Precision-Oriented Argument Retrieval

Bachelor’s Thesis Defence

Danik Hollatz

16.06.2022

Motivation

- Argumentation plays an enormous role when it comes to building an opinion on a controversial topic or making personal choices [Wachsmuth et al, 2017].
- Most approaches implemented in Touché use recall-oriented techniques.
- This motivates an idea to test precision-oriented approaches.

Touché—collaborative platform for researchers
https://webis.de/events/touche-22/

Approaches

In this thesis three different approaches were tested:

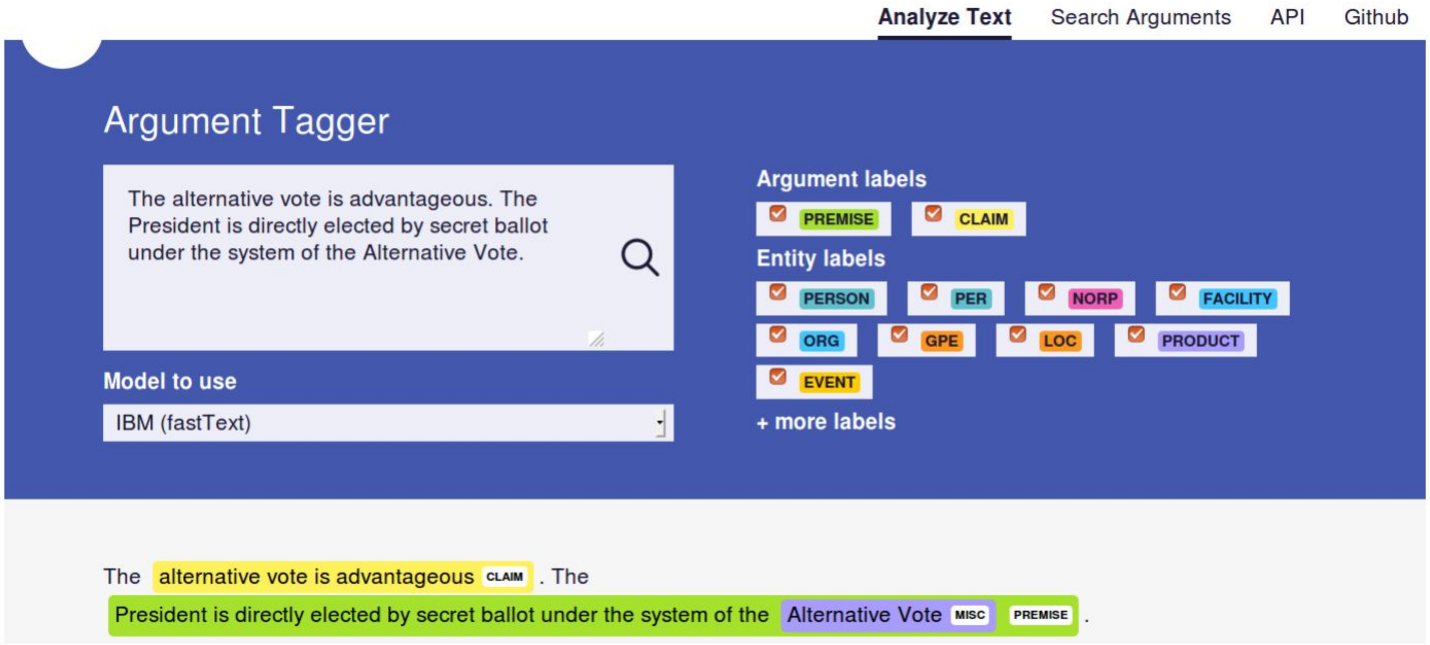
1. Find premises and claims in the argument and dismiss the rest of the argument.
2. Find for each token in the argument its semantic importance/relevance in the passage and consider it at retrieval. Pre-trained deep learning network DeepCT [Dai and Callan, 2020] is used for this approach.
3. Same as approach (2), but instead of pretrained network, the DeepCT model is first being fine-tuned.

Approach 1. TARGER.

“Find premises and claims in the argument and dismiss the rest of the argument.”

