attributes of argument authors play a major role in the experiments, it makes sense to include diversity metrics alongside relevance metrics. As mentioned before, the system is purely a ranking system and does not actually predict relevance, which is a direct result of how the shared task is formulated. Because of this, it is better to use ranking-based metrics instead of traditional retrieval metrics. The shared task organizers use four metrics: two for relevance (NDCG and precision) and two for diversity (alpha-NDCG and KL divergence).

NDCG is the main metric for evaluating ranking performance. It is designed specifically for ranking tasks because it assigns higher weight to arguments ranked at the top. Like precision, it is also calculated at predefined cutoff points, but because it discounts lower-ranked arguments, it is more flexible and less dependent on where the cutoff is set. Precision, while not ideal for a ranking task, is still useful because it is easy to interpret, and computing it at multiple cutoff points can help understand how well the ranking performs.

For diversity, alpha-NDCG is a modified version of NDCG that penalizes repeated occurrences of the same socio-demographic group. If a group appears multiple times in the ranking, their scores are reduced, meaning that a high alpha-NDCG score suggests a more diverse ranking. However, since it is still based on NDCG, it depends heavily on how well the system ranks relevant arguments. KL divergence, on the other hand, compares the distribution of socio-demographic groups in the predicted relevant arguments with their distribution in the whole dataset. A low KL divergence means that the demographic distribution in the ranked arguments is similar to that of the entire dataset, which is often seen as a sign of fairness. The shared task organizers suggest that a high KL divergence could mean the model is biased towards dominant groups, but it could also mean that underrepresented groups are getting more visibility — which arguably might be desirable in a more diverse ranking.

In the shared task, all metrics are computed at cutoff points 4, 8, 16, and 20, meaning that only a small number of arguments are considered in each evaluation.

The evaluation in this thesis follows the same metrics as the shared task for several reasons. First, it allows for a direct comparison between the methods tested in this thesis and those developed by other shared task participants. Second, the system is purely a ranking system, meaning there is no way to automatically determine a cutoff between relevant and irrelevant arguments. Third, NDCG is the most suitable metric for this type of task, as it is specifically designed for ranking problems. Precision, while not as meaningful for ranking, is still useful because of its simple interpretation. Alpha-NDCG provides some insight into diversity but, since it is still an NDCG-based metric, it does not give a complete picture of diversity on its own. KL divergence, on the other hand, shows whether the demographic distribution of the ranked arguments differs significantly from the full dataset. However, since a high KL divergence could indicate either bias towards dominant groups or increased representation of minorities, it does not necessarily give a clear answer about how diverse a ranking actually is.

While these diversity metrics help in analyzing demographic representation, they are not enough to make strong claims about diversity. Since diversity is not a main research question in this thesis, it is not the primary focus of the evaluation.

In addition to computing NDCG, precision, alpha-NDCG, and KL divergence at cutoffs 4, 8, 16, and 20, for some predictions, precision and recall are plotted across cutoff points from 1 to 50. This helps to better visualize how different ranking methods compare.

5.2 Experimental Outcomes

This section aims to evaluate the different feature scores and methods investigated in this thesis with the goal of answering all three research questions. Therefore, in the following, the feature scores used in the final pipeline for each of the experiments are looked at more closely, to showcase possible differences and similarities in the distribution of the different feature scores. Then, the research questions are addressed one at a time, by evaluating all methods that were investigated in the context of answering the respective research question. Finally, the system that was submitted to the Perspective Argument Retrieval shared task (Falk et al., 2024) is presented and evaluated in comparison to other submissions.

5.2.1 Feature Score Distributions

This section aims to showcase the distribution of the feature scores that are used in the pipeline for any of the experiments 1, 2, or 3.

Experiment 1

For *experiment 1*, four different feature scores are used to predict relevant arguments. These are the *SBERT similarity scores*, the *topic scores*, the *explicit demographic scores*, and the *LLM relevance scores*.

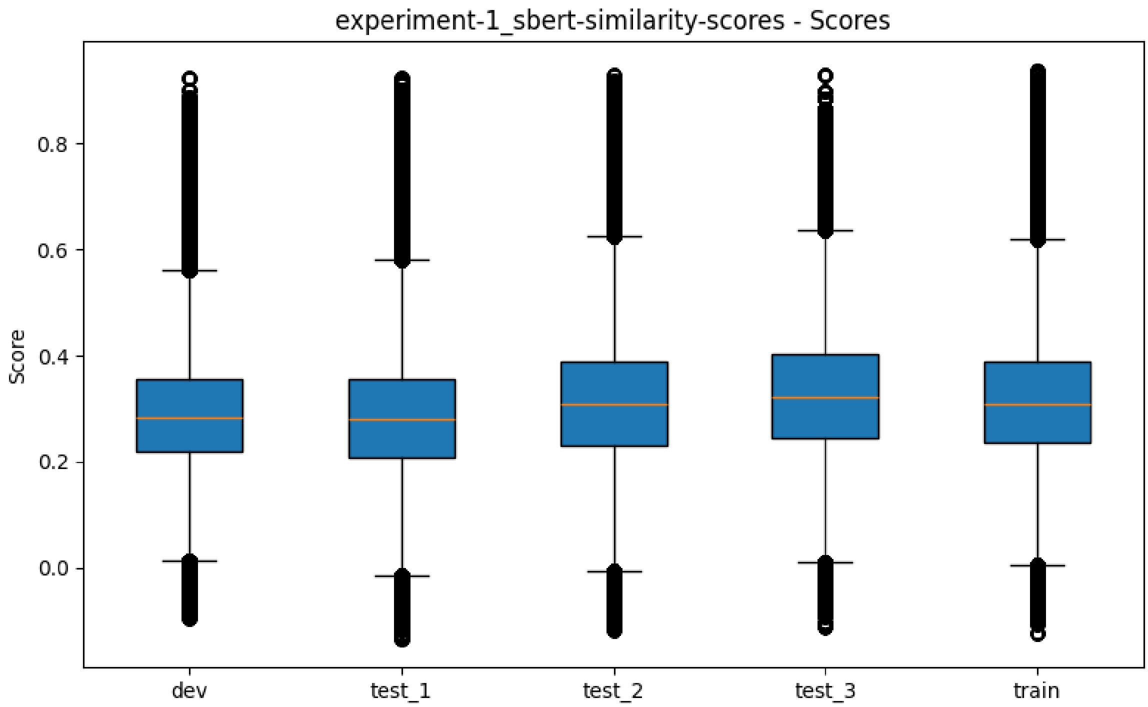


Figure 5.2: Distribution of SBERT similarity scores

Graphic 5.2 shows the distribution of the *SBERT similarity scores* for each data split. Generally, the distribution is similar for each split of the data with the median ranging from 0.280 for the first test set to 0.321 for the third test set. The *SBERT similarity scores* are roughly normally distributed. However, as aforementioned, during the evaluation of

Split	Score (%)	
	1 (Author Matches Demographic)	0 (Author Doesn't Match Demographic)
Dev Set	17.727	82.273
Test Set 1	18.162	81.838
Test Set 2	13.448	86.552
Test Set 3	18.295	81.704
Train	17.921	82.079

Table 5.1: Distribution of scores across different dataset splits, indicating the percentage of cases where the author's demographic matches or does not match the target demographic.

the predictions for each of the experiments, the different metrics are mostly computed only for the top 20 highest ranking arguments for each query. Therefore, most of the time, only the very upper end of the score distribution will be considered in the evaluation of the final ranking. The maximum similarity for one query-argument pair ranges from 0.922 on the dev set to 0.938 on the train set.

Table 5.1 shows the distribution of the *explicit demographic scores*. The explicit demographic score for a query-argument pair is 0 if the argument's author does not match the socio-demographic property given in the query, and 1 if it does. The table indicates that on average per query between 13.4 and 18.3 percent of the arguments match the query based on their author's demographic profile. For the first experiment this means that on average between 81.7 and 86.6 percent of the arguments can effectively be filtered out both from the predictions. These scores also give some insight concerning the third experiment, in which the demographic profiles of the arguments' authors are not available. Instead, to predict whether the argument matches the demographic feature, an LLM is prompted to score the top 50 most relevant arguments based on their demographic matching. As, on average and varying depending on the dataset, only around 15% of arguments match the demographic property that is asked for in the query, on average only around 8 of the arguments 50 arguments judged by the LLM will match the demographic property given. As the different evaluation metrics are regarded up to rank 20 this is a limitation to the *Demographic LLM scores* that cannot be neglected.

The *topic scores* try to predict the topic of a query from the highest ranking arguments based on *SBERT similarity scores*. In section 4.2.1 different approaches to compute the topic scores are introduced that differ mainly in the way the positions of arguments within the top 50 ranking are weighted towards the *topic scores*. All topic scores are normalized, thus range from 0 to 1. Additionally, for one query, the scores assigned to all individual topics add up to 1. Therefore, a score of 0 is assigned to a topic if that topic is ruled out to match the regarded query based on the current approach. A score of 1, on the other hand, implies that all other topics are ruled out from matching the regarded query based on the current approach. Scores strictly between 0 and 1 occur only if the approach used considers multiple topics as possible relevant to the query. Generally a high density in 0-scores is expected, as it is highly unlikely that all different topics occur in the top 50 arguments based on *SBERT similarity scores* for any query. Therefore, some topics will always be ruled out by all of the approaches and receive a *topic score* of 0.

The earliest version of the topic scores, the *binary topic scores*, per definition only assigns score other than 0 to a single topic, predicting the most likely topic. A strategy like this immediately rules out all relevant arguments if the wrong topic is predicted

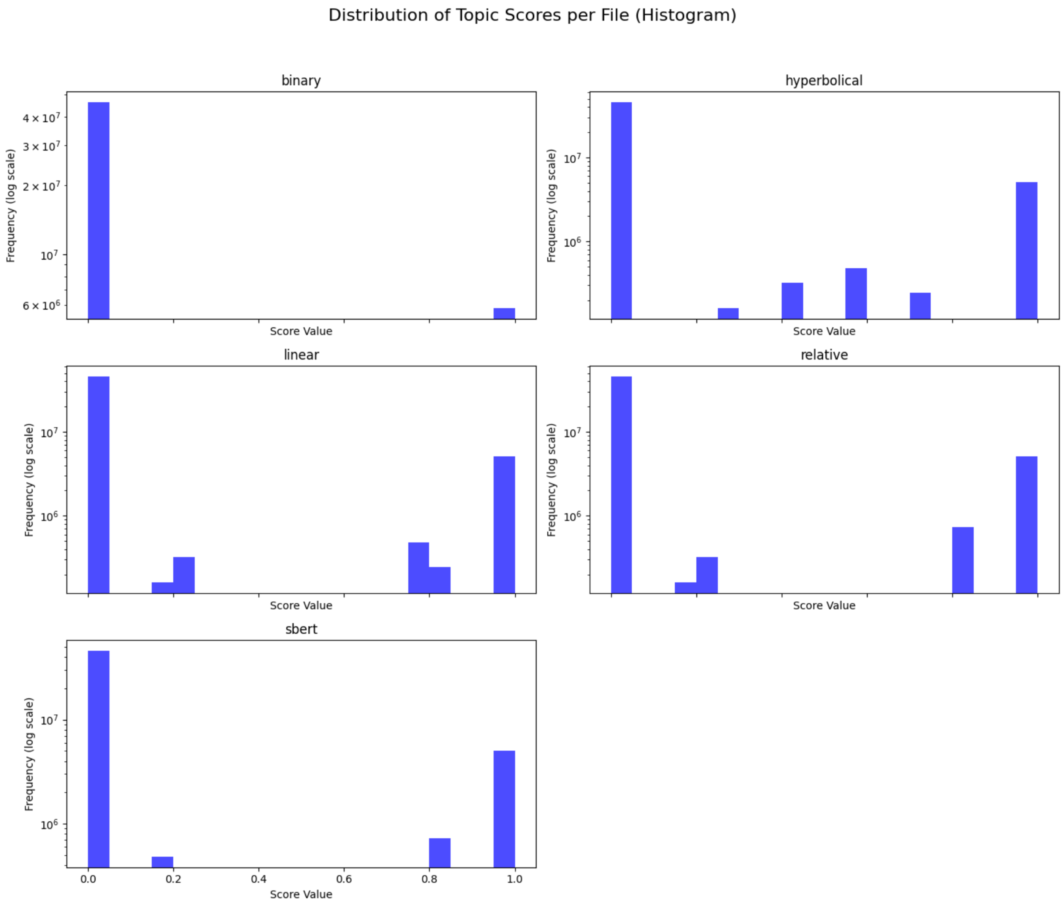


Figure 5.3: Different Topic score approaches.

for any query. The other approaches to compute topic scores aim to solve this issue by allowing less polarized distributions in ambiguous situations. Graphic 5.3 shows the distribution for each of the computation approaches. While the newer approaches produce values other than 0 and 1, the distributions are still strongly polarized with a strong tendency towards the scores 0 and 1.

The scores used to compute the final predictions for each experiment are the *relative topic scores*. Graphic 5.4 shows the distribution of the *relative topic scores* for the different splits of the data. It shows the most polarized distribution for the dev set. The scores for the train set and for the first test set also show strong tendencies towards scores 0 and 1. The arguments and queries from these three sets were all collected in the same context (Falk et al., 2024). As expected, test sets 2 and 3 show a high concentration of the score 0, however both distributions are much more diverse. Especially the distribution for test set 3 shows a very low tendency towards a score of 1, which indicates ambiguity for nearly every query.

