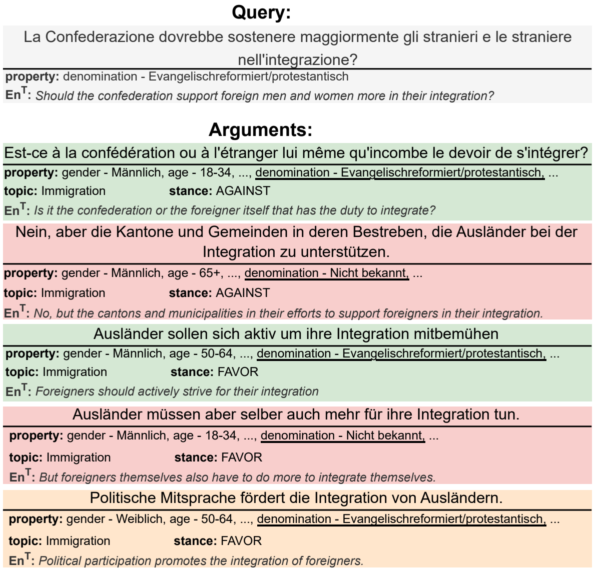
Sövereign at PerpectiveArg2024: Using LLMs with Argument Mining

Robert Günzler, Özge Sevgili[*], Steffen Remus, Chris Biemann, Irina Nikishina[*]

Universität Hamburg
Germany

Abstract

This paper presents the *Sövereign* submission for the shared task on perspective argument retrieval for the Argument Mining Workshop 2024 (Falk et al., 2024). To address the challenge, we apply open-access Large Language Models (Mistral-8x7b) in a zero-shot fashion for re-ranking and explicit similarity scoring. Additionally, we combine different features in an ensemble setup using logistic regression. Our system ranks second in the competition for all test set rounds on average for the logistic regression approach using LLM similarity scores as a feature. We also make the code publicly available[1].



Query:

🟩 La Confederazione dovrebbe sostenere maggiormente gli stranieri e le straniere nell'integrazione?

property: denomination - Evangelischreformiert/protestantisch

En^T: *Should the confederation support foreign men and women more in their integration?*

Arguments:

🟩 Est-ce à la confédération ou à l'étranger lui même qu'incombe le devoir de s'intégrer?

property: gender - Männlich, age - 18-34, ..., denomination - Evangelischreformiert/protestantisch, ...

topic: Immigration **stance:** AGAINST

En^T: *Is it the confederation or the foreigner itself that has the duty to integrate?*

🟥 Nein, aber die Kantone und Gemeinden in deren Bestreben, die Ausländer bei der Integration zu unterstützen.

property: gender - Männlich, age - 65+, ..., denomination - Nicht bekannt, ...

topic: Immigration **stance:** AGAINST

En^T: *No, but the cantons and municipalities in their efforts to support foreigners in their integration.*

🟩 Ausländer sollen sich aktiv um ihre Integration mitbemühen

property: gender - Männlich, age - 50-64, ..., denomination - Evangelischreformiert/protestantisch, ...

topic: Immigration **stance:** FAVOR

En^T: *Foreigners should actively strive for their integration*

🟥 Ausländer müssen aber selber auch mehr für ihre Integration tun.

property: gender - Männlich, age - 18-34, ..., denomination - Nicht bekannt, ...

topic: Immigration **stance:** FAVOR

En^T: *But foreigners themselves also have to do more to integrate themselves.*

🟧 Politische Mitsprache fördert die Integration von Ausländern.

property: gender - Weiblich, age - 50-64, ..., denomination - Evangelischreformiert/protestantisch, ...

topic: Immigration **stance:** FAVOR

En^T: *Political participation promotes the integration of foreigners.*

Figure 1: Cross-lingual perspective argument mining: relevant arguments are marked in green, and irrelevant arguments are marked in red or orange. In the orange one, demographic properties match, yet the texts are not relevant. The English translations (using Google Translate[2]) for the query and the arguments are denoted as En^T.

1 Introduction

Although the World Wide Web is full of content, search engines nowadays still lack support for extracting information regarding argument retrieval (Bondarenko et al., 2022). Argument retrieval addresses the issue of retrieving relevant arguments from a corpus based on a specific query (Falk et al., 2024). Further issues arise for particular perspectives, i.e., an argument might only be relevant in a special situation with certain restrictions. The shared task on "Perspective Argument Retrieval" (Falk et al., 2024), investigates these challenges by including sociocultural properties/factors (e.g. political interests, occupation, age, and gender) in a multilingual setup (see Figure 1 for illustration). The data includes documents in German, French, and Italian.