- In this approach, the models deployed in TARGER [Chernodub et al, 2019] are used to extract the claims and premises of an argument.
- TARGER is an open-source mining framework, based on neural networks, which is used for tagging arguments in texts and for argument retrieval. TARGER provides models pre-trained on 4 different argument mining datasets and 3 precomputed word embeddings.

Approach 1. Example of TARGER web-interface.



Argument Tagger

The alternative vote is advantageous. The President is directly elected by secret ballot under the system of the Alternative Vote.

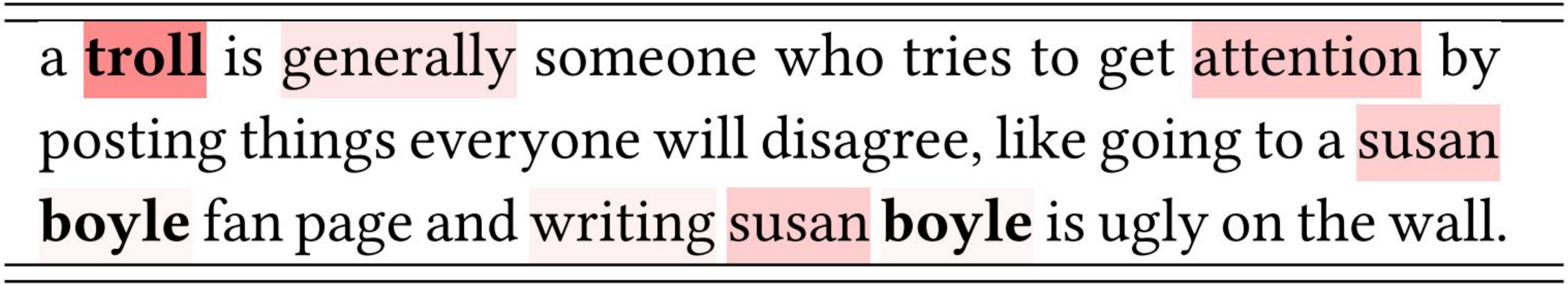
Model to use: IBM (fastText)

The alternative vote is advantageous CLAIM . The President is directly elected by secret ballot under the system of the Alternative Vote MISC PREMISE .

Using TARGER web-interface, an argument is split into different labels.
Source: https://aclanthology.org/P19-3031v2.pdf

Approach 2. DeepCT weighting example.

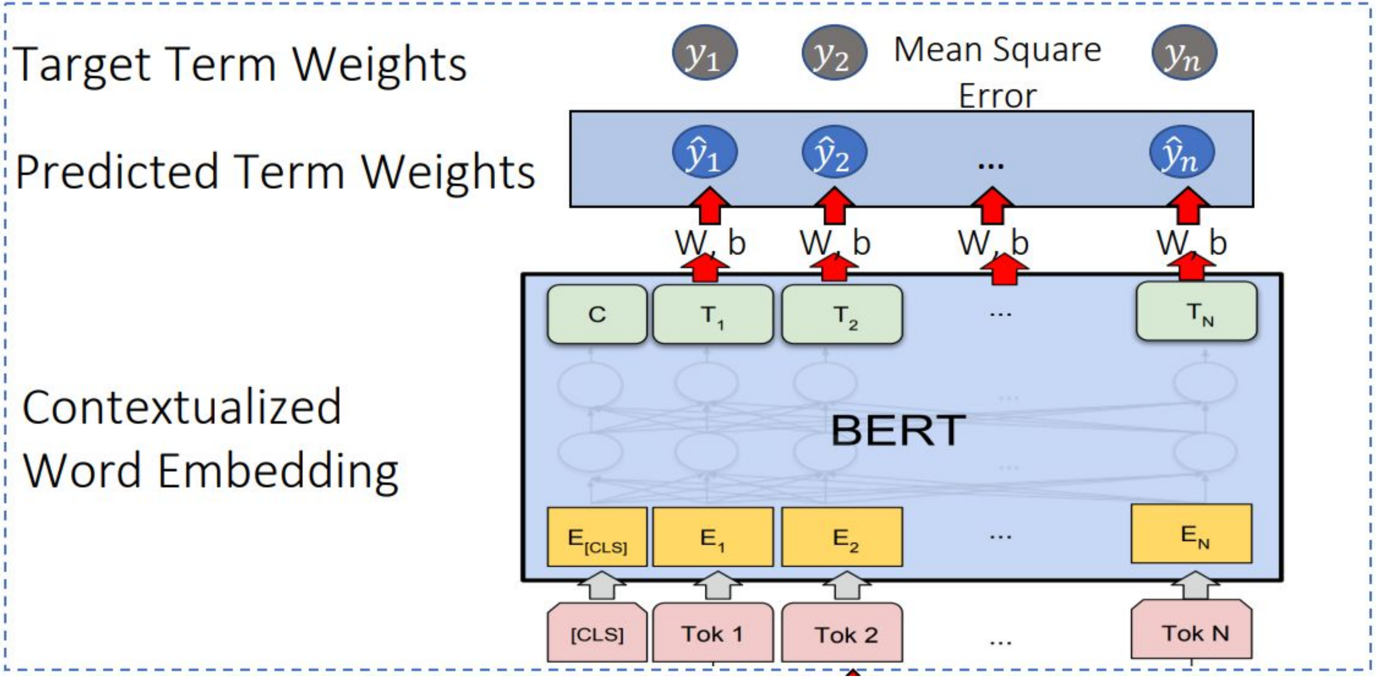
“Find for each token in the argument its semantic importance/relance in the passage and consider it at retrieval. Pre-trained deep learning network DeepCT is used for this approach.”