The LLM scores used for the final predictions for each experiment predicts *LLM relevance scores* only for the top 50 arguments for each query. Graphic 5.5 shows the distribution of those 50 topic scores for each split of the data. The means of the distributions range from 0.55 on test set 3 to 0.66 on the dev set. The median is 0.7 for every set but the third test set, which has a median of 0.6. Test sets 2 and 3 exhibit lower scores than the train set, the dev set and the first test set. It is worth noting that any argument that is provided in the prompt to the LLM but is not included in the LLM

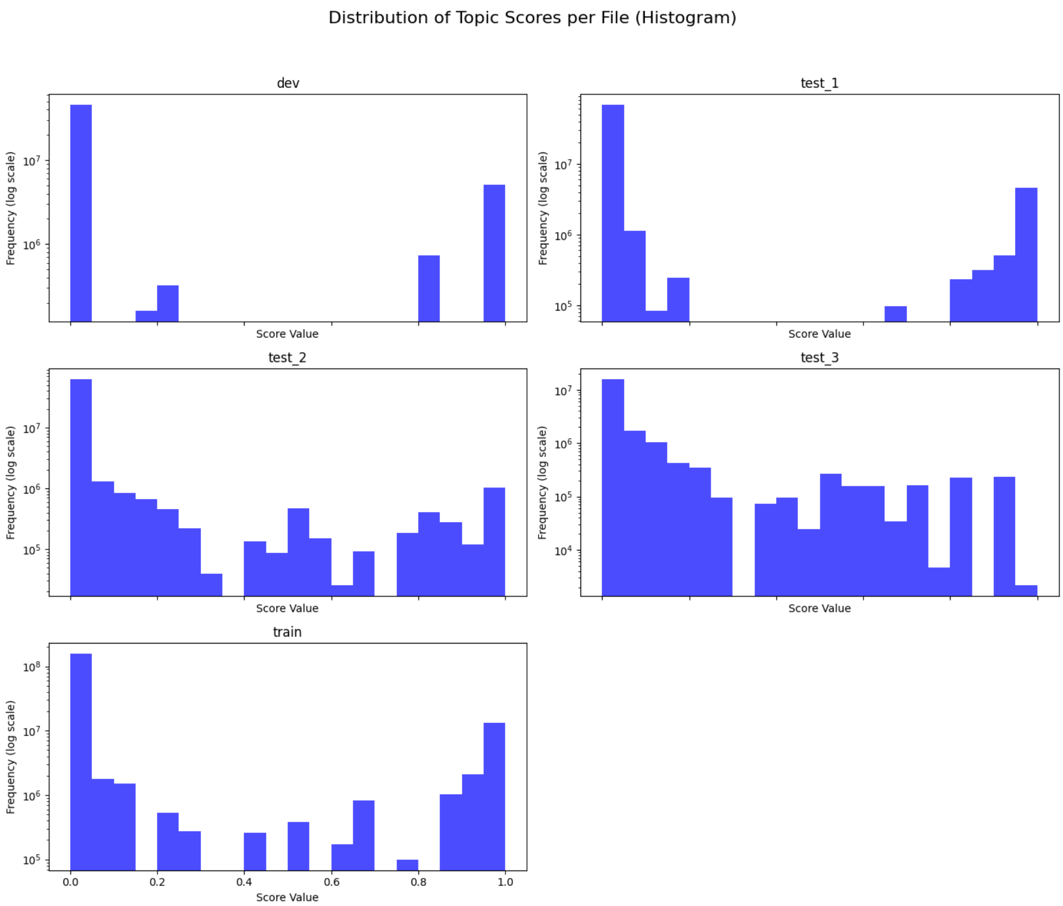


Figure 5.4: Topic scores across different test sets

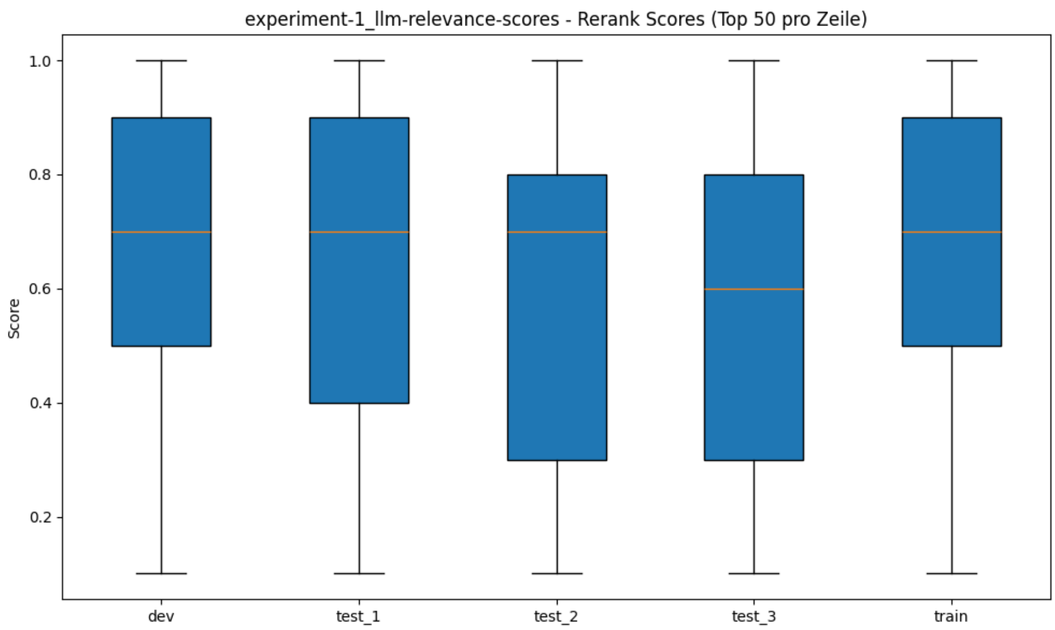


Figure 5.5: Distribition of the LLM relevance scores across the top 50 arguments for each query across different test sets for experiment 1

answer receives an *LLM relevance score* of 0. Therefore, lower scores as seen for test sets 2 and 3 could also arise from more incomplete model answers.

Experiment 2

The *second experiment* does not include a demographic property in the query and therefore does not compute any kind of *demographic score*. Therefore, there are three scores computed for *experiment 2*. These are the *SBERT similarity scores*, the *topic scores* and the *LLM relevance scores*.

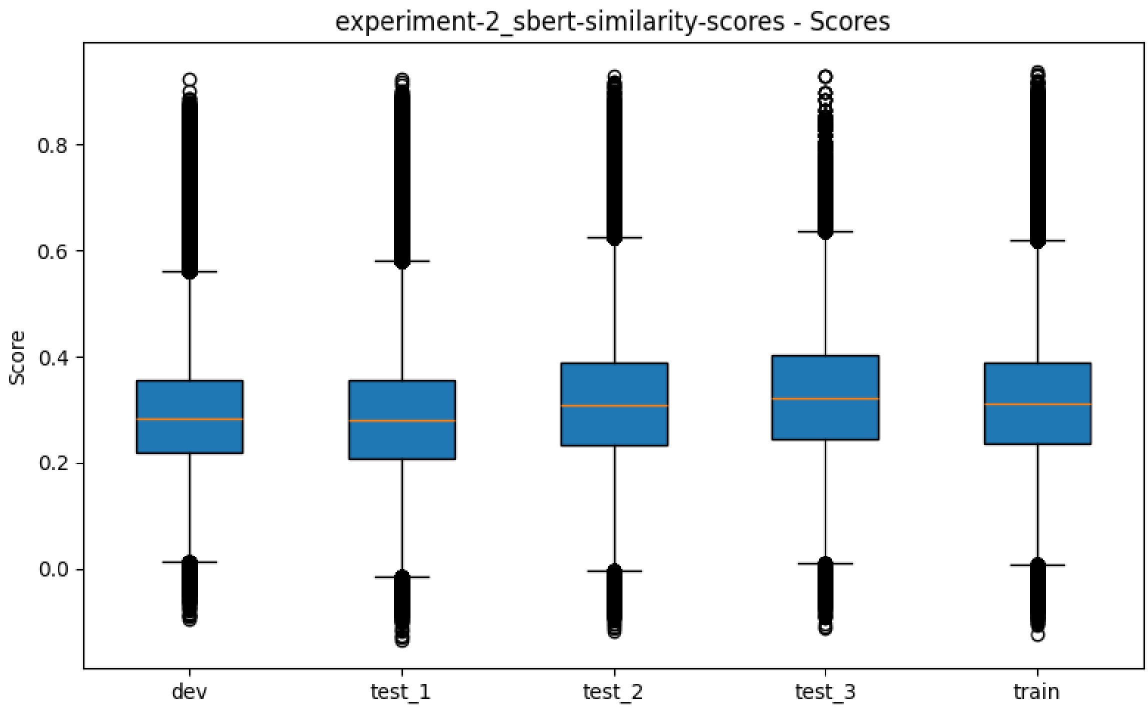


Figure 5.6: Distribution of SBERT similarity scores for experiment 2

Graphic 5.6 shows the distribution of the *SBERT similarity scores* for each split of the data. While the distribution does differ from the distribution for *experiment 1* shown in graphic 5.2, the relative distribution shown in the boxplots is exactly the same for *experiments 1* and *2*. This is because the *SBERT similarity scores* only consider the texts of the queries and the arguments themselves. *Experiment 1* uses the same queries as *experiment 2*, where every query is used multiple times annotated with a different demographic property every time. Because the demographic property is not considered in the *SBERT similarity scores*, the similarity scores for *experiment 1* are effectively the same scores as for *experiment 2*, computed multiple times, once for every demographic property. This explains the difference in absolute numbers resulting in the same boxplot.

Graphic 5.7 shows the *relative topic scores* for *experiment 2*. All variants of *topic scores* are computed on the top 50 ranking arguments based on *SBERT similarity scores*. Therefore, just like the *SBERT similarity scores*, the *topic scores* do not depend on any socio-demographic aspects. Thus, the distribution of *topic scores* between *experiment 1* and *experiment 2* differs only in absolute numbers. Graphics 5.4 and 5.7 use absolute frequencies for each the *topic scores*, which is why both graphics look identical, only

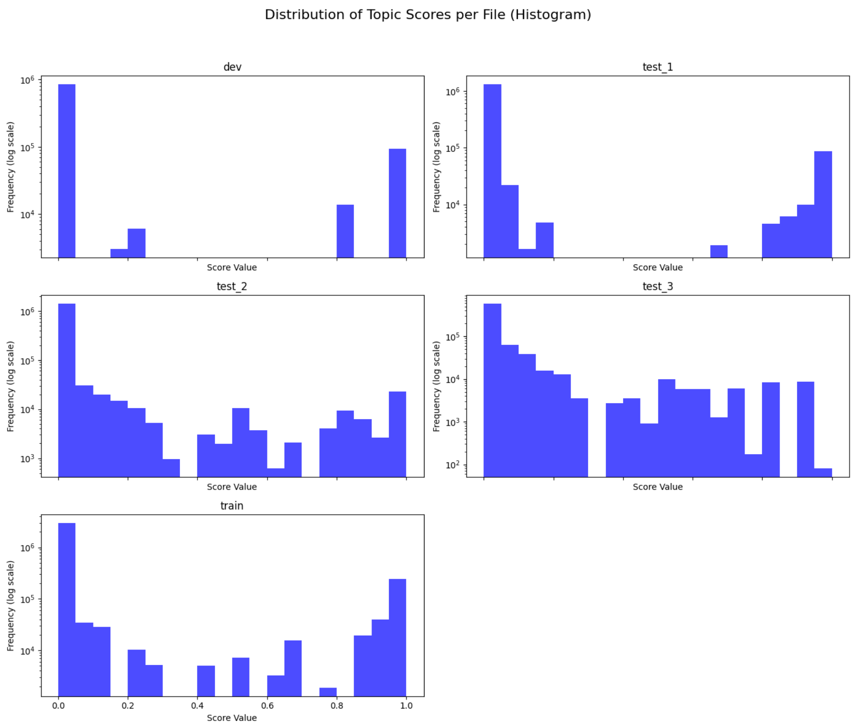


Figure 5.7: Relative topic scores across different data splits for experiment 2

differing in the y-axis scale. As mentioned above, for the dev set, the train set and the first test set, the *topic scores* show strong polarization indicating rather unambiguous topics. Test set 2 and especially test set 3 show less tendency towards the *topic score* 1, indicating more ambiguous topics.

Just like in *experiment 1*, the *LLM relevance scores* are only computed on the top 50 arguments for each query (shown in graphic 5.8). However, in the *second experiment*, the top 50 are determined based on *SBERT similarity scores* and *topic scores* alone, as opposed to the *first experiment*, where the top 50 arguments are determined using *SBERT similarity scores*, *topic scores* and *explicit demographic scores*. This means that in the *second experiment* the arguments are selected purely based on content relevance measures and are therefore likely more relevant. This shows in the *LLM relevance scores* that are overall significantly higher than they are in the *first experiment*. While the mean during *experiment 1* was ranging from 0.556 to 0.662, for the *second experiment* it is ranging from 0.722 for the train set to 0.827 for the dev set. For the *first experiment* the *LLM relevance scores* for test sets 2 and 3 were significantly lower compared to the other splits of the data. This trend is not visible during the *second experiment*. Instead the scores for the dev set are significantly higher compared to every other split of the dataset.

Experiment 3

The *third experiment* does include a demographic property in the query but does not consider the demographic profile given about each argument's author available. To compute a prediction for *experiment 3*, four scores are computed. Similar to *experiments 1* and *2*, the *SBERT similarity score*, the *topic score* and the *LLM relevance score* are computed. As a demographic property is given in the query but the demographic

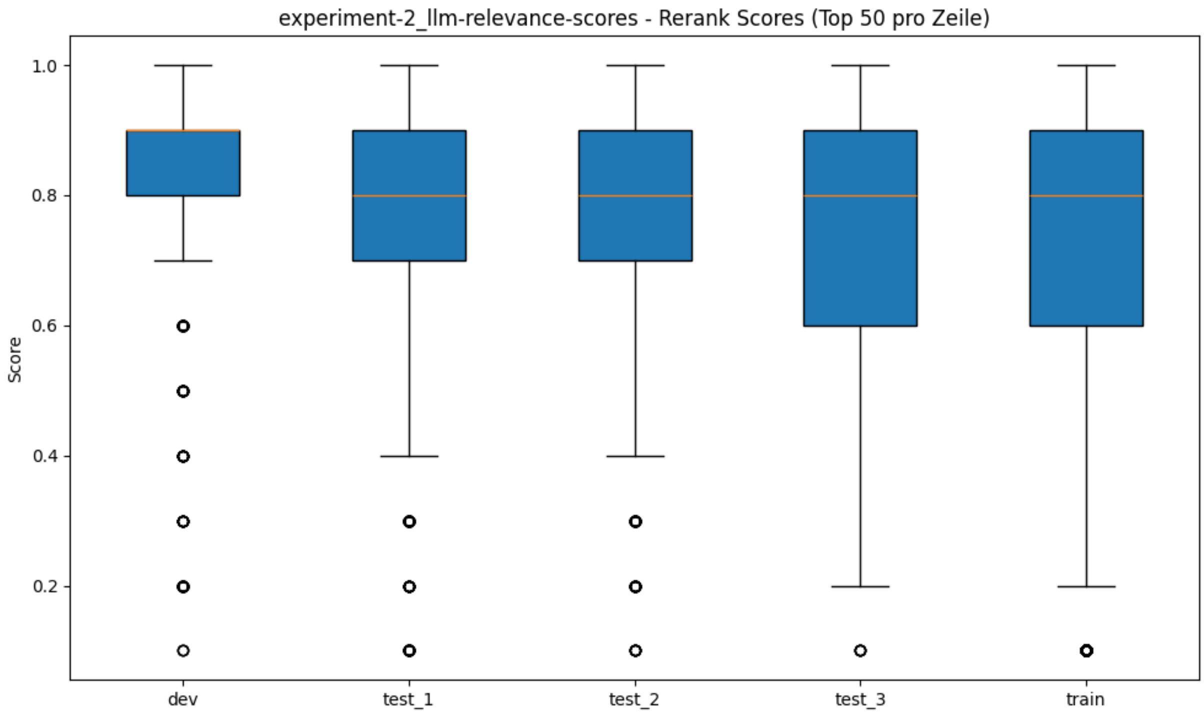


Figure 5.8: Distribution of the LLM relevance scores across the top 50 arguments for each query across different test sets for experiment 2

profiles are not considered, for *experiment 3 Demographic LLM scores* are used instead of *explicit demographic scores*.

Because the queries for *experiment 3* include a demographic property, the same set of queries is used as for *experiment 1*. As aforementioned, neither the *SBERT similarity scores*, nor the *topic scores* depend in any way on the demographic properties of queries or arguments. Hence, the *SBERT similarity scores* and the *topic scores* used for the *third experiment* are exactly the same scores as in *experiment 1*.

For *experiment 3*, both the *LLM relevance scores* and the *Demographic LLM scores* are computed for the top 50 arguments of an initial prediction. Because no demographic information about the arguments' authors are available, this initial prediction in *experiment 3* is based on the *topic scores* and the *SBERT similarity scores* alone, just like it is in *experiment 2*. As explained, the set of queries for *experiments 1* and *3* is based on the set of queries from *experiment 2*, the only difference being that every query is included multiple times annotated with different demographic aspects. Because the *LLM relevance scores* do not consider demographic aspects, and because they are effectively based on the same initial ranking, the distribution of the *LLM relevance scores* for *experiment 3* is essentially the same as the distribution for *experiment 2*, only differing in absolute numbers. Therefore, the *LLM relevance scores* for *experiment 3* are generally quite high, with especially high scores on the dev set (as shown in graphic 5.9), just as described for *experiment 2*.