Motivated by the abilities of large language models (LLMs; cf. Zhao et al., 2023), we investigate methods to leverage them for this task. We consider two approaches: implicitly re-ranking the argument candidates, and explicitly computing relevance scores for candidate arguments. First, we use the cosine similarity of Sentence BERT (Reimers and Gurevych (2019) between the encoded arguments and the query to retrieve the nearest neighbors (arguments) as candidates similar to the baseline approach by Falk et al. (2024). Note that the given arguments also contain topic labels and socio-cultural factors (e.g. in Scenario 2, Section 3), which we also benefit from. We then supply the query and the retrieved candidates to an LLM and ask it to re-rank the arguments. In our second method, the LLM is asked to produce a score for a given query-argument pair. We further train a logistic regression classifier using several initial similarity scores as features and use the computed feature weights in an ensemble fashion to compute a final relevance score.

[*] Equal contribution.

[1] https://github.com/uhh-lt/sovereign-perspectiveArg24

[2] https://translate.google.com

Proceedings of the 11th Workshop on Argument Mining, pages 150–158
August 15, 2024 ©2024 Association for Computational Linguistics

Our LLM scoring based method shows improved performance for Scenario 1 and 2, while the LLM re-ranking performs competitively in Scenario 3, on the development set. Therefore, we submit the results obtained with the logistic regression using the LLM scoring as the final solution to the competition. The name of our team in the leaderboard of the organizers is ***"Sövereign"***.

The contributions of this paper are as follows:

- We investigate the ability of LLMs in argument mining with socio-cultural factors, experimenting with two approaches in a zero-shot setup: ranking by LLM directly and predicting relevance scores using LLM.

- We present a runner-up model, ranked as the second-best system, in the shared task in 2024.

2 Related Work

In this section, we briefly describe the existing studies that we take into consideration while developing our proposed approach.

Argument Retrieval

Apart from the current Perspective Argument Mining shared task (Falk et al., 2024), there exists a series of scientific events and shared tasks on computational argumentation and causality which named Touché (Bondarenko et al., 2022, 2023). Traditionally, the shared task is related to the specific topics, e.g., Retrieval for Comparatives / Controversy (Bondarenko et al., 2022, 2023), Image Retrieval (Bondarenko et al., 2022, 2023), etc. For a detailed overview of the Argument Mining field, we refer the reader to the papers by Lawrence and Reed (2019) and Bondarenko et al. (2023).

LLMs for Ranking

According to Qin et al. (2023), LLMs in zero-shot ranking tasks can be categorized into pointwise, listwise, pairwise, and setwise. Our approach applies the listwise method by Sun et al. (2023). The authors propose RankGPT, a generative LLM (here ChatGPT and GPT-4) for passage relevance ranking in information retrieval (IR) settings. Despite the fear of data contamination, they eventually concluded, that properly instructed LLMs can deliver competitive performance compared to supervised IR methods and can rank unknown knowledge.

3 Task Description

In this shared task, the goal is to retrieve multilingual arguments gathered from the voting recommendation platform[3]. For a description of the dataset, we refer to Falk et al. (2024). The key challenge here is to consider socio-cultural factors during retrieval. For the shared task 2024 we submit systems for all the three competition scenarios:

- **Scenario 1:** Default retrieval ranks argument candidates from a given corpus for a specific query ignoring any social-cultural attributes.

- **Scenario 2:** Explicit perspectivism adds socio-cultural information to the query and the arguments, which limits relevant arguments that match the corresponding socio-cultural factors.

- **Scenario 3:** Implicit perspectivism adds socio-cultural information only to the query, while it is not provided for the arguments.

For each evaluation round, the data consists of a set of queries and a set of candidates/arguments. The set of queries includes the query text, and for Scenarios 2 and 3 it also contains an explicitly given socio-cultural/demographic attribute. The set of candidates contains the argument text, a "stance" parameter ("favor" or "against"), and a "topic" parameter. The retrieval performance is measured using: *a*) relevance: **NDCG@k** and **Pr@k** (precision @ k), and *b*) diversity: α**NDCG@k** and **klDiv@k** (Kullback-Leibler Divergence @ k), where k is the rank of retrieved arguments.

4 Methodology

In this section, we present two approaches for the argument mining task. The first method applies the LLM directly to rank the arguments, the second integrates LLM scores as a feature for a logistic regression model. In the next subsection, we introduce scores utilized in both approaches.