a **troll** is generally someone who tries to get attention by posting things everyone will disagree, like going to a susan **boyle** fan page and writing susan **boyle** is ugly on the wall.

Visualization of an weighted passage. Deeper color represents higher weights.
Source: https://dl.acm.org/doi/abs/10.1145/3366423.3380258

Approach 2. DeepCT Architecture.



Architecture of DeepCT model.
Source: https://dl.acm.org/doi/abs/10.1145/3366423.3380258

Approach 3. Overview

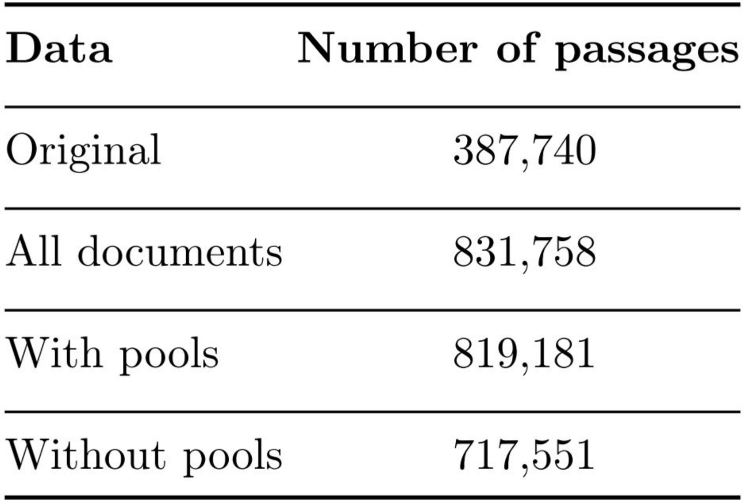
“Same as approach (2), but instead of pretrained network, the DeepCT model is first being fine-tuned.”

- To fine-tune DeepCT model an args.me [Ajjour et al., 2019] corpus was used.
- The args.me corpus contains 387,740 arguments crawled from four debate portals.
- Each data entry consists of the text of the debate, its conclusion, and the corresponding topic (later used as “Reference Field”).
- Nine different datasets were created to finetune and evaluate the performance of the DeepCT model.

Approach 3. Split of Corpus

The args.me corpus was first divided into three different datasets:

1. “All documents”: all documents stored, even ones that are judged.
2. “With pools”: all documents stored, except those that are judged.
3. “Without pools”: all documents stored, except those that are judged and except top-50 documents that were retrieved by participants of Touché 2021.