Graphic 5.10 shows the distribution of the *Demographic LLM scores* for every split of the data. As aforementioned, the *Demographic LLM scores* are computed only for the top 50 arguments per query, which is why graphic 5.10 only includes these arguments. The means range from 0.621 on the train set to 0.749 on the dev set. The highest *Demographic*

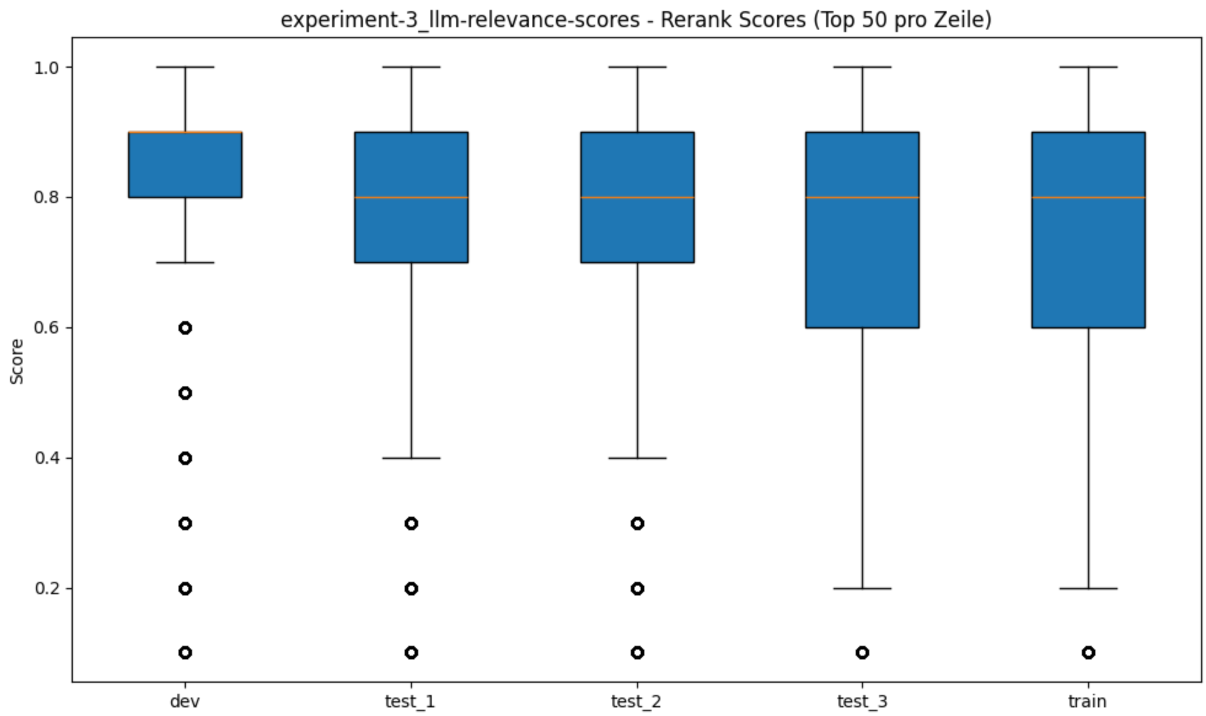


Figure 5.9: Distribition of the LLM relevance scores across the top 50 arguments for each query across different test sets for experiment 3

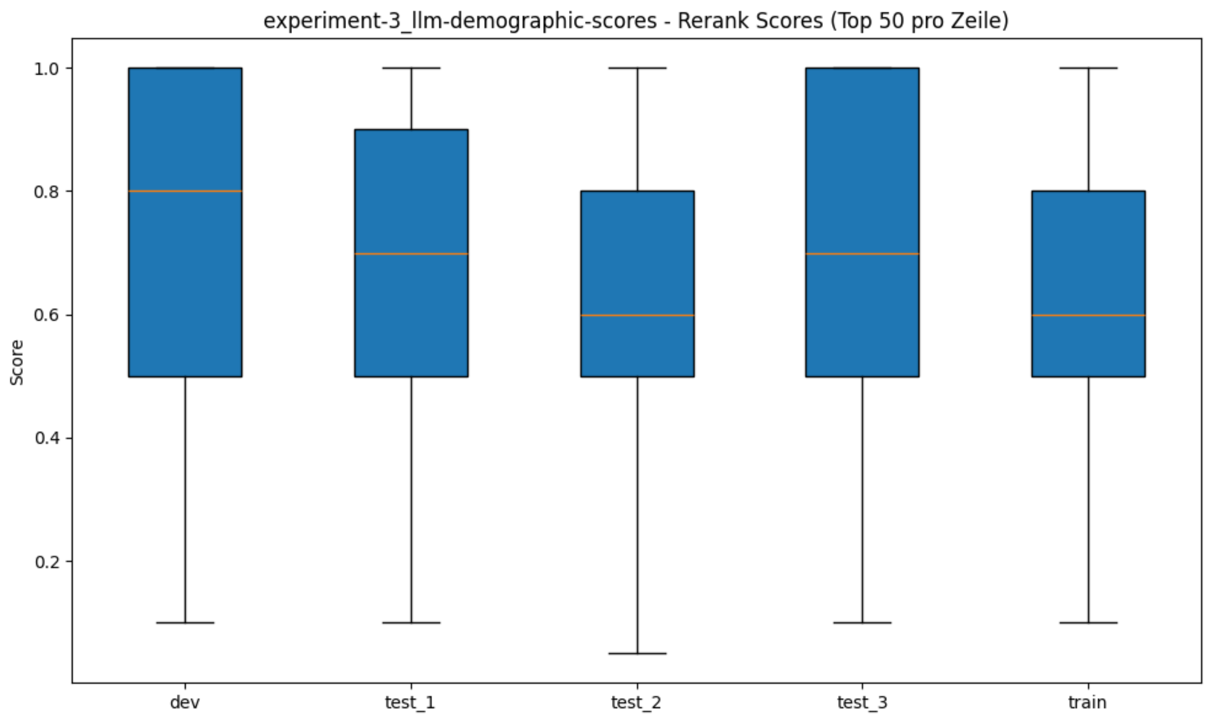


Figure 5.10: Distribition of the demographic LLM scores across the top 50 arguments for each query across different test sets for experiment 3

LLM scores are distributed for the dev set, while the train set and second test set receive lower scores. As mentioned above the amount of demographically matching arguments for each query is lowest on the second test set (see table 5.1), which could explain the lower *Demographic LLM scores*. However, the low *Demographic LLM scores* for the train set and the higher scores for the dev set cannot be explained with this logic.

5.2.2 Research Question 1

Research question 1 aims to investigate the usage of LLMs in an argument retrieval process. Due to the high computational cost of LLM inference, the LLMs are deployed only in the reranking step of a two-stage retrieval pipeline. To comprehensively answer research question 1, this section will focus on the reranking step of the two-step retrieval pipeline for *experiments 1* and *2*, analyzing the usage and effectiveness of *LLM relevance scores*. However, as the initial retrieval step of the two-step approach will serve as a basis for all LLM scores computed throughout this thesis by determining the 50 arguments passed to the LLM, it will also be analyzed in this section.

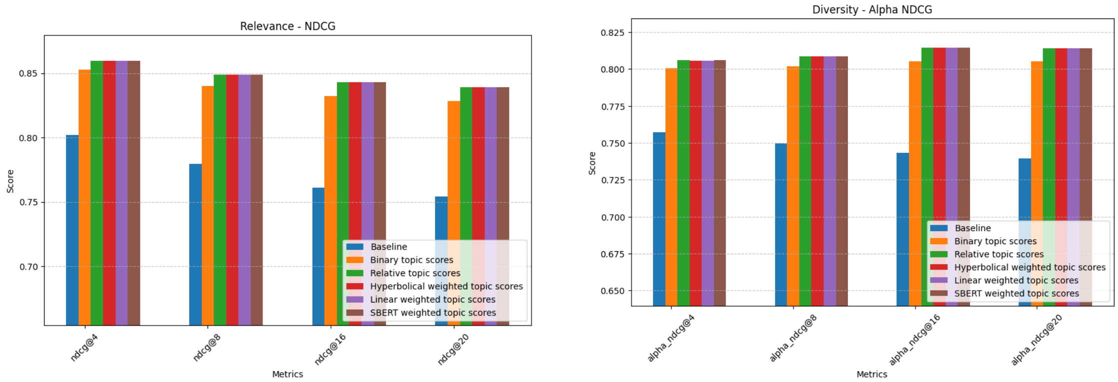
First, the two-step retrieval pipeline for *experiment 1* will be analyzed, including a detailed analysis of the content- and demographic-based methods used in the initial retrieval step and of the different variations of LLM scores used in the reranking step. The analysis for *experiment 1* also includes reviewing the overall results of the two-step pipeline on all different test sets. Subsequently, the performance of the relevance-based retrieval methods and the *LLM relevance scores* on *experiment 2* is evaluated.

Experiment 1

The goal of *experiment 1* is to retrieve arguments that are relevant to a query, which includes a demographic property, based on the argument's content and its author's demographic profile. The pipeline for this experiment involves *SBERT similarity scores*, which are introduced by (Falk et al., 2024) as a baseline for the Perspective Argument Retrieval shared task and are thus considered the baseline throughout *experiments 2* and *3* in this thesis as well. It also involves the *explicit demographic scores*, which are used to effectively filter out all arguments that do not match the demographic property given in the query. For *experiment 1*, the combination of *SBERT similarity scores* and *explicit demographic scores* is considered the baseline.

Aside from these scores, the pipeline also involves *topic scores* during the initial retrieval step as well as *LLM relevance scores* for the reranking step. These feature scores are not only used for *experiment 1* but rather thoughout all experiments and research questions investigated in this thesis. However, the development of the *topic scores* and the *LLM relevance scores*, where different variations were tested and compared, was conducted mostly during *experiment 1*. Therefore, in following, the different variations of the *topic scores* and the *LLM relevance scores* are analyzed, before the performance of an LLM supported retrieval pipeline across all splits of the data set is evaluated.

Topic Scores This section evaluates the performance of the different topic scores introduced. For the evaluation, each variation of topic scores is combined with the baseline for *experiment 1* consisting of the *SBERT similarity scores* and the *explicit demographic scores* to calculate a predicition on the dev split of the data. Both the relevance metrics (NDCG precision) and the diversity metrics (alpha-NDCG & kl-divergence) are computed at ranks 4, 8, 16, and 20.



(a) NDCG for different topic score variations on the dev-set

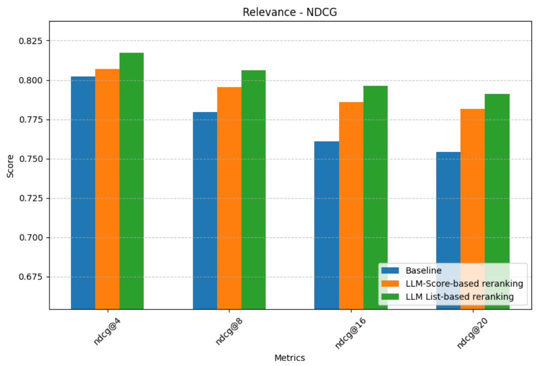
(b) alpha-NDCG for different topic score variations on the dev-set

Figure 5.11: NDCG and alpha NDCG for the different topic scores. Blue is the baseline without topic scores, yellow includes binary topic scores, green includes relative topic scores, red includes hyperbolical weighted topic scores, purple includes linear weighted topic scores and brown includes SBERT weighted topic scores

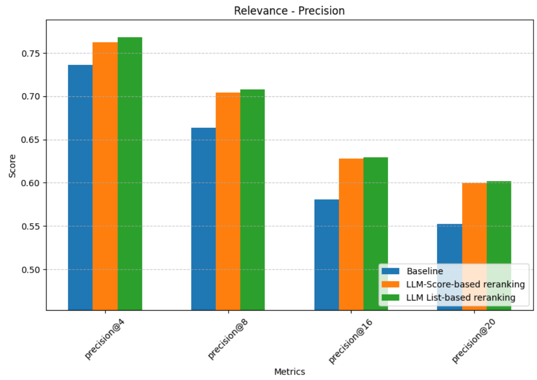
Graphic 5.11 representatively shows the NDCG (a) and the alpha-NDCG (b) results for each of the topic score variations. The graphs for both NDCG and alpha-NDCG show an improvement for any version of the topic scores compared to the baseline method. The graphs also show a slightly better performance of *relative topic scores*, *hyperbolical weighted topic scores*, *linear weighted topic scores*, and *sbert weighted topic scores* compared to the *binary topic scores* in terms of NDCG and alpha-NDCG. However, comparing the *relative topic scores,* the *hyperbolical weigthed topic scores*, the *linear weighted topic scores*, and the *sbert weighted topic scores*, no difference can be observed on either NDCG and alpha-NDCG. Similar effects can be observed for precision and kl-divergence (see appendix A1). Since there is no noticeable difference between the topic score variations except for the *binary topic scores*, for any upcoming experiment, the *relative topic scores* will be used.

Task Modes This section compares the two task modes used when prompting the LLM. Two styles of prompts were tested - one, where the LLM was asked to return the reranked list of arguments, and one, where the LLM was asked to return a score for each argument based on it's relevance to the query to then order the arguments based on these relevance scores. Both approaches are compared to the baseline scenario for *experiment 1* based on NDCG, precision, alpha-NDCG and KL-divergence at ranks 4, 8, 16, and 20. The results are computed on the dev split of the data.

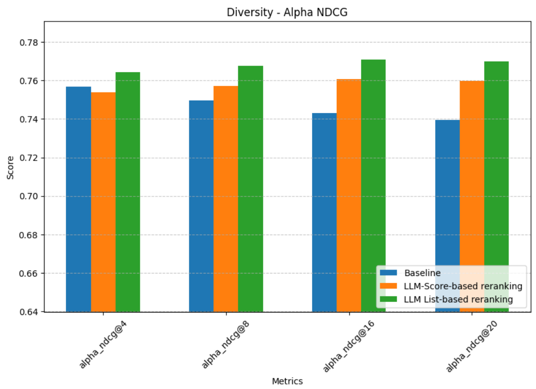
Graphic 5.12 compares the relevance and diversity metrics for each of the task modes with the baseline method. For both NDCG and precision, both LLM reranking approaches improve the relevance of the predictions at all ranks, while the list-based reranking yields a better performance than the score-based approach. Concerning the diversity, the picture is less uniform. While alpha-NDCG shows a similar trend, where LLMs, especially the list-based approach generally improves the results, with one exception at rank 4, where the baseline receives better results than the score-based



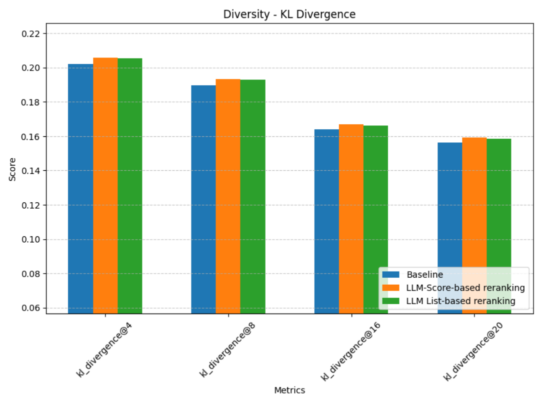
(a) NDCG for different task modes



(b) Precision for different task modes



(c) Alpha-NDCG for different task modes



(d) KL-divergence for different task modes

Figure 5.12: Relevance and diversity metrics for different task modes regarding the LLM. Blue is the baseline, yellow is a ranking based on LLM relevance scores, green is a ranking based on the list-based reranking approach

approach, the KL-divergence suggests a different trend, where the baseline achieves a lower divergence than both LLM scores and the score based approach receives the highest scores at all ranks.