4.1 Feature Scores

In both approaches, the LLM re-ranking and the logistic regression re-ranking, we employ three different scores. We describe each score in details below. In the LLM re-ranking approach presented in Section 4.2 below, scores are summed up and

[3] https://www.smartvote.ch/

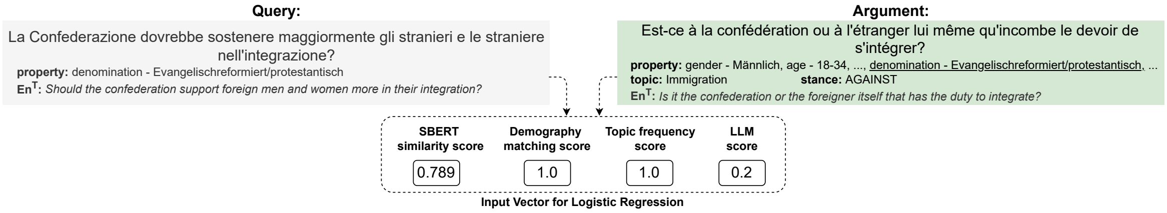

Query:
La Confederazione dovrebbe sostenere maggiormente gli stranieri e le straniere nell'integrazione?
property: denomination - Evangelischreformiert/protestantisch
En^T: Should the confederation support foreign men and women more in their integration?
Argument:
Est-ce à la confédération ou à l'étranger lui même qu'incombe le devoir de s'intégrer?
property: gender - Männlich, age - 18-34, ..., denomination - Evangelischreformiert/protestantisch, ...
topic: Immigration
stance: AGAINST
En^T: Is it the confederation or the foreigner itself that has the duty to integrate?
SBERT similarity score
0.789
Demography matching score
1.0
Topic frequency score
1.0
LLM score
0.2
Input Vector for Logistic Regression

Figure 2: Inputs to the logistic regression for an example query-argument pair with scores of SBERT similarity, demography matching (1.0 in this example as they are matching), topic frequency (1.0, since for this example all 50 arguments have the same topic), and the score predicted by LLM.

the arguments are re-ranked accordingly before being sent to the LLM for re-ranking. We use each score as a feature to train a logistic regression classifier which then produces the final relevance score (Section 4.3). We demonstrate an example input for logistic regression in Figure 2 for better understanding of each score.

SBERT Cosine Similarity Score We encode queries and arguments using SBERT and utilize the cosine similarity of their embeddings to rank arguments given a query. This strategy replicates the baseline approach by the organizers.

Demography Matching Score For Scenario 2, the explicitly given socio-cultural attributes allow hard filtering of the arguments that do not match the socio-cultural attributes of the query. We assign a score of 1 to all arguments that match the given socio-cultural attribute parameter and a score of 0 to all other arguments.

Topic Frequency Score We predict the relevance of each topic to a query as the frequency of that topic among the 50 highest-ranked arguments based on SBERT cosine similarity. For each query-argument pair we use only the relevance of the given topic to the query.

4.2 LLM Ranking

As the first approach, we prompt the LLM with the current query and a list of the 50 highest-ranked arguments based on our SBERT scores. For Scenarios 1 and 3, these scores are the sum of the similarities and the topic scores, and for Scenario 2 they include the scores based on socio-cultural attributes, as well. We then ask the model to return an ordered list of the arguments based on their relevance to the query. The template prompt that we use is presented in Appendix A in Example 1.

Despite producing the machine-readable lists, the LLM output barely includes all 50 argument IDs submitted as input. We interpret all missing IDs as irrelevant to the query and rank them lower at the end of the list.

For Scenario 3, the model re-ranks the arguments according to the socio-cultural attribute from the query and the implicit socio-cultural backgrounds for each argument in Appendix A in Example 2.

4.3 LLM as Score Predictor

For the second approach, we provide the LLM with a list of the 50 highest-ranked arguments (based on the summed feature scores of SBERT, cf. Section 4.1) and prompting it to assign a relevance score between 0 and 1 for each candidate. The prompt for Scenarios 1 and 2 is presented in Appendix A in Example 3.

The expected result is supposed to render a Python dictionary, where keys are sentence IDs, and values are the assigned relevance scores. This approach is also limited by omitting argument IDs in the LLM output. In such cases, we score the missed argument IDs as 0.

For Scenario 3, we all ask the model to predict the relevance score between the given socio-cultural attribute from the query and the implicit socio-cultural backgrounds for each argument with the prompt present in Appendix A in Example 4.

4.4 Ensemble Learning

The previously computed scores are aggregated as features for a logistic regression classifier. More specifically, the feature set is comprised of the LLM relevance score, SBERT cosine similarity between query and argument, the topic frequency score, and the demography matching score based on socio-cultural attributes, as shown in Figure 2. We train a logistic regression classifier for each scenario separately; the goal is to predict whether an argument is relevant to a query or not (label 0 or 1). To train the model, we use the top-100 (Scenario 2) or top-500 (Scenarios 1 and 3) highest-ranked candidate

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

arguments for each query from the training set.