Data	Number of passages
Original	387,740
All documents	831,758
With pools	819,181
Without pools	717,551

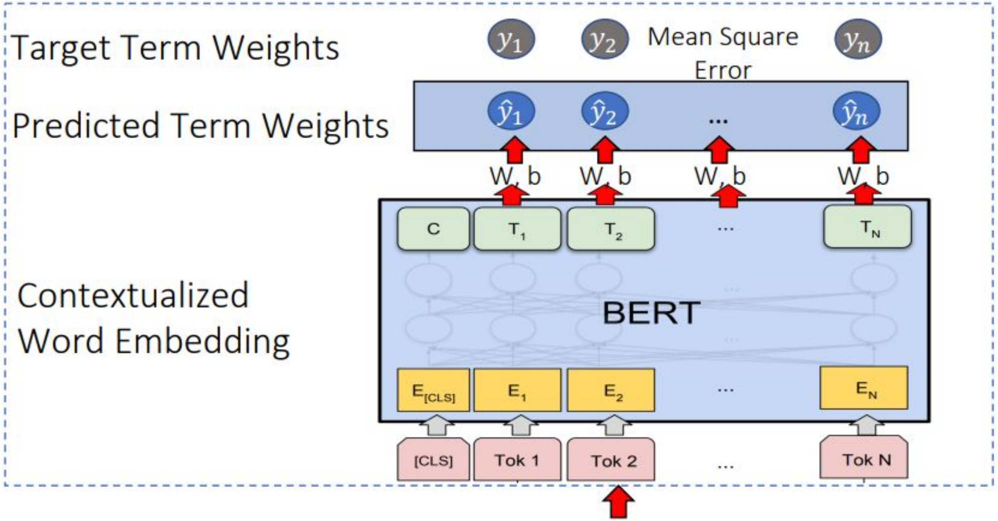
Because of split into smaller passages, the number of passages is increased (next slide).

Approach 3. Split of Corpus

The args.me corpus was first divided into three different datasets.

- For controversial topics some of the arguments from args.me corpus were annotated with relevance judgments.
- Judgments provided by Touché lab and were manually annotated.
- Since the text of arguments can be too long and the input to finetune DeepCT model is limited to 512 tokens, the text of arguments is first being divided into smaller passages of maximum size 500 tokens.

Approach 3. Content-based training strategy



Passage Content "Yellowstone experiences thousands of small earthquakes …"

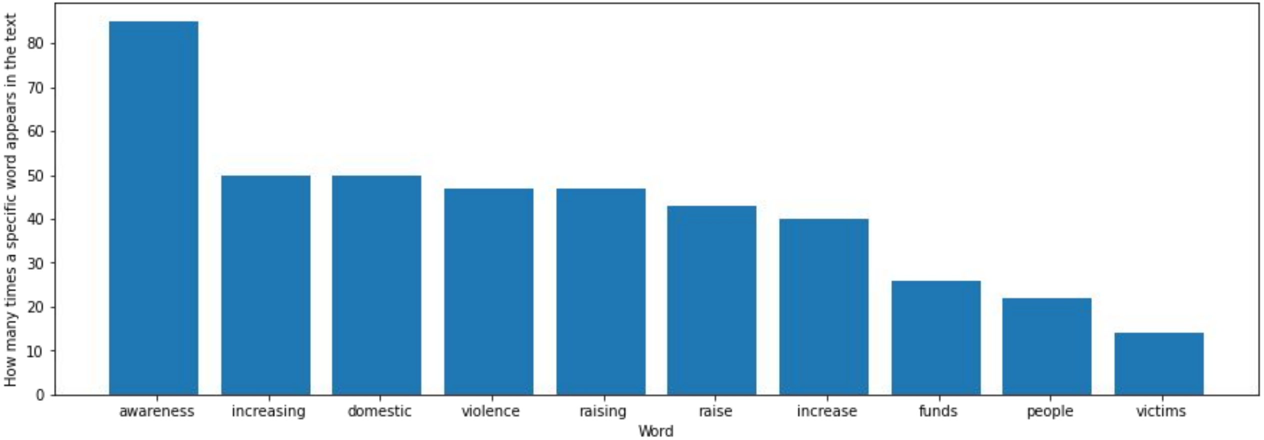
	Content	**Relevance**	**PRF**
Target Term Weights (Training Labels)	{"Yellowstone":1, "National": 1, "Park": 1}	{"Yellowstone":0.89, "Earthquake":0.12, "Wildlife": 0.2}	{"Park":0.75, "service":0.75, "nps": 0.25}

Approach 3. Training Example and Output of DeepCT.