The two general trends that can be observed in the graphs for NDCG and alpha-NDCG, which, due to the ranking-nature of the task are considered the most crucial metrics, are:

1. LLM-based reranking approaches mostly improve the results compared to the baseline. 2. List-based reranking results yield greater improvement than score-based methods.

Because of the higher flexibility of LLM-scores compared to list-based rankings, further investigation focuses mainly on the LLM-scores rather than the list-based approach - even though the list-based approach yields better performance on all metrics evaluated.

Window Sizes Figure 5.13 compares the NDCG for different window sizes, while graphic 5.14 compares the NDCG for the sliding window approach with sliding window size 50, step size 20 and a context window of size 100. Generally the results are inconsistent. At the top ranks the baseline performs better than the different reranking approaches, while for NDCG at higher ranks the reranking of window sizes form 40 to 60 achieves the best results. The investigates sliding window approach achieves similar

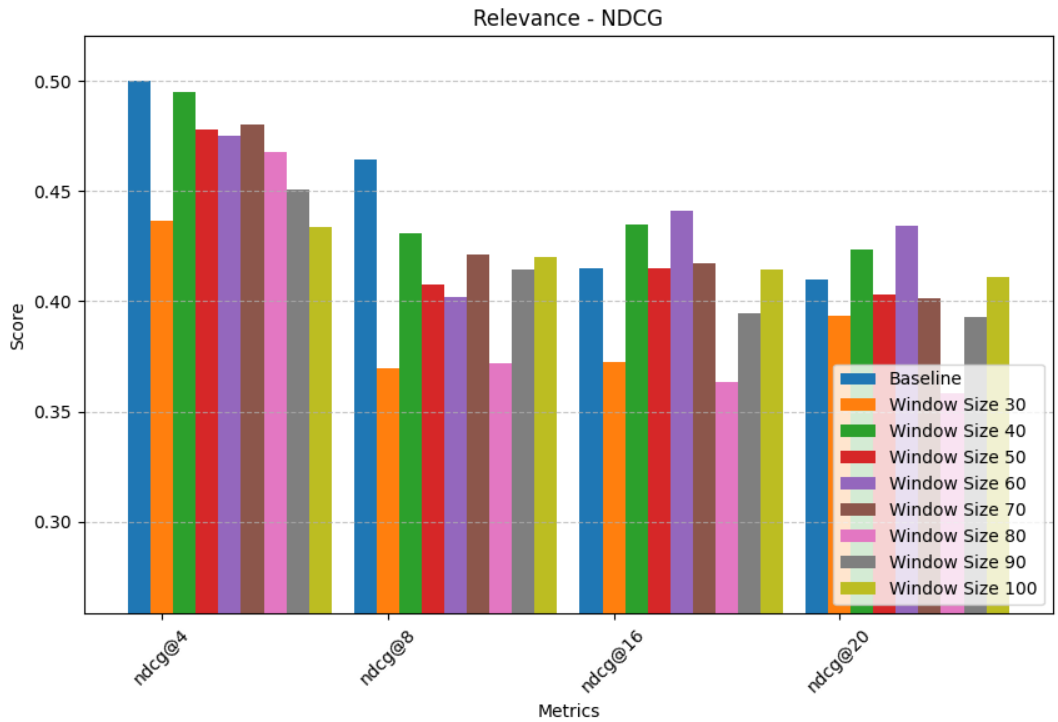


Figure 5.13: NDCG for different window sizes

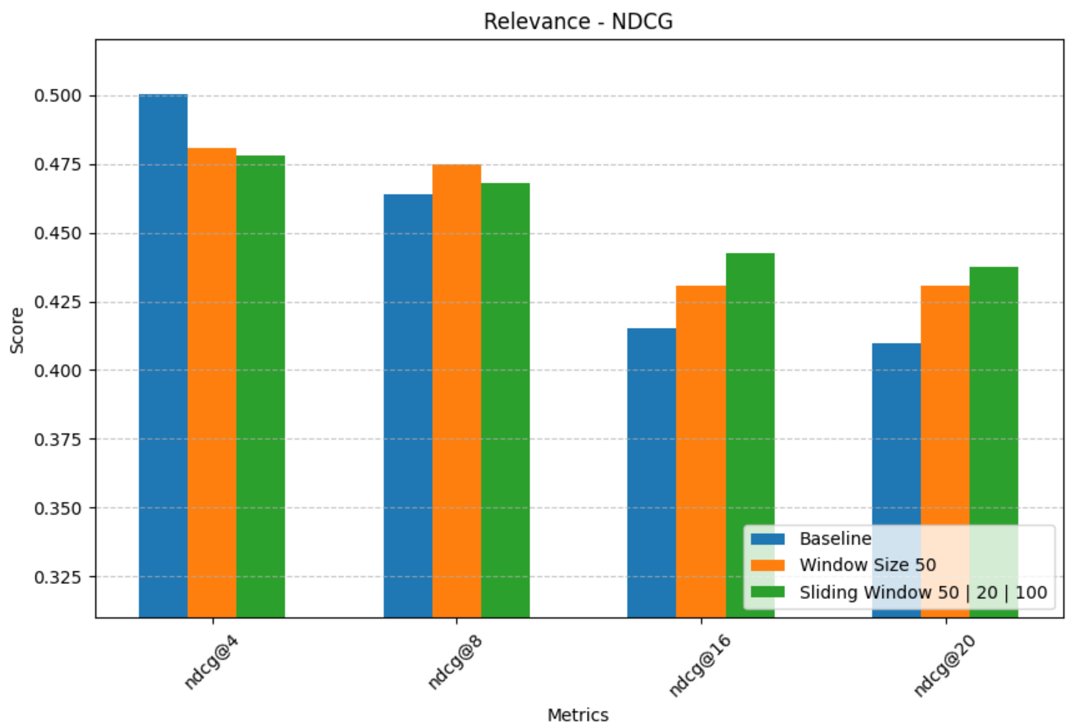
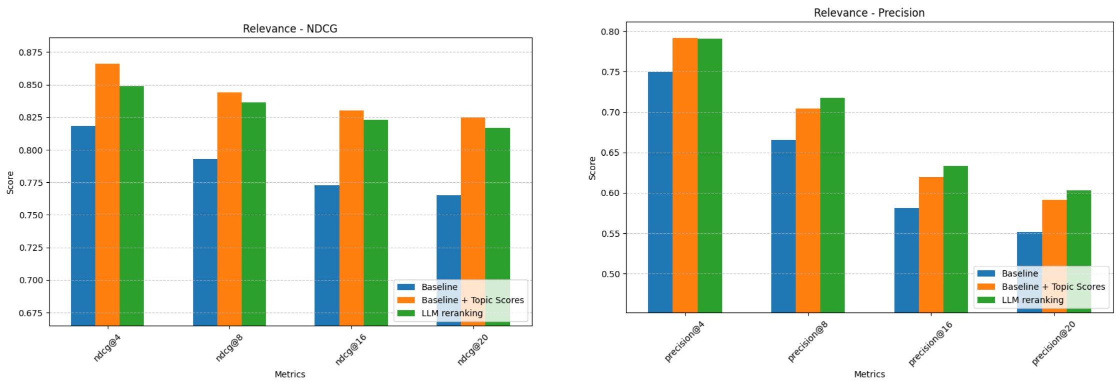


Figure 5.14: NDCG for comparison of the sliding window approach and fixed window sizes

results compared to simple reranking with window size 50. Because for window sizes of around 50 arguments the best results are achieved and because the sliding window approach despite a more complex implementation does not achieve results significantly better or worse compared to the standard reranking approach, a window size of 50 arguments per query is adapted for all following experiments.

Score-based reranking Above, the different variations of *topic scores* and *LLM-based reranking methods* were explored for *experiment 1* on the dev split of the data. In the following, the findings will be applied to the different test splits of the data. For the test-sets 1, 2, and 3 the baseline method is compared against the use of *topic scores* with the baseline (representing the initial retrieval step) and the use of *LLM relevance scores* to rerank the top arguments.



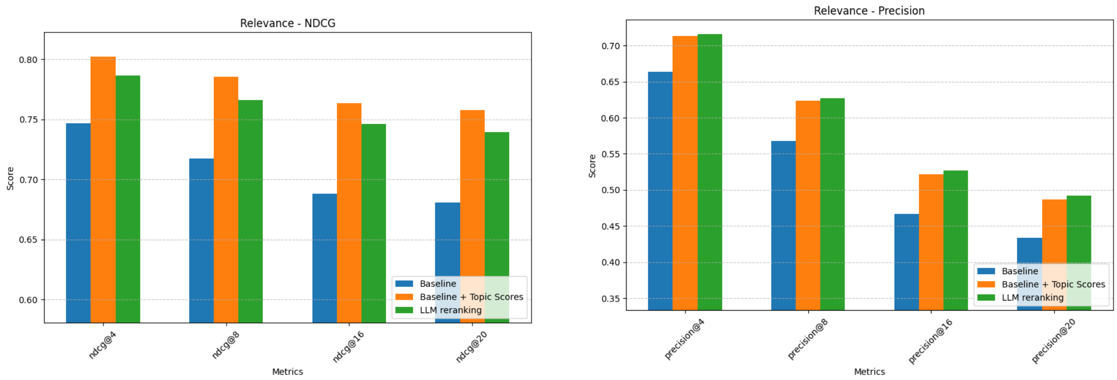
(a) NDCG for the score-based rerank on the first test set

(b) Precision for the score-based rerank on the first test set

Figure 5.15: Relevance metrics for score-based reranking on the first test set

Graphic 5.15 shows the relevance metrics for the comparison on test set 1. The NDCG in Figure 5.15 (a) shows that both the initial retrieval step including the *topic scores* and the final ranking reordered based on the *LLM relevance scores* perform better than the baseline. However, the LLM reranking approach yields worse NDCG results at all ranks compared to the retrieval step using only the baseline and the topic scores. The precision metric in Figure 5.15 (b) gives insights as to where the approaches differ. While the LLM ranking approach has better precision at most ranks, the approach based on the baseline method and the topic scores achieves slightly higher precision at rank 4. This suggests that the *LLM ranking* can benefit the overall ranking especially at lower ranks. However, there seem to be some effects, where the *LLM ranking* includes disproportionately more mistakes at the highest ranks compared to the baseline and the baseline including topic scores.

Similar tendencies compared to those observed on test-set-1 can also be found for test-set-2. While both the combination of baseline method with topic scores and the *LLM ranking* score achieve better relevance metrics than the baseline, the combination of baseline and topic scores achieves higher NDCG at all ranks compared to the *LLM ranking* approach (see graphic 5.16 (a)). However, as graphic 5.16 (b) shows, the precision for the *LLM ranking* approach is higher at all ranks, even at the lowest rank investigated, which is 4. As the NDCG weighs the top ranks higher than lower ranks, these results suggest that the *LLM ranking* approach performs worse compared to the combination



(a) NDCG for the score-based rerank on the second test set

(b) Precision for the score-based rerank on the second test set

Figure 5.16: Relevance metrics for score-based reranking on the second test set

of baseline and topic scores only for the very highest ranking arguments at ranks 1 and 2. This is because these ranks have the most impact on the NDCG value at all ranks it is computed, but the same impact on precision as e.g. ranks 3 and 4 for the computation of the precision metric at rank 4.

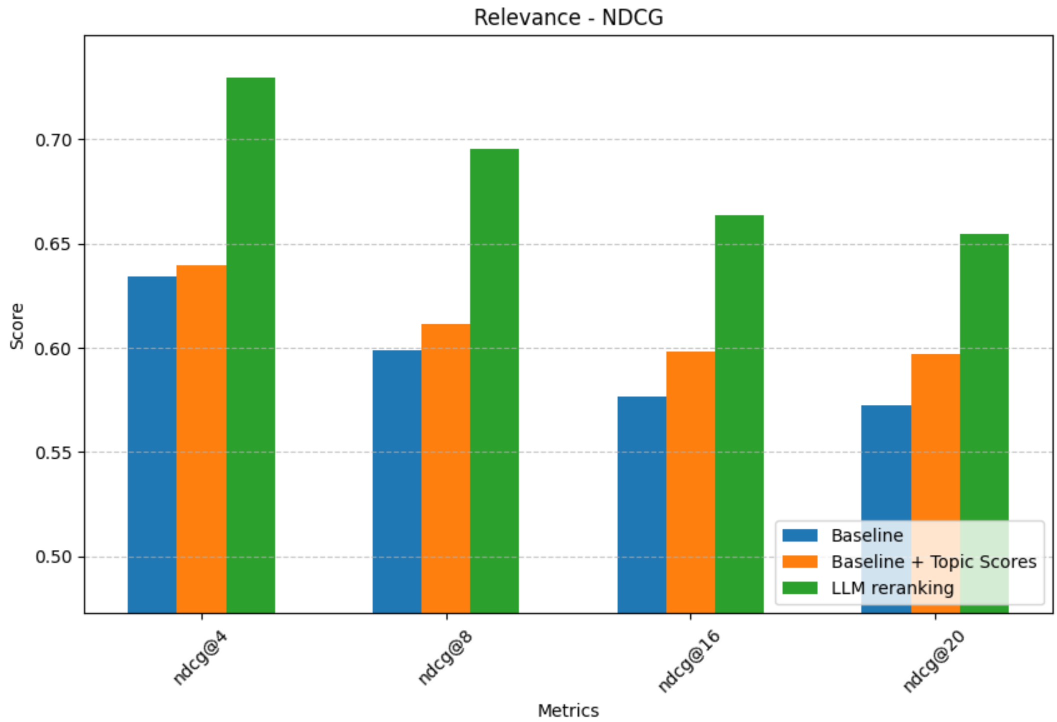


Figure 5.17: NDCG for score-based reranking on the third test set

The results on test set 3 significantly differ from those described for the first two test sets. While on this set the combination of topic scores with the baseline methods yields a slight improvement over the baseline methods results, the usage of *LLM ranking* makes a much greater improvements compared to the other methods at all ranks. Graphic 5.17 shows this for NDCG, a similar effect can be observed for alpha-NDCG and precision as well. All detailed results for test set 3 can be found in appendix A5.

It is worth noting that the overall results for the third test set, even when using the *LLM ranking* approach, are generally lower compared to the results for test sets 1 and 2 across the important metrics. The largest differences can be found concerning the baseline and the combination of baseline and topic scores. The shortcoming of these initial retrieval methods could be a reason for the strong improvement achieved by using the *LLM scores*, because they leave more room for improvement. Also, as aforementioned, the results for test set 1 and 2 indicate, that the *LLM ranking* approach makes disproportionately many mistakes at the highest ranks. This supposed effect, however, might have a smaller impact in comparison to the baseline if the baseline itself yields worse results and makes more mistakes at all ranks.

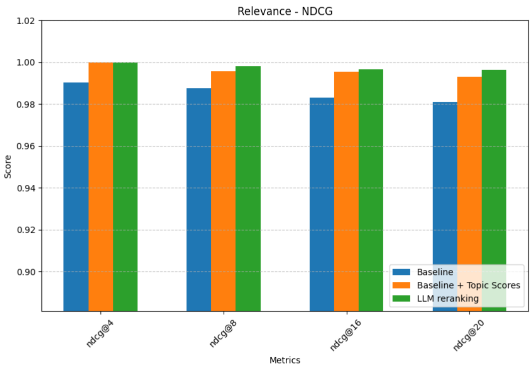
While the test sets 1 and 2 as well as the train and dev set collect arguments from the Swiss elections 2019 and 2023 directly from different politicians, the arguments for test set 3 were collected through an annotation study (Falk et al., 2024). Therefore, the arguments in test set 3 might have a different perspective compared to arguments from the other corpora. The results suggest, that this change of perspective might be a bigger issue concerning the *SBERT similarity scores* or even the *topic scores* than it is when looking at the *LLM relevance scores*. Aside from the change of perspective it is also possible that test set 3 might focus on different topics or differentiate from the other data sets in another way, which might lead to the significant differences found concerning the performance of the *LLM ranking* compared to the combination of baseline method and topic scores.