The resulting weights from the logistic regression are presented in Table 2 in Appendix A. We interpret those scores as importance weights to rebalance the individual features of the candidate arguments. We additionally normalize them to sum up to 1. The weighted sum of the features is then used for re-ranking previously retrieved arguments.

5 Experimental Setup

We use the `Mixtral-8x7B-Instruct-v0.1`[4] LLM model by `mistralai`[5] with the default parameters using `HuggingChat`[6]. This model comes with a lenient license and offers a good balance between performance and model size[7]. By using the `HuggingChat` framework, we explicitly make the model exchangeable, and we expect increased performance by using larger models. We refrain from the model fine-tuning and apply it as a zero-shot.

Regarding SBERT, we use the pre-trained model `paraphrase-multilingual-mpnet-base-v2`[8], likewise the baseline from the organizers (Falk et al., 2024). We trained the logistic regression classifier using the `scikit-learn`[9] framework on the training dataset with the default parameters.

To choose the solution for the final evaluation round, we test our approaches on the development set and submit the test set ranking using the best-performing algorithm.

Scenario 1: Default Argument Retrieval Here, SBERT is already a very strong baseline. Logistic regression achieves better scores for **NDCG**, **Pr**, and α**NDCG** and $k > 4$ (cf. Table 4, in Appendix). Thus, for Scenario 1 we submit the results achieved with logistic regression.

Scenario 2: Explicit Perspectivism Results are shown in Appendix Table 3. LLM re-ranking performs well as compared to the SBERT baseline, however, the logistic regression ensemble achieves the best scores. For this scenario, we also submit the results achieved with logistic regression.

Scenario 3: Implicit Perspectivism In this scenario, both approaches perform almost on par; the LLM re-ranking methods perform better than other approaches, as shown in Table 5 in the Appendix. However, we still decided to submit the logistic regression approach, as we consider learned weights to be more fair for the unseen data.

6 Results on the Test Sets

In this section, we present the test results of our approach from logistic regression. These results are evaluated and shared by task organizers. Table 1 presents the average results for all scenarios and test rounds. Additionally, we show the average results of our approaches across different test rounds and scenarios in Tables 6, 7 and 8 in Appendix A. We achieve competitive results for all scenarios on test set 1 and test set 2, however, our predictions for test set 3 fall short of first place quite significantly. In Scenario 1 test 3 "Sövereign" underperforms even the SBERT baseline. We believe this happens because of the topic scores, included in the final logistic regression. If the SBERT baseline predicts relevant arguments that match the expected topic, this will improve the results by increasing the final scores for those arguments, that match the expected topic. Otherwise, this will impair the results by increasing the final scores for the arguments that do not match the topic. For test set 1 (Precision@20 $= 0.978$)

[4] `https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1`

[5] `https://mistral.ai/`

[6] We use HuggingChat version v0.8.4: `https://huggingface.co/chat/`

[7] Measured by personal experience.

[8] `https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2`

[9] `https://scikit-learn.org`

and test set 2 (Precision@20 $= 0.824$) the topic scores are predominantly valuable, hence our results are significantly higher than the baseline. On test set 3 (Precision@20 $= 0.565$) the topic scores are deceptive for a significant amount of queries: the usage of this feature puts the irrelevant arguments higher. The reason for the difference across different test sets needs to be further investigated.

7 Conclusion

We present *Sövereign*, an LLM re-ranking approach for perspective argument retrieval. We show an investigation of two LLM utilizations, *a*) implicit re-ranking, and *b*) explicit relevance scoring. The explicit relevance scoring methods achieve better scores for explicit perspectivism when used in an ensemble with other similarity features, i.e., SBERT, topic, and socio-cultural (if applicable). In Scenario 3, implicit perspectivism, LLM re-ranking performs better than the LLM scoring. We believe this might be due to the formulations of the prompts in Scenario 3: ranking prompt emphasizes socio-cultural property, directly. In future work, we would like to explore more utilization methods of LLMs in this task, e.g., trying different prompts. The data additionally contains “stance” attributes, which we omitted to use for our submission, but might be an important feature. We also plan to try other LLM models and improve the results for test set 3 by classifying the topic from the query and matching it with the topics from the arguments.

Limitations

Nowadays, dozens of large pre-trained generative models exist and we report results only on `mistralai/Mixtral-8x7B-Instruct-v0.1`. It might be that some other foundation models could further push the results, however, our main goal was to investigate the ability of LLMs to re-rank arguments given socio-cultural factors.