Passage: “As I mentioned in my previous claim, raising awareness can make the perpetrator feel targeted and cause an increase in abuse. Rather than raise awareness for … and increasing awareness will not benefit victims because they will be risking their lives to receive help but not be able to receive it on time. The National Network to End Domestic Violence reported that ”more than 22,000 calls were answered by local domestic violence hotlines, and on that same day, more than 9,500 requests for services were unmet due to inadequate funding or staff available to assist these survivors” (2).”.

Reference Field (Conclusion): “Domestic Violence Awareness should be increased”

Ground truth: “violence”: 1.0, “awareness”: 1.0, “increase”: 1.0, “increasing”: 1.0, “domestic”: 1.0



Overview of most repeated words in passage from training’s sample after inference of DeepCT model.

Approach 3. Algorithm

To create training samples for the network the content-based strategy proposed by authors of DeepCT model is used.

For the given text input and reference field:

1. Lower and tokenize the text input and reference field.
2. Remove stopwords from text input and reference field.
3. Apply stemming to the remaining tokens.
4. Create a set of overlapping stemmed tokens from the text input and reference field. The original tokens of overlapped stems are then considered as important ones.

Evaluation

Performed on the (incomplete) relevance judgments and topics of Touché in years 2020, 2021 and both 2020 and 2021.

For the evaluation were chosen 3 metrics:

1. nDCG@5. (Can be directly compared with the results of participants approaches of Touché task 1)
2. nDCG@25.
3. Bpref.

Retrieval models (tuned), PyTerrier implementation:

1. BM25.
2. BM25 + RM3.
3. Dirichlet-smoothed Language model.
4. Dirichlet-smoothed Language model + RM3.

Evaluation. Tuning of retrieval models parameter.