All three test sets suggest that the *LLM ranking* approach could be helpful to improve results in an argument retrieval process. While the *LLM-based* approach mainly improves results on the higher ranks while sometimes still producing worse results at the top ranks, the *LLM ranking* approach significantly improves the results on all metrics for the third test set, which seems to be more difficult for the SBERT baseline.

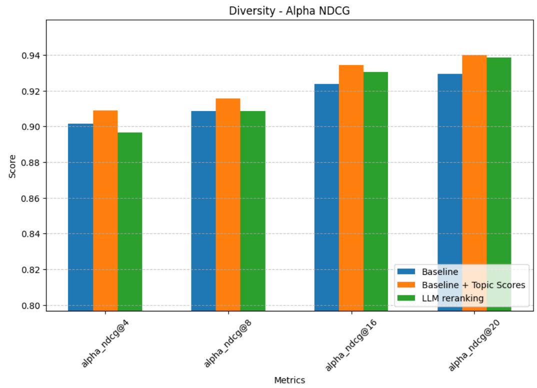
Experiment 2

This section applies the findings concerning the *topic scores* and the *LLM relevance scores* to *experiment 2*, which aims to find arguments that are relevant to a query based on their content alone, without any requirements concerning the arguments' authors' socio-demographic attributes. Therefore, on each of the test sets, the baseline method consisting of only the *SBERT relevance scores* is compared with the initial retrieval step combining the baseline method with the *topic scores*, and the *LLM ranking method*, which uses *LLM relevance scores* to rerank the top 50 arguments from the initial retrieval step for each query. For each subset of data the relevance metrics, NDCG and precision, as well as the diversity metrics, alpha-NDCG and KL-divergence, are computed at ranks 4, 8, 16, and 20.

Graphic 5.18 (a) shows the NDCG results on test set 1. It shows that the *LLM reranking* approach can achieve slightly better results than the baseline methods with topic scores, which in turn achieve better results than the baseline alone. However, for this test set, the differences are marginal, for the NDCG at rank 4 there is no difference, because both the initial retrieval step with baseline and topic scores and the *LLM reranking* achieve the maximum results. Similar observations can be made concerning the precision. Interestingly, the alpha-NDCG value differs significantly from the NDCG in *experiment 2*. The *LLM reranking* achieves worse alpha-NDCG at all ranks compared



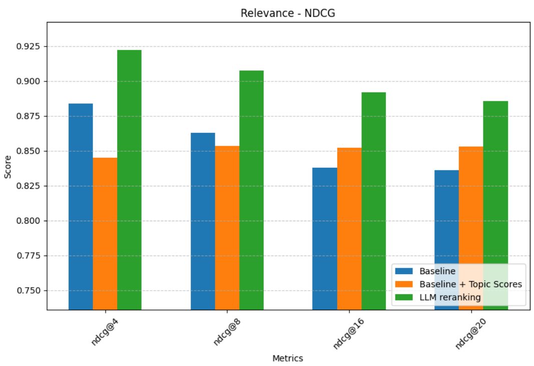
(a) NDCG for score-based reranking on the first test set for experiment 2



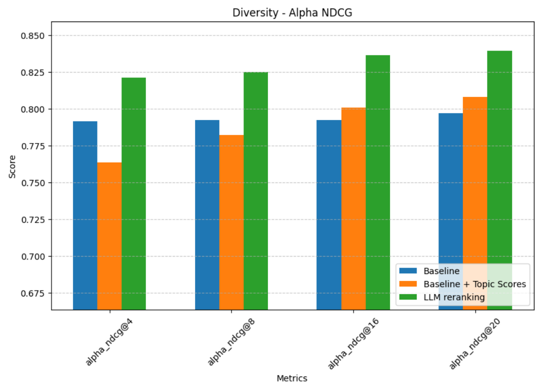
(b) alpha-NDCG for score-based reranking on the first test set for experiment 2

Figure 5.18: NDCG and alpha-NDCG for score-based reranking in experiment 2 on the first test set.

to the baseline with topic scores, at rank 4 even the plain SBERT baseline achieves higher alpha-NDCG than the *LLM reranking* approach.



(a) NDCG for score-based reranking on the second test set for experiment 2

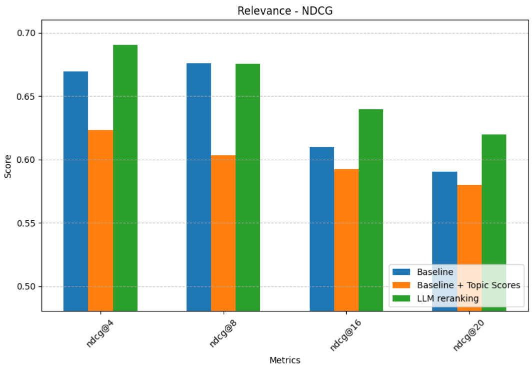


(b) alpa-NDCG for score-based reranking on the second test set for experiment 2

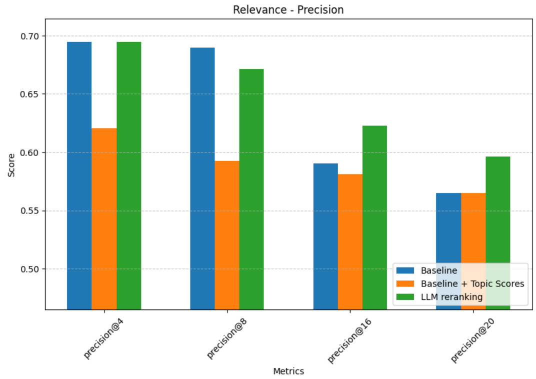
Figure 5.19: NDCG and alpha-NDCG for score-based reranking in experiment 2 on the second test set.

For test set 2 the *LLM ranking* approach achieves better results than both the baseline and the baseline with topic scores across the relevance metrics and alpha-NDCG as shown in graphic 5.19 (a & b). Interestingly, the initial retrieval step including the topic scores performs worse than the SBERT baseline up to rank 8 on both NDCG and alpha-NDCG.

Graphics 5.20 (a & b) show NDCG and precision for experiment 2 on test set 3. On this test set, the inclusion of topic scores performs significantly worse than the standard SBERT baseline at all ranks, while the *LLM ranking* shows similar results compared to the baseline, only showing significantly better results towards the higher ranking arguments for NDCG at 16 and at 20 respectively. As the *LLM relevance scores*, which are the bases of the *LLM reranking* approach, are computed on the initial retrieval step, which is made up of the baseline combined with the topic scores, it is plausible that the weak performance of the topic scores has a negative impact on the *LLM reranking*



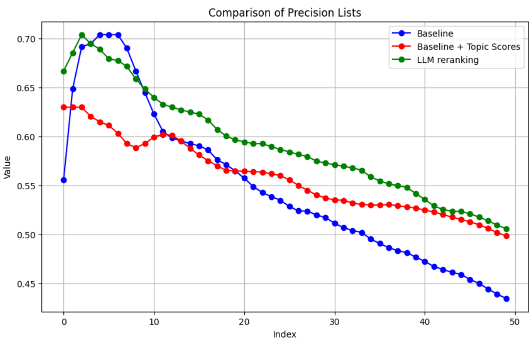
(a) NDCG for score-based reranking on the third test set for experiment 2



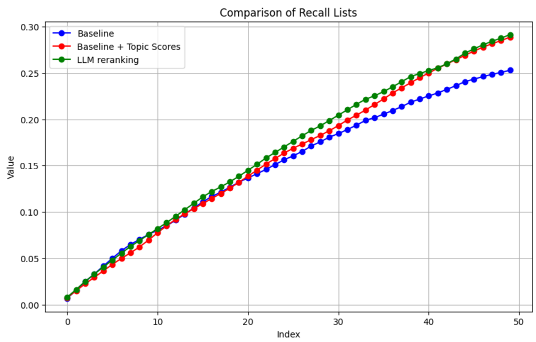
(b) precision for score-based reranking on the third test set for experiment 2

Figure 5.20: NDCG and precision for score-based reranking in experiment 2 on the third test set.

approach as well. Analyzing NDCG and precision at rank 4 even shows an effect opposite to the effect described for *experiment 1* at test sets 1 and 2. While in *experiment 1* an effect was observed, where the LLM had a higher error rate towards the top ranking arguments compared to the baseline with and without the topic scores, the results given in graphics 5.20 (a & b) suggest, that the *LLM ranking* in fact is more precise than the baseline for the very highest ranking arguments, because the NDCG at rank 4 is higher for the *LLM ranking* approach, even though the precision at rank 4 is similar for both approaches.



(a) Average precision for the top 50 arguments per query



(b) Average recall for the top 50 arguments per query

Figure 5.21: Average precision and recall plotted for the top 50 arguments per query

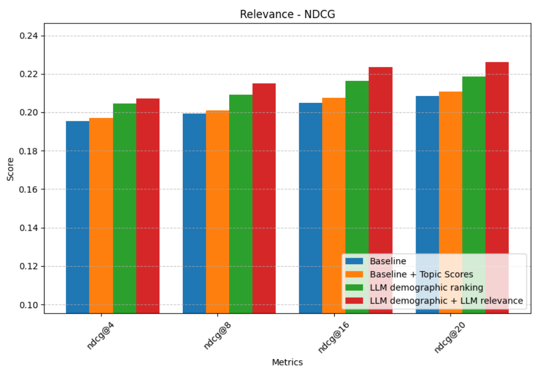
Graphic 5.21 shows the average precision and recall plotted for the top 50 arguments per query. The graphs show significantly higher average precision and recall at rank 50 for the baseline with topic scores compared to the plain sbert baseline. This suggests that the bad performance of the topic scores on test set 3 should not have had an adverse effect on the *LLM reranking*, as the *LLM relevance scores* were computed for 50 arguments with relatively high recall compared to the top 50 arguments of the SBERT baseline.

5.2.3 Research Question 2

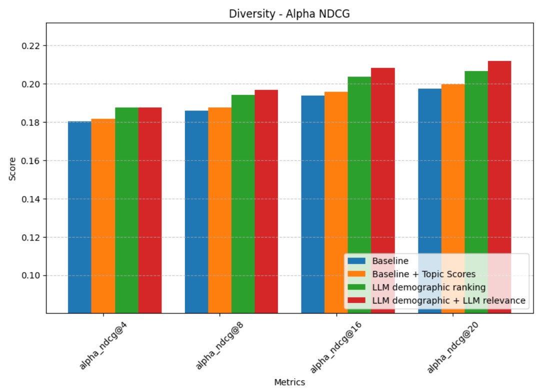
Research question 2 targets the usage of LLMs in an argument retrieval process for an author profiling task. Therefore, the findings from research question 1 will be applied to experiment 3, which involves finding arguments that are relevant to a query based on both the content and the demographic properties of the argument's author. In contrast to experiment 1, for experiment 3 the demographic profile of the author will not be regarded whatsoever - instead, the Demographic LLM scores are used to predict the demographic match based on the content of the argument alone. In the following, this approach in combination with the findings concerning the argument retrieval process from research question 1 will be investigated on experiment 3.

Experiment 3

The approach investigated in *experiment 3* is heavily based on the results concerning the usage of LLMs in in argument retrieval process for relevance based retrieval. Therefore, the relevance-score-based approach evaluated in the following section strongly resembles the relevance score based approach investigated during research question 1. The *Demographic LLM scores* are predicted by an LLM to predict how well an argument's author matches a given socio-demographic property. In the following, a ranking based on these *Demographic LLM scores* will be predicted for each of the test-split of the data. This will be compared to the baseline based on SBERT and to the initial retrieval step including the topic scores and the SBERT baseline. Additionally, for every dataset, a prediction based on both the *Demographic LLM scores* and the *LLM relevance scores* is evaluated, which is created, by adding all scores together and ranking them in descending order. For the evaluation on teach test set the relevance metrics, NDCG and precision, as well as the diversity metrics, alpha-NDCG and KL-divergence, are evaluated.



(a) NDCG for Demographic-LLM-score-based rerank on test set 1



(b) Alpha-NDCG for Demographic-LLM-score-based rerank on test set 1

Figure 5.22: NDCG and alpha-NDCG for Demographic-LLM-score-based rerank on test set 1

Graphic 5.22 shows the results for NDCG (a) and alpha-NDCG (b) for *experiment 3* on the first test set. The ranking based on *Demographic LLM scores* improves the relevance and diversity metrics compared to the baseline with and without topic scores at every rank it is computed. The graphics also show that including both the *Demographic LLM scores* and the *LLM relevance scores* in the reranking step can yield even better results.

Graphic 5.23 shows NDCG for test set 2. For NDCG computed at ranks 16 and 20 there is a slight tendency, where the approaches involving LLM reranking perform better than the baseline approaches. For NDCG at 4 and NDCG at 8, no such tendency can be observed. Even the tendency observed for the higher ranks for test set 2 is based of marginal differences in the scores.

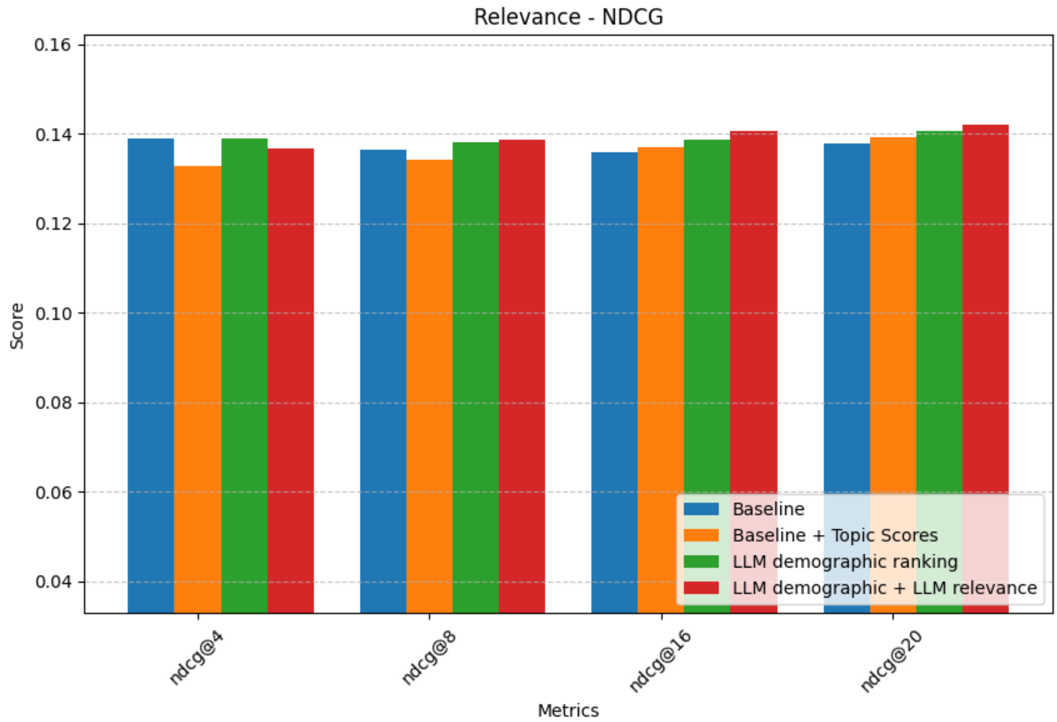
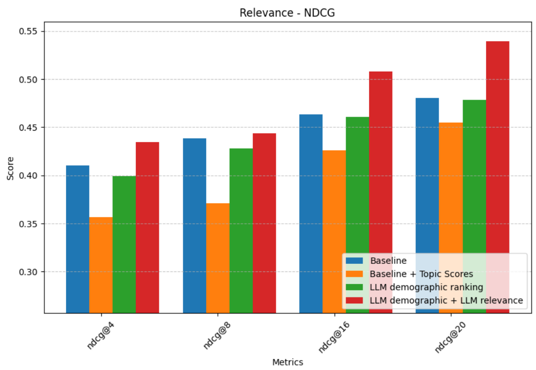
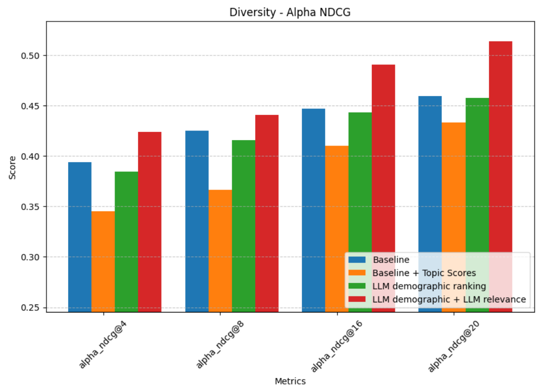


Figure 5.23: NDCG for Demographic-LLM-score-based rerank on test set 2



(a) NDCG for Demographic-LLM-score-based rerank on test set 3



(b) Alpha-NDCG for Demographic-LLM-score-based rerank on test set 3

Figure 5.24: NDCG and alpha-NDCG for Demographic-LLM-score-based rerank on test set 3

Graphic 5.24 shows NDCG (a) and alpha-NDCG (b) for test set 3. As already observed for the third test set in experiment 2, the inclusion of topic scores yields significantly worse results compared to the plain SBERT baseline. The inclusion of *LLM relevance scores* combined with *Demographic LLM scores* improves the results significantly. However, the investigation of experiment 2 on the third test set shows that the *LLM relevance scores* alone can already yield a comparable improvement. As the reranking using *Demographic LLM scores* is performed based on the initial retrieval step, which combines the topic scores with the SBERT baseline, the *Demographic LLM scores* do improve the ranking. However, the final ranking based on the *Demographic LLM scores* still achieves slightly lower results than the baseline, on NDCG, alpha-NDCG and also precision.