As we use HuggingChat API[10], it could produce every time different responses, which might slightly affect the results if reproducing the approach from scratch. However, we have saved the model output used for the final score submission, therefore, they can be used to reproduce the results.

[10] https://huggingface.co/chat/

References

Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Ferdinand Schlatt, Valentin Barriere, Brian Ravenet, Léo Hemamou, Simon Luck, Jan Heinrich Reimer, Benno Stein, Martin Potthast, and Matthias Hagen. 2023. Overview of Touché 2023: Argument and Causal Retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 14th International Conference of the CLEF Association (CLEF 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.

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*, pages 311–336, Cham. Springer International Publishing.

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*, Bangkok. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

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.

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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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. *Preprint*, arXiv:2304.09542.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Appendix

Here is the template prompt for the Scenario 1 or 2 performed with the LLM re-ranking:

(1) <<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]

Here is the template prompt for the Scenario 3 performed with the LLM re-ranking:

(2) <<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]

Here is the template prompt for Scenario 1 or 2 performed with similarity scores as a feature for Logistic Regression:

(3) <<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]

Here is the template prompt for Scenario 3 performed with similarity scores as a feature for Logistic Regression:

(4) <<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]

Scenario	SBERT similarity	Topic Frequency	Demographic Matching	LLM relevance
1	0.771	0.037	-	0.191
2	0.407	0.064	0.479	0.049
3	0.467	0.287	-	0.246

Table 2: Normalized Logistic Regression weights for the features calculated on the train set.

k	Method	Relevance		Diversity	
		NDCG@k	Pr@k	αNDCG@k	klDiv@k
4	SBERT baseline	0.180	0.182	0.167	**0.151**
	LLM reranking	0.772	0.732	0.724	0.205
	LogReg	**0.866**	**0.796**	**0.812**	0.206
8	SBERT baseline	0.181	0.181	0.169	**0.136**
	LLM reranking	0.752	0.666	0.719	0.192
	LogReg	**0.853**	**0.723**	**0.813**	0.193
16	SBERT baseline	0.180	0.178	0.172	**0.107**
	LLM reranking	0.740	0.590	0.718	0.165
	LogReg	**0.844**	**0.641**	**0.817**	0.167
20	SBERT baseline	0.180	0.176	0.172	**0.099**
	LLM reranking	0.735	0.563	0.716	0.157
	LogReg	**0.840**	**0.612**	**0.817**	0.160

Table 3: Results for Scenario 2 on the development set.

k	Method	Relevance		Diversity	
		NDCG@k	Pr@k	αNDCG@k	klDiv@k
4	SBERT baseline	**0.968**	**0.975**	**0.878**	**0.151**
	LLM reranking	0.962	0.967	0.865	0.164
	LogReg	0.967	**0.975**	0.873	0.162
8	SBERT baseline	0.965	0.967	0.880	**0.137**
	LLM reranking	0.973	0.979	0.881	0.151
	LogReg	**0.976**	**0.983**	**0.885**	0.149
16	SBERT baseline	0.957	0.954	0.892	**0.107**
	LLM reranking	0.966	0.967	0.896	0.124
	LogReg	**0.968**	**0.969**	**0.899**	0.121
20	SBERT baseline	0.954	0.950	0.897	**0.100**
	LLM reranking	0.963	0.962	0.901	0.116
	LogReg	**0.966**	**0.965**	**0.905**	0.114

Table 4: Results for Scenario 1 on the development set.

k	Method	Relevance		Diversity	
		NDCG@k	Pr@k	αNDCG@k	klDiv@k
4	SBERT baseline	0.187	0.188	0.172	**0.151**
	LLM reranking	**0.198**	**0.198**	**0.181**	0.157
	LogReg	0.193	0.194	0.177	0.156
8	SBERT baseline	0.191	0.193	0.177	**0.136**
	LLM reranking	**0.201**	**0.202**	**0.186**	0.144
	LogReg	0.198	0.200	0.184	0.142
16	SBERT baseline	0.198	0.199	0.186	**0.107**
	LLM reranking	**0.209**	**0.211**	**0.196**	0.118
	LogReg	0.204	0.206	0.192	0.114
20	SBERT baseline	0.200	0.201	0.189	**0.099**
	LLM reranking	**0.212**	**0.213**	**0.199**	0.111
	LogReg	0.207	0.207	0.195	0.106

Table 5: Results for Scenario 3 on the development set.

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 6: Average results for Scenario 1 on all test sets.

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 7: 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 8: Average results for Scenario 3 on all test sets.