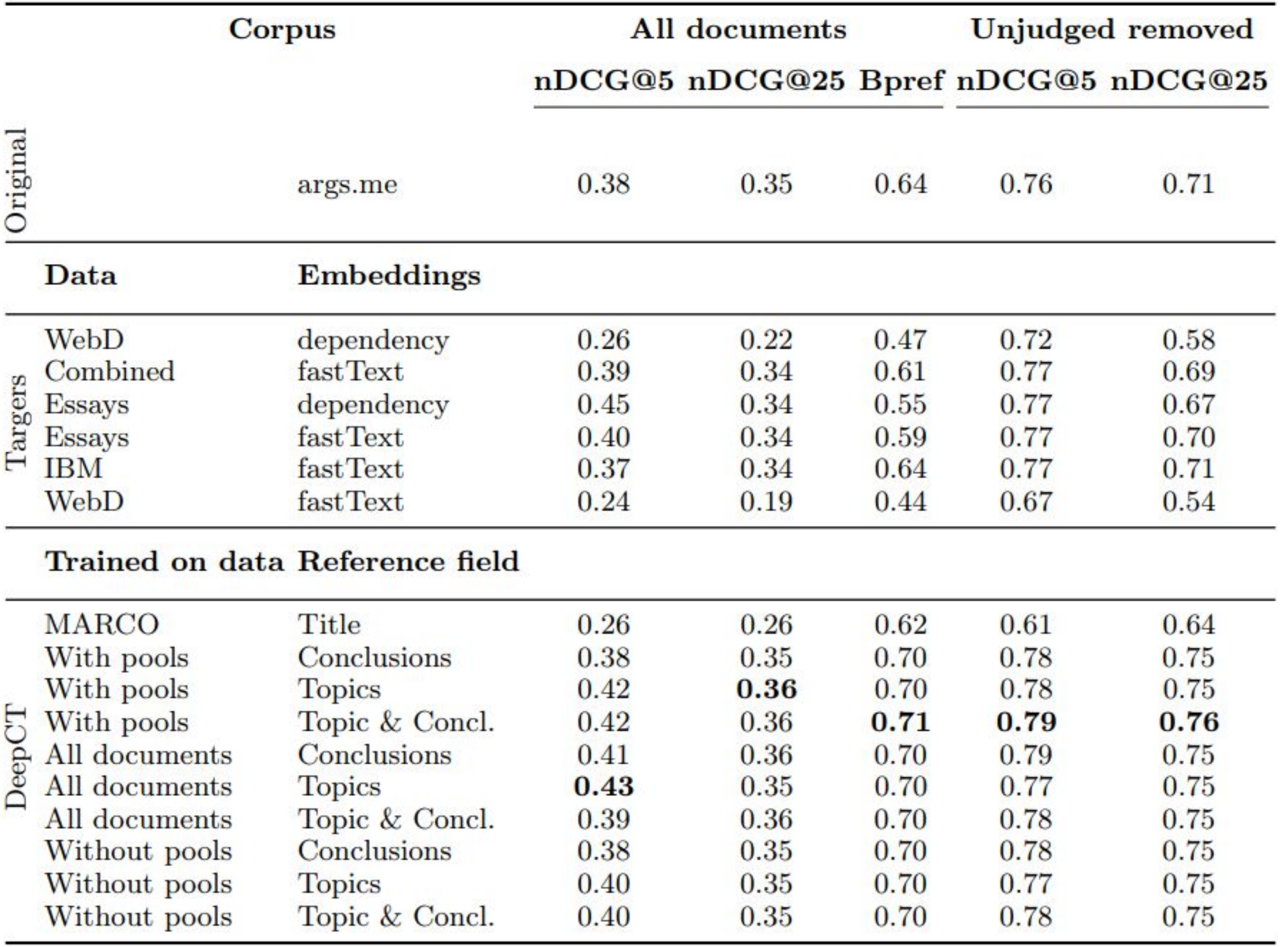
According to authors of DeepCT and as it was shown in practice, tuning of parameters is essential and increases the effectiveness of approaches. Best parameters were found via grid-search in the following ranges:

- BM25, b from [0.15, 0.75] with a step size of 0.2, k_1 from [0.6, 4.4] with a step size of 0.6 and k_2 for following values: [2,5,8,10].
- DLM, the smoothing parameter μ from [0,10000] with a step size of 250.
- RM3 variant of retrieval models, number of terms M to add to the query from [4, 16] with a step size of 2, number of feedback documents N from [4, 10] with a step size of 2 and the relevance of the original query λ from [0.2, 1] with a step size of 0.2.

For the query expansion method of RM3 only the parameters of RM3 are tuned while the default parameters of the used retrieval model are used.