Experiment	Weight				
	SBERT	Topic	Demographic	Demographic LLM	LLM Relevance
Experiment 1	8.555	1.356	10.069		1.027
Experiment 2	7.933	0.390			1.966
Experiment 3	1.158	0.713		0.610	
Experiment 4	7.061	4.085		0.082	-0.436

Table 5.2: Weights for different features across multiple experiments based on logistic regression

It can not be shown that the *Demographic LLM scores* investigated as a means to utilize LLMs in demographic relevance prediction can consistently deliver useful information that can be used to improve perspective argument retrieval tasks or fulfill similar purposes. However, the investigation shows that *Demographic LLM scores* and *LLM relevance scores* can possibly be combined to produce valuable results. While the gaps between the results on some of the test sets are marginal, the combination of *Demographic LLM scores* and *LLM relevance scores* can generally be considered the supreme approach, achieving the best results across the majority of the metrics computed.

5.2.4 Research Question 3

Research question 2 shows promising results for the combination of multiple different LLM scores, namely the Demographic LLM scores and the LLM relevance scores. Research question 3 therefore aims to investigate combining the different feature scores available for each experiment. For every experiment, the logistic regression is trained on the train split of the data for all feature scores available. In the following sections the weights resulting from the logreg will be introduced, before the logreg will be applied to each experiment and compared with the methods introduced prior.

Logistic Regression weights

This section will describe the weights that were gained from training the logistic regression and used to calculate a weighted sum of all available feature scores to gain a final prediction. For every prediction the logistic regression gets the feature scores for one query-argument pairing as an input and predicts a binary gold-relevance score as an output. After training, the internal logit scores are used as weights to compute the weighted sum. These logit scores are described for every experiment in this section.

For experiment 1 there are four feature scores available: the *SBERT similarity score*, the *explicit similarity score*, the *topic score*, and the *LLM relevance score.* The logistic regression is trained to predict the gold-score for each query-argument pairing for these four input features. The internal logit scores after training that are used as weights for a weighted sum are shown in table 5.2. For experiment 2 the input features are the *SBERT similarity score*, the *topic score* and the *LLM relevance score.* The logit scores are shown in table 5.2. Experiment 3 has the same three input features as experiment 2, the weights are shown in table 5.2. However, as shown in research question 2, combining the different LLM scores for demographic and content relevance might yield even better results compared to focusing on either one of them. Therefore, for experiment 3, another logistic regression is computed with four input features, as in experiment 1, but it with the *Demographic LLM scores* instead of the *explicit demographic scores*. The input features for experiment 3 are *SBERT relevance score, topic score*, *Demographic LLM score*, and *LLM relevance score*. The weights derived from the logreg are depicted in table 5.2).

The scores show that the *explicit demographic score* as well as the *SBERT similarity score* mostly get assigned significantly higher weights than the other feature scores. The high *explicit demographic scores* lead to them overpowering all other feature scores. This is reasonable as the *explicit demographic scores* are introduced as an alternative to strictly filtering out arguments that do not match the demographic feature. With the high weight and looking at the top-part of the ranking only, the *explicit demographic scores* achieve the same effect as this filtering. The very high weight for the *SBERT similarity scores* shows that the scores are often valuable for predicting relevant arguments. The fact that the weight is oftentimes so much higher than it is for the *topic scores* or the *LLM relevance scores* might be due to the more fine-grained distribution of the *SBERT similarity scores*.

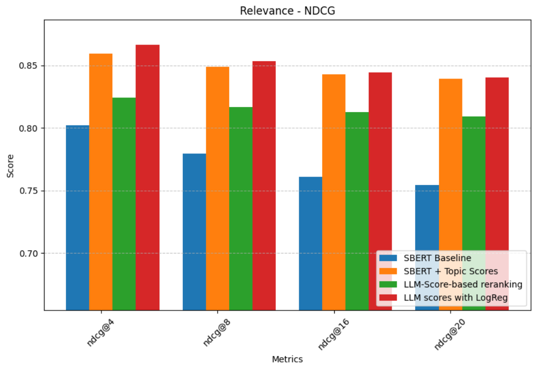
For experiment 3, the weights for the approach including both *Demographic LLM scores* and *LLM relevance scores* show a very small weight for the *Demographic LLM score* and a negative weight for the *LLM relevance score*. However, in research question 2, the combination of both LLM scores, achieved the best results across most of the metrics. The fact, that the logistic regression is trained for all query-argument pairs from the train split of the data can be challenging here, because it means that for every query about 40.000 quadruples of feature scores are regarded, even though both LLM scores were only predicted for 50 arguments per query and the evaluation mostly regards only the top 20 arguments for each query. Brief testing on training the logistic regression only on the top arguments for each query however has shown inconsistent weights that mostly produce even worse results regarding the evaluation metrics. A higher quantity of LLM scores for each query might help producing more sensible weights from the logistic regression but is not further investigated in this thesis.

Experiment 1

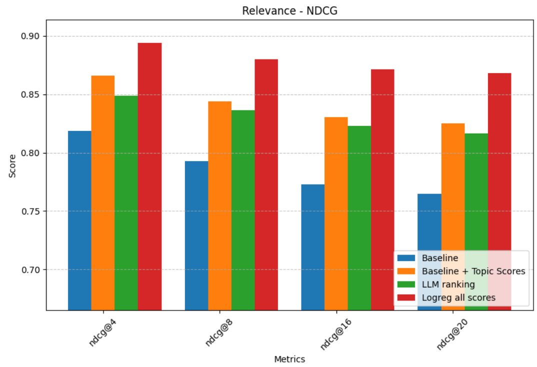
Graphic 5.25 compares the NDCG scores for the SBERT baseline, the baseline with topic scores, and the LLM ranking with the weighted sum using the logistic regression on the dev set (a), test set 1 (b), test set 2 (c), and test set 3 (d). On the dev set as well as test sets 1 and 2 the approach using the logistic regression achieves significantly stronger NDCG scores compared to the LLM ranking approach investigated during research question 1. While the difference between the baseline with topic scores compared to the weighted sum of all scores is relatively small, on test sets 2 and 3 the logistic regression achieves significantly higher NDCG values compared to all other methods investigated. On the third test set, both the LLM ranking investigated in research question 1 and the weighted sum using logistic regression weights achieve significantly stronger NDCG results compared to the baseline with and without topic scores. However, the weighted sum using logistic regression weights achieves lower scores for NDCG computed at ranks 4 and 8 and only achieves similar scores for NDCG at ranks 16 and 20.

Experiment 2

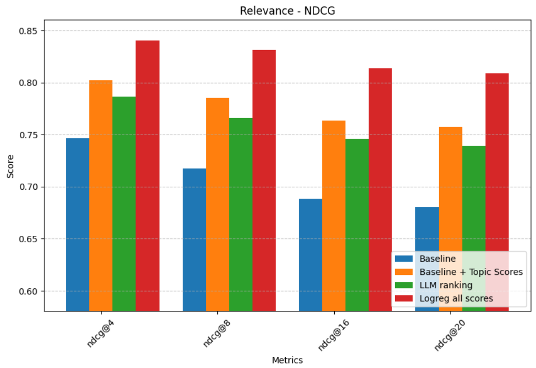
Graphic 5.26 compares the NDCG scores for the SBERT baseline, the baseline with topic scores, and the LLM ranking with the weighted sum using the logistic regression on the dev set (a), test set 1 (b), test set 2 (c), and test set 3 (d). For the dev set and test set



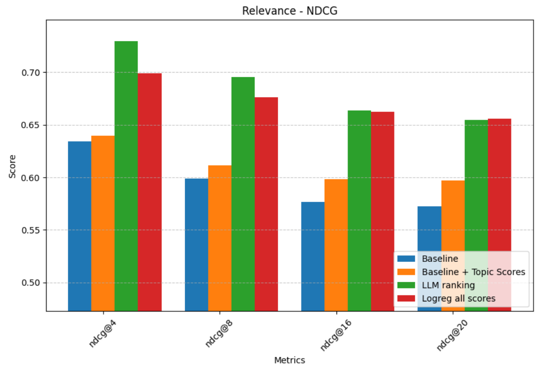
(a) NDCG for weigthed sum on dev set for experiment 1



(b) NDCG for weigthed sum on test set 1 for experiment 1



(c) NDCG for weigthed sum on test set 2 for experiment 1



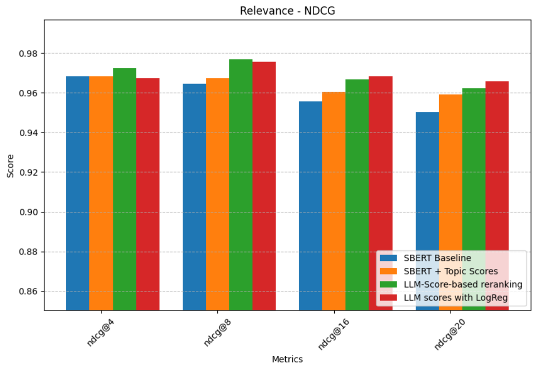
(d) NDCG for weigthed sum on test set 3 for experiment 1

Figure 5.25: NDCG for weigthed sum approach across different sets for experiment 1

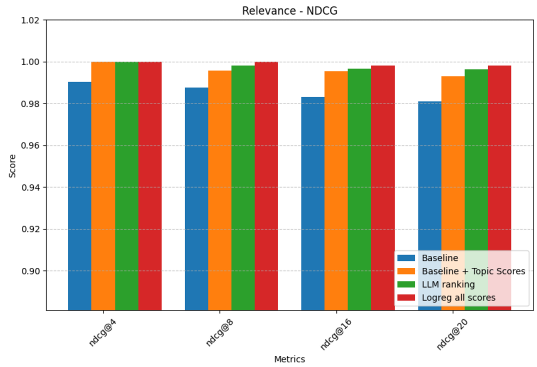
1, the weighted sum mostly achieves the best results, improving even over the LLM ranking approach except for the NDCG at ranks 4 and 8 for the dev set. On test set 2, the weighted sum approach still achieves significantly better results than both baselines, however it scores slightly lower than the LLM ranking approach, especially for NDCG computed at the top ranks. On the third test set the NDCG scores for the weighted sum are significantly lower compared to the LLM ranking approach. For NDCG at ranks 4 and 8, even the SBERT baseline achieves higher scores than the weighted sum approach.

Experiment 3

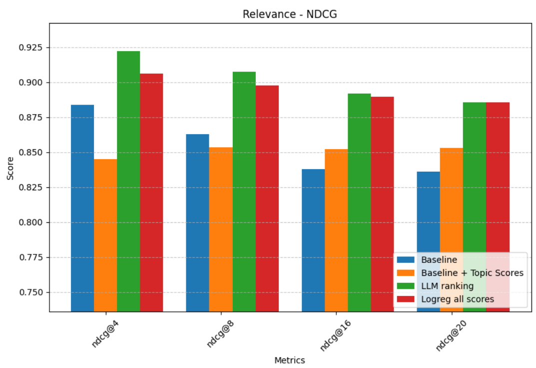
Graphic 5.27 compares the NDCG scores for the SBERT baseline, the baseline with topic scores, and the LLM ranking appoaches with the weighted sum approaches using the logistic regression on the dev set (a), test set 1 (b), test set 2 (c), and test set 3 (d). Both for the LLM ranking approach and for the weighted sum approach, two different variations are tested. The *implicit demographic-score-based rerank* and the *implicit demographic score logreg* include only the *Demographic LLM scores* but not the *LLM relevance scores*. The *LLM demographic + LLM relevance* as well as the *implicit demographic score + LLM rerank score logreg* on the other hand include both the *Demographic LLM scores* and the *LLM relevance scores*. For the dev set and for test set 3 the differences between the results are only marginal, none of the approaches can be identifies as the best across all of the ranks NDCG is computed at. For the dev set however, the approach were



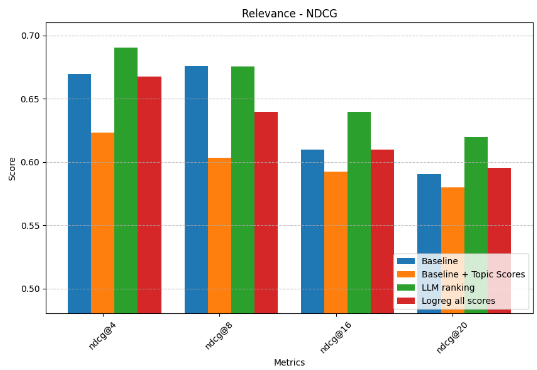
(a) NDCG for weigthed sum on dev set for experiment 2



(b) NDCG for weigthed sum on test set 1 for experiment 2



(c) NDCG for weigthed sum on test set 2 for experiment 2



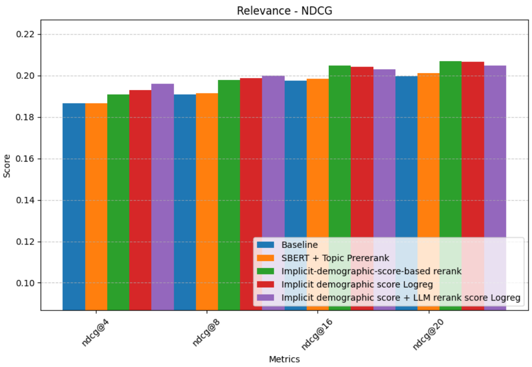
(d) NDCG for weigthed sum on test set 3 for experiment 2

Figure 5.26: NDCG for weigthed sum approach across different sets for experiment 2

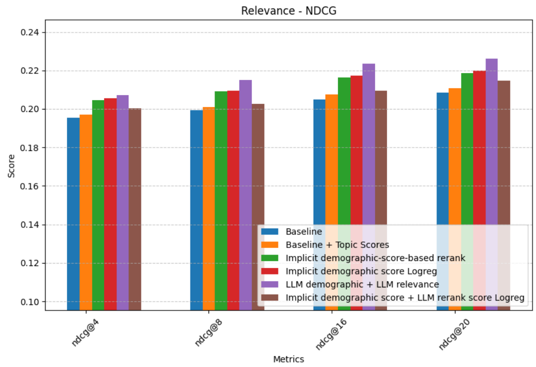
Demographic LLM scores and *LLM relevance scores* are combined is missing, which turns out to be the strongest approach on test sets 1 and 3. The weighted sum combining both LLM scores achieves low scores across the board and is, with the exception of the dev set, consistently beaten by all other approaches involving LLM scores, only beating the baseline methods occasionally.

Overall, the approach using weighted sums with weights based on a logistic regression yields rather mixed results. In the first experiment, the results are promising. However, in the second experiment the weighted sum approach can not provide a clear advantage compared to other methods, while in the third experiment especially the weighted sum including both LLM scores achieves significantly lower results compared to the other approaches involving LLMs.

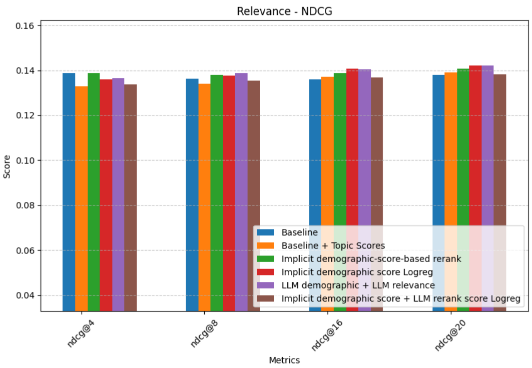
Moreover, the effectiveness of the weighted sum approach varies strongly with the dataset in question. Particularly noticeable are the bad results for the logistic regression approaches on the third test set, while the approaches show stronger performances on the dev set and on test set 1. This is not surprising, because the dev set and test set 1 as well as the train set used to train the logistic regression function are all split from the same original dataset of political arguments from the Swiss election 2019. Test set 2 originates from political arguments from a different Swiss election while test set 3 was created through an annotation process with readers (Falk et al., 2024). The arguments for test set 3 are therefore likely different from the arguments in the train set, leading



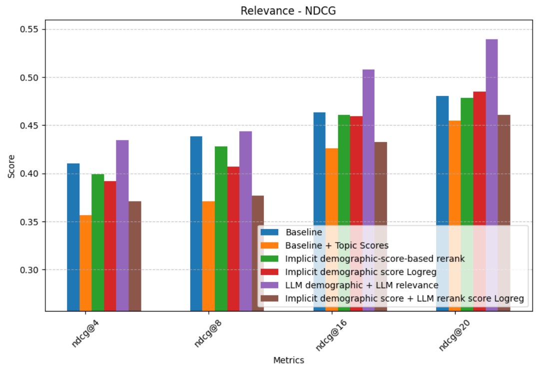
(a) NDCG for weigthed sum on dev set for experiment 3



(b) NDCG for weigthed sum on test set 1 for experiment 3



(c) NDCG for weigthed sum on test set 2 for experiment 3



(d) NDCG for weigthed sum on test set 3 for experiment 3

Figure 5.27: NDCG for weigthed sum approach across different sets for experiment 3

to a different effectiveness of the individual feature scores. This of course impairs the performance of the logistic regression approach for the third test set.

5.2.5 Shared Task Participation

In the context of this thesis, participation in the Perspective Argument Retrieval shared task was achieved (Günzler et al., 2024) in May 2024. For each of the shared task's three scenarios, a system was submitted. Each system was built on the pipeline introduced in this thesis, where scenario 1 of the shared task uses the system described for experiment 2, scenario 2 corresponds to experiment 1 and scenario 3 corresponds to experiment 3 described in this thesis. However, since the final submission for the shared task was made in May 2024, not all methods described and evaluated in this thesis were taken into consideration during the submission for the shared task.

The submissions to the shared task are based on feature scores. Scenario 1 of the shared task uses *SBERT similarity scores*, *relative topic scores*, and *LLM relevance scores*. Scenario 2 of the shared task uses *SBERT similarity scores*, *relative topic scores*, *explicit demographic scores* as well as *LLM relevance scores*. For scenario 3, *SBERT similarity scores*, *relative topic scores*, and *Demographic LLM scores* are used. For each of the three scenarios, a weighted sum of all available scores is computed to calculate the final ranking of arguments for each query. All scores used in the submissions for the shared task are computed exactly as decribed in this thesis. However, due to the

Rank	Team	Relevance		Diversity	
		Mean Rank	Mean NDCG	Mean Rank	Mean αNDCG@k
1	twente-bms-nlp (top-1)	1.33	0.707	1.67	0.672
2	**Sövereign (top-2)**	**2.22**	**0.632**	**2.22**	**0.601**
5	sbert_baseline	5.0	0.445	5.0	0.419
8	bm25_baseline	7.67	0.195	8.00	0.185

Table 5.3: Average results on all test sets and scenarios. We present the results for the baseline and the model that achieved better performance for comparison.

team	Relevance			Diversity		
	Rank	NDCG	Precision	Rank	αNDCG	klDiv
Test set 1						
sövereign	*1*	***0.999***	***0.999***	*1*	***0.922***	*0.143*
twente-bms-nlp	2	0.987	0.989	5	0.910	0.142
GESIS-DSM	3	0.986	0.983	2	0.916	0.124
sbert_baseline	3	0.986	0.983	3	0.916	0.125
bm25_baseline	7	0.651	0.613	8	0.629	**0.121**
Test set 2						
twente-bms-nlp	1	**0.936**	**0.930**	1	**0.870**	**0.115**
sövereign	*3*	*0.895*	*0.888*	*3*	*0.827*	*0.135*
sbert_baseline	5	0.855	0.848	5	0.793	0.118
bm25_baseline	7	0.737	0.722	8	0.690	0.122
Test set 3						
twente-bms-nlp	1	**0.944**	**0.938**	1	**0.880**	0.213
sbert_baseline	4	0.637	0.635	5	0.593	0.153
sövereign	*5*	*0.628*	*0.614*	*4*	*0.595*	*0.161*
bm25_baseline	7	0.368	0.372	8	0.342	**0.152**

Table 5.4: Average results for Scenario 1 on all test sets.

early stage of the research in which the methods were submitted to the shared task, the topic score versions using different weights inluding *linear weighted topic scores*, *hyperbolical weighted topic scores* as well as *SBERT weighted topic scores* had not been further investigated. Also, the usage of both *LLM relevance scores* and *Demographic LLM scores* at the same time for the third scenario had not been investigated, which is why the submissions only include the *Demographic LLM scores*.

The shared task evaluates NDCG and precision as relevance metrics as well as alpha-NDCG and KL-divergence for diversity. All metrics are computed at ranks 4, 8, 16, and 20. The final ranking of the submissions in the shared task is based on NDCG and alpha-NDCG alone. Averaged across the three scenarios and across the three test sets, the submission this thesis is based around reached second place in the shared task (see table 5.3).

For scenario 1 the submission achieved the top results on test set 1 for both NDCG and alpha-NDCG, reaching a near perfect NDCG score of 0.999 averaged across all ranks. On test set 2 the submission achieved the third place in the overall rankings for relevance and diversity, still beating the SBERT baseline on both metrics. On test set

team	Relevance			Diversity		
	Rank	NDCG	Precision	Rank	αNDCG	klDiv
Test set 1						
twente-bms-nlp	1	**0.895**	**0.717**	1	**0.852**	0.181
sövereign	*2*	*0.878*	*0.707*	*2*	*0.844*	*0.181*
sbert_baseline	5	0.222	0.218	5	0.208	**0.139**
Test set 2						
sövereign	*1*	***0.823***	***0.623***	*1*	***0.794***	*0.166*
twente-bms-nlp	2	0.798	0.610	2	0.771	0.165
sbert_baseline	5	0.148	0.140	5	0.142	**0.124**
Test set 3						
twente-bms-nlp	1	**0.798**	**0.613**	1	**0.793**	0.256
sövereign	*2*	*0.673*	*0.504*	*2*	*0.675*	*0.221*
sbert_baseline	6	0.406	0.339	6	0.400	**0.163**

Table 5.5: Average results for Scenario 2 on all test sets.

team	Relevance			Diversity		
	Rank	NDCG	Precision	Rank	αNDCG	klDiv
Test set 1						
sövereign	*1*	***0.213***	***0.211***	*1*	***0.199***	*0.135*
twente-bms-nlp	2	0.203	0.202	2	0.190	**0.124**
sbert_baseline	3	0.202	0.201	4	0.189	0.125
Test set 2						
twente-bms-nlp	1	**0.149**	**0.144**	1	**0.143**	**0.121**
sövereign	*2*	*0.139*	*0.136*	*3*	*0.132*	*0.125*
sbert_baseline	4	0.136	0.129	4	0.131	0122
Test set 3						
twente-bms-nlp	1	**0.655**	**0.560**	1	**0.636**	0.189
sövereign	*3*	*0.436*	*0.365*	*3*	*0.425*	*0.160*
sbert_baseline	5	0.409	0.349	5	0.397	**0.158**

Table 5.6: Average results for Scenario 3 on all test sets.

3 the submission reached fourth place in the diversity ranking and only fifth place in the relevance ranking based on NDCG, achieving a worse averaged NDCG score than the SBERT baseline (see table 5.4). For scenario 2 the submission reaches second place in the relevance and diversity rankings for both the first and the third test set. For the second test set, the submission achieved the best results for both diversity and relevance based on NDCG and alpha-NDCG (see table 5.5). For the third scenario the submission reaches first place for both relevance and diversity on the first test set, it reaches second in relevance and third in diversity on the second test set and it reaches third in both relevance and diversity for the third test set (see table 5.6).

Overall, the submitted system manages to beat the SBERT baseline provided by the shared task organizers on every scenario and every test set, the only exception being the relevance metrics for scenario 1 on test set 3. Generally speaking, the best results compared to other competitors are achieved on the first test set, ranking first in scenarios 1 and 3 and second in scenario 2. The worst results are achieved on test set 3, reaching second for scenario 2, third for scenario 3 and fifth based on relevance metrics for scenario 1. These findings correlate with the observations made for the weighted sum approach in research question 3, where the weighted sum approach achieves particularly bad results across all scenarios for the third test set

Regarding the scenarios, the best performance can be observed for scenario 2, which corresponds to experiment 1 from this thesis. In this scenario the submitted system ranked first on test set 2 and second on test sets 1 and 3. This is also not surprising as most of the feature scores were developed and initially tested in experiment 1 (scenario 2) before they were transferred to the other experiment, so naturally they will achieve the best results in the task they were developed on. However, the fact that the pipeline manages to beat the baseline on nearly every single scenario and dataset shows the effectiveness of the scores and can thus be considered a success.

As aforementioned, the submitted system does not fully reflect the results from this thesis as further investigation was conducted after the final submission of the systems to the shared task. However, even if the submission had included the best approach for every single scenario and dataset regarded, only some minor changes to the rankings would occur. One of these changes would concern the third test set for scenario 1 (experiment 2), where the best approach investigated in this thesis does actually perform better than the SBERT baseline.

6 Conclusions & Future Work

So far in this thesis, the research questions have been introduced, and the methods used to investigate them have been explained and evaluated. The following chapter revisits each research question to summarize the core findings before examining the study's limitations, suggesting directions for future work, and providing a final conclusion.

6.1 Research Questions Revisited

Throughout this thesis, three research questions were investigated in the context of argument retrieval. The research questions focus on predicting content relevance using LLMs, predicting demographic relevance using LLMs, and effectively combining different approaches and measures. This section aims to summarize the core findings for each of these questions.

6.1.1 Research Question 1

How can LLMs be used to enhance the results of an argument retrieval process?

Due to the high computational costs, LLMs are employed exclusively in a re-ranking step within a two-step retrieval pipeline in this study. The initial retrieval step, based on SBERT similarity scores and topic scores, determines the top-ranked arguments, which are subsequently re-ranked by the LLM. Different topic scores for the initial retrieval step are investigated, as the retrieval step directly impacts what arguments are available for re-ranking and is therefore crucial across all research questions and experiments. The results show that relative topic scores significantly improve the SBERT baseline, leading to their adoption in this study.

For the re-ranking step, the effectiveness of two re-ranking strategies is explored, including list-based ranking, where the LLM directly orders the retrieved arguments, and score-based ranking, where the LLM assigns relevance scores to each argument, which are then used to construct the final ranking.

The evaluation reveals that both approaches improve retrieval effectiveness compared to the baseline. The list-based ranking slightly outperforms the score-based ranking in direct comparisons; however, due to its flexibility and better integration with other scoring mechanisms, the score-based ranking approach is ultimately chosen for further experiments.

The number of arguments passed to the LLM for re-ranking also influences performance. While increasing the number of arguments generally leads to better results, this effect plateaus beyond a certain threshold. The best results are achieved with a fixed window size of around 50 arguments, leading to the decision to re-rank the top 50 retrieved arguments in all subsequent experiments. Sliding window approaches, where the model is prompted with different argument subsets at different ranks, do not yield significant improvements.

The findings are applied to three different test sets, showing varying effects. For test sets 1 and 2, topic scores significantly improve the SBERT baseline. While NDCG slightly decreases with re-ranking, higher precision at broader ranks suggests that LLMs capture useful relevance signals, though they struggle with top-ranking arguments. For test set 3, topic scores provide only minor improvements, likely due to ambiguity concerning the topic of some queries, whereas LLM relevance scores achieve a notable improvement, demonstrating their potential when topic-based retrieval is less effective.

Overall, these findings indicate that LLMs can effectively enhance argument retrieval by re-ranking retrieved arguments based on contextual relevance. Score-based ranking offers the best balance between effectiveness and adaptability.

6.1.2 Research Question 2

How can LLMs be used to implicitly predict the demographic or sociocultural perspective with only little text input?

The use of LLMs to predict demographic relevance based solely on argument content yields inconsistent results. While demographic LLM scores can, in some cases, improve retrieval performance, their standalone application in the re-ranking step does not achieve the expected success. Their impact varies across different test sets, with improvements in some cases but a decline in ranking quality in others.

However, when demographic LLM scores are combined with LLM relevance scores, the results are more promising. This suggests that demographic LLM scores may contribute useful information when integrated with content-based relevance but are not reliable enough to serve as an independent ranking criterion. Their effectiveness depends on how they are used within the retrieval process, reinforcing the need for sophisticated integration rather than a standalone application.

6.1.3 Research Question 3

How can LLM predictions on argument relevance and perspective relevance effectively be combined to retrieve relevant arguments for a given question or topic with additional sociocultural or demographic aspects?

The analysis for research question 3 investigates the combination of multiple feature scores, including demographic LLM scores, LLM relevance scores, topic scores, and SBERT similarity scores, to improve argument retrieval, using a weighted sum based on weights from a logistic regression.

The results indicate that the weighted sum approach produces mixed outcomes. In some cases, particularly in experiment 1, the approach improves retrieval effectiveness compared to methods using individual scores. However, in experiment 2, the benefits are less clear, and in experiment 3, the weighted sum approach—especially when combining both LLM scores—often underperforms relative to other re-ranking strategies. The approach is also highly dependent on dataset characteristics. Performance is notably worse on test set 3, likely due to differences between the training and test data, whereas it is more effective on test sets that are more similar to the training distribution.