Evaluation

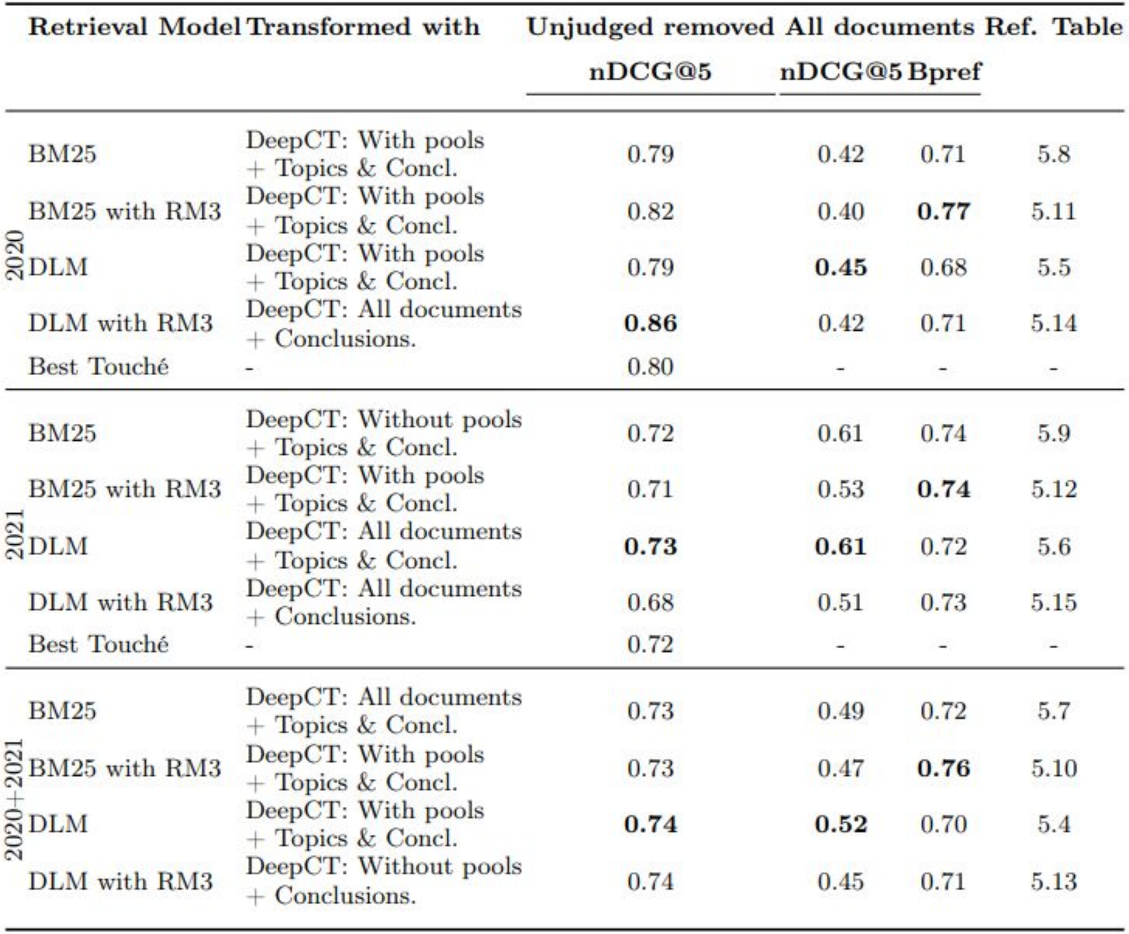
Effectiveness of the tuned BM25 approach on topics of the year 2020.



	Corpus		All documents			Unjudged removed	
			nDCG@5	nDCG@25	Bpref	nDCG@5	nDCG@25
Original		args.me	0.38	0.35	0.64	0.76	0.71
	Data	**Embeddings**					
Targers	WebD	dependency	0.26	0.22	0.47	0.72	0.58
	Combined	fastText	0.39	0.34	0.61	0.77	0.69
	Essays	dependency	0.45	0.34	0.55	0.77	0.67
	Essays	fastText	0.40	0.34	0.59	0.77	0.70
	IBM	fastText	0.37	0.34	0.64	0.77	0.71
	WebD	fastText	0.24	0.19	0.44	0.67	0.54
	Trained on data	**Reference field**					
DeepCT	MARCO	Title	0.26	0.26	0.62	0.61	0.64
	With pools	Conclusions	0.38	0.35	0.70	0.78	0.75
	With pools	Topics	0.42	**0.36**	0.70	0.78	0.75
	With pools	Topic & Concl.	0.42	0.36	**0.71**	**0.79**	**0.76**
	All documents	Conclusions	0.41	0.36	0.70	0.79	0.75
	All documents	Topics	**0.43**	0.35	0.70	0.77	0.75
	All documents	Topic & Concl.	0.39	0.36	0.70	0.78	0.75
	Without pools	Conclusions	0.38	0.35	0.70	0.78	0.75
	Without pools	Topics	0.40	0.35	0.70	0.77	0.75
	Without pools	Topic & Concl.	0.40	0.35	0.70	0.78	0.75

Evaluation