Overall, while combining multiple feature scores through logistic regression can enhance retrieval in certain scenarios, its effectiveness is inconsistent. The findings suggest that the effectiveness of individual feature scores varies depending on the dataset, and simple linear weighting may not be the optimal method for integrating different signals in argument retrieval. In experiment 3, the combination of demographic LLM scores and LLM relevance scores without logistic regression performs well, while the weighted sum approach with logistic regression leads to significantly lower results. This suggests that the scores themselves hold value, but logistic regression may not be the ideal method for integrating them across all scenarios. The weights assigned by logistic regression in this case, particularly the negative weight for LLM relevance scores, indicate that the learned weighting is not always sensible, and refining how these scores are combined could further improve the performance.

6.2 Limitations

In the following, some fundamental limitations affecting this study will be outlined.

First, bias in LLM predictions remains a challenge, as LLMs inherit biases from their training data. This can influence both relevance scoring and demographic predictions, potentially impairing retrieval results.

Second, the quality of the training data directly impacts the effectiveness of models such as the logistic regression approach. While the training data itself is not necessarily of low quality, it differs significantly from some of the test data, particularly test set 3, which originates from a different source and perspective. This mismatch affects the generalizability of the trained logistic regression model and contributes to its lower performance on that test set.

Third, the scope of the evaluation is limited to specific datasets and argument structures, meaning the findings may not generalize to other retrieval tasks or domains. For example, the topic scores used in the initial retrieval step improve results significantly for some datasets but rely on a dataset-specific topic feature that may not be available in other argument retrieval tasks. The evaluation may also be limited by the choice of metrics, particularly for diversity. While alpha_NDCG accounts for both relevance and diversity, isolating diversity alone is difficult. On the other hand, the applicability of KL divergence for this task has already been questioned in this thesis.

Finally, the fixed retrieval pipeline constrains flexibility in how arguments are retrieved and re-ranked. The two-step approach assumes a predefined initial retrieval step, which may not be optimal for all queries. Again, the third test set serves as an example for this, since the topic scores are significantly less effective on that set, which might impair the performance of the initial retrieval step. A more adaptive retrieval method could further enhance performance.

These limitations highlight broader challenges in applying LLMs for argument retrieval and should be considered when interpreting the results.

6.3 Suggestions for Future Work

In this section some suggestions for possible directions of future work are made.

One important direction for future work is improving the combination of feature scores. The logistic regression approach used in this study shows potential but is also inconsistent, particularly when applied to test sets that differ significantly from the training set. More adaptable approaches could address this issue, such as neural networks or other machine learning models that dynamically adjust score weightings based on the query characteristics or on the distributions of the different feature scores. An adaptive weighting mechanism could help improve ranking effectiveness across different datasets.

Another promising direction is exploring more different LLM-based scoring approaches. The re-ranking step could benefit from more sophisticated LLM prompting techniques, such as few-shot learning, where the model is provided with examples to improve scoring accuracy. Additionally, fine-tuning an LLM on argument retrieval tasks or using pre-trained models specifically designed for retrieval could lead to better results. Instead of relying solely on large general-purpose LLMs, future work could also explore smaller, fine-tuned models that are optimized for argument ranking.

Lastly, addressing dataset limitations could improve both model training and evaluation. The logistic regression approach struggled on test set 3 because the training strongly differs from that test set. Training on a broader, more diverse dataset could help mitigate this issue and make the ranking model more robust. Similarly, future work could explore evaluating argument retrieval on different types of datasets beyond political arguments to test generalizability. However, finding suitable annotated datasets remains a challenge. Especially problematic is the fact that the topic scores used in the initial retrieval step rely on the annotation of all arguments with such a topic, which is not guaranteed when generalizing the approach to other datasets.

6.4 Conclusion

The methods investigated in this thesis demonstrate on the given data that argument retrieval based on content relevance can be improved by incorporating the dataset-specific topic scores in the initial retrieval step and by using LLMs for content relevance prediction. While these approaches yield notable improvements, the demographic relevance prediction, although promising in certain scenarios on some test sets, shows inconsistent results across the different test sets. Similarly, the weighted sum approach based on logistic regression offers a sophisticated means to combine multiple feature scores, yet it fails to produce strong results consistently across all datasets. These inconsistencies indicate that a more refined approach is needed to exploit the full potential of the methods and approaches investigated throughout this thesis. However, the strong performance in the shared task - where the baseline was consistently outperformed, achieving the second place overall - demonstrates that the methods investigated indeed hold practical value.

Appendices

A

Additional Material

Contents

A.1 Prompts

A.1.1 List-based ranking based on content relevance

<<SYS>>Answer with a python list containing all ranked argument ids<</SYS>

[INST]The following are passages related to question **<query text>** [/INST]

[0] **<1st argument text>** ... [49] **<50th argument text>**

[INST]Rank these passages based on their relevance to the question.[/INST]

A.1.2 List-based ranking based on demographic relevance

<<SYS>>Answer with a python list containing all ranked argument ids<</SYS>

[INST]The task is to rank arguments, if they fit the sociocultural property: **<query demographic property>**.[/INST]

[0] **<1st argument text>** ... [49] **<50th argument text>**

[INST]Rank these passages based on their relevance to the sociocultural property.[/INST]

A.1.3 Retrieving LLM relevance scores

<<SYS>>Answer with a python dictionary containing a score between 0 and 1 for each argument id<</SYS>

[INST]Given the question **<query text>** and a list of arguments with IDs. The task is to rank the arguments according to the question. The higher the score the more relevant it is to the question[/INST]

[0] **<1st argument text>** ... [49] **<50th argument text>**

[INST]Return a python dict with every single argument id and the scores only! No text!!! e.g. 1: 0.9, 2: 0.3[/INST]

A.1.4 Retrieving Demographic LLM scores

<<SYS>>Answer with a python dictionary containing a score between 0 and 1 for each argument id<</SYS>

[INST]The task is to rank arguments, if they fit the sociocultural property: **<query demographic property>**[/INST]

[0] **<1st argument text>** ... [49] **<50th argument text>**

[INST]Return a python dict with all argument IDs between 0 and 49 and a score between 0 if the argument does not fit the demographic and 1 if it fits very well.[/INST]

A.1.5 Predict topic prompt

Given a question or an argument, classify it into one of the provided topics.

Question/Argument: <text>

[0: **<topic 0>**] ... [n: **<topic n>**]

Return the integer id of the most relevant topic only.

References

Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems* 226:107134. (Cited on page 1).

Christopher M. Bishop. 2007. Pattern recognition and machine learning [in eng]. 5. (corr. print.) XX, 738 S. Information science and statistics. Literaturverz. S. 711 - 728. New York [u.a.]: Springer. (Cited on page 4).

Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Shahbaz Syed, Timon Gurcke, Meriem Beloucif, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2022. Overview of Touché 2022: Argument Retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction,* edited by Alberto Barrón-Cedeño, Giovanni Da San Martino, Mirko Degli Esposti, Fabrizio Sebastiani, Craig Macdonald, Gabriella Pasi, Allan Hanbury, Martin Potthast, Guglielmo Faggioli, and Nicola Ferro, 311–336. Cham: Springer International Publishing. (Cited on page 2).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv: 2005.14165 `[cs.CL]`. (Cited on page 7).

Matteo Cardaioli, Pallavi Kaliyar, Pasquale Capuozzo, Mauro Conti, Giuseppe Sartori, and Merylin Monaro. 2020. Predicting Twitter Users' Political Orientation: An Application to the Italian Political Scenario [in English]. In *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020,* edited by Martin Atzmuller, Michele Coscia, and Rokia Missaoui, 159–165. Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020. United States: IEEE. (Cited on page 2).

Carlos Castillo. 2019. Fairness and Transparency in Ranking. *SIGIR Forum* (New York, NY, USA) 52, no. 2 (January): 64–71. (Cited on page 10).

Hongyu Chen, Michael Roth, and Agnieszka Falenska. 2024. What Can Go Wrong in Authorship Profiling: Cross-Domain Analysis of Gender and Age Prediction. In *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP),* edited by Agnieszka Faleńska, Christine Basta, Marta Costa-jussà, Seraphina Goldfarb-Tarrant, and Debora Nozza, 150–166. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on page 6).

Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 659–666. SIGIR '08. Singapore, Singapore: Association for Computing Machinery. (Cited on page 11).

Przemysław Czuba. 2023. Reinforcement Learning in Algorithmic Trading: A Survey. In *Theory and Practice in Modern Computing: From Artificial Intelligence to Computer Security.* January. (Cited on page 5).

Van Dang, Michael Bendersky, and W. Croft. 2013. Two-Stage Learning to Rank for Information Retrieval. In *Advances in Information Retrieval,* 7814:423–434. March. (Cited on pages 6, 26).

Esther David, Maayan Zhitomirsky-Geffet, Moshe Koppel, and Hodaya Uzan. 2016. Utilizing Facebook pages of the political parties to automatically predict the political orientation of Facebook users. *Online Inf. Rev.* 40 (5): 610–623. (Cited on page 2).

Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. 2022. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing* 503:92–108. (Cited on page 7).

Neele Falk, Andreas Waldis, and Iryna Gurevych. 2024. Overview of PerpectiveArg2024 The First Shared Task on Perspective Argument Retrieval. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024),* edited by Yamen Ajjour, Roy Bar-Haim, Roxanne El Baff, Zhexiong Liu, and Gabriella Skitalinskaya, 130–149. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on pages 2, 12, 15, 30 sq., 35, 37, 43, 49, 57).

Robert Günzler, Özge Sevgili, Steffen Remus, Chris Biemann, and Irina Nikishina. 2024. Sövereign at The Perspective Argument Retrieval Shared Task 2024: Using LLMs with Argument Mining. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024),* edited by Yamen Ajjour, Roy Bar-Haim, Roxanne El Baff, Zhexiong Liu, and Gabriella Skitalinskaya, 150–158. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on pages 30 sq., 58).

Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics* (Cambridge, MA) 43, no. 1 (April): 125–179. (Cited on page 6).

Yaakov HaCohen-Kerner. 2022. Survey on profiling age and gender of text authors. *Expert Systems with Applications* 199:117140. (Cited on page 2).

Kailash A. Hambarde and Hugo Proença. 2023. Information Retrieval: Recent Advances and Beyond. *IEEE Access* 11:76581–76604. (Cited on page 6).

D. J. Hand and W. E. Henley. 2007. Statistical Classification Methods in Consumer Credit Scoring: A Review. *Journal of the Royal Statistical Society Series A: Statistics in Society* 160, no. 3 (July): 523–541. eprint: https://academic.oup.com/jrsssa/article-pdf/160/3/523/49760733/jrsssa_160_3_523.pdf. (Cited on page 5).

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine* 29 (6): 82–97. (Cited on page 5).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv: 2310.06825 `[cs.CL]`. (Cited on page 7).

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv: 2401.04088 `[cs.LG]`. (Cited on pages 7, 21, 32).

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine. arXiv: 2301.08745 `[cs.CL]`. (Cited on page 2).

M. I. Jordan and T. M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science* 349 (6245): 255–260. eprint: https://www.science.org/doi/pdf/10.1126/science.aaa8415. (Cited on page 4).

Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd. Online manuscript released January 12, 2025. (Cited on pages 5, 8).

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. 1996. Reinforcement learning: a survey. *J. Artif. Int. Res.* (El Segundo, CA, USA) 4, no. 1 (May): 237–285. (Cited on page 5).

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 6769–6781. Online: Association for Computational Linguistics, November. (Cited on pages 6, 17).

Ravpreet Kaur and Sarbjeet Singh. 2023. A comprehensive review of object detection with deep learning. *Digital Signal Processing* 132:103812. (Cited on page 5).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* 25 (January). (Cited on page 5).

John Lawrence and Chris Reed. 2019. Argument Mining: A Survey. *Computational Linguistics* (Cambridge, MA) 45, no. 4 (December): 765–818. (Cited on page 1).

Adam Lopez. 2008. Statistical machine translation. *ACM Comput. Surv.* (New York, NY, USA) 40, no. 3 (August). (Cited on page 5).

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. Edited by Brian Roark and Ani Nenkova. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 9:329–345. (Cited on pages 2, 17).

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023.* Association for Computational Linguistics. (Cited on page 2).

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press. (Cited on page 8).

Maximilian Maurer, Julia Romberg, Myrthe Reuver, Negash Weldekiros, and Gabriella Lapesa. 2024. GESIS-DSM at PerpectiveArg2024: A Matter of Style? Socio-Cultural Differences in Argumentation. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024),* edited by Yamen Ajjour, Roy Bar-Haim, Roxanne El Baff, Zhexiong Liu, and Gabriella Skitalinskaya, 169–181. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on page 13).

T.M. Mitchell. 1997. Machine Learning. McGraw-Hill International Editions. McGraw-Hill. (Cited on page 4).

Sarra Ouni, Fethi Fkih, and Mohamed Nazih Omri. 2023. A survey of machine learning-based author profiling from texts analysis in social networks. *Multimedia Tools Appl.* (USA) 82, no. 24 (March): 36653–36686. (Cited on page 2).

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval* 2 (1–2): 1–135. (Cited on page 5).

Sachin Pathiyan Cherumanal, Damiano Spina, Falk Scholer, and W. Bruce Croft. 2021. Evaluating Fairness in Argument Retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management,* 3363–3367. CIKM '21. Virtual Event, Queensland, Australia: Association for Computing Machinery. (Cited on pages 10 sq.).

Rohit Prabhavalkar, Takaaki Hori, Tara N. Sainath, Ralf Schlüter, and Shinji Watanabe. 2024. End-to-End Speech Recognition: A Survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32:325–351. (Cited on page 5).

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. *CoRR* abs/2306.17563. (Cited on page 12).

Francisco Manuel Rangel Pardo, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. 2015. Overview of the 3rd Author Profiling Task at PAN 2015 [in English]. In *CLEF 2015 Evaluation Labs and Workshop Working Notes Papers,* 1–8. CEUR Workshop Proceedings 1391. P3 Proceeding. Faculty of Arts, Linguistics; Research group: Centre for Computational Linguistics, Psycholinguistics and Sociolinguistics (CLiPS). Handle: https://hdl.handle.net/10067/1299240151162165141. Record Identifier: c:irua:129924. Toulouse. (Cited on page 6).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3982–3992. Hong Kong, China: Association for Computational Linguistics, November. (Cited on pages 7, 17, 32).

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. arXiv: 2304.09542 [cs.CL]. (Cited on pages 2, 13, 20 sq.).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems,* edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30. Curran Associates, Inc. (Cited on page 7).

Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an Argument Search Engine for the Web. In *Proceedings of the 4th Workshop on Argument Mining,* edited by Ivan Habernal, Iryna Gurevych, Kevin Ashley, Claire Cardie, Nancy Green, Diane Litman, Georgios Petasis, Chris Reed, Noam Slonim, and Vern Walker, 49–59. Copenhagen, Denmark: Association for Computational Linguistics, September. (Cited on pages 1, 6).

Hongzhi Zhang and M. Omair Shafiq. 2024. Survey of transformers and towards ensemble learning using transformers for natural language processing. *Journal of Big Data* 11 (1): 25. (Cited on page 5).

Leixin Zhang and Daniel Braun. 2024. Twente-BMS-NLP at PerspectiveArg 2024: Combining Bi-Encoder and Cross-Encoder for Argument Retrieval. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024),* edited by Yamen Ajjour, Roy Bar-Haim, Roxanne El Baff, Zhexiong Liu, and Gabriella Skitalinskaya, 164–168. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on page 13).

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large Language Models for Information Retrieval: A Survey. arXiv: 2308.07107 [cs.CL]. (Cited on page 20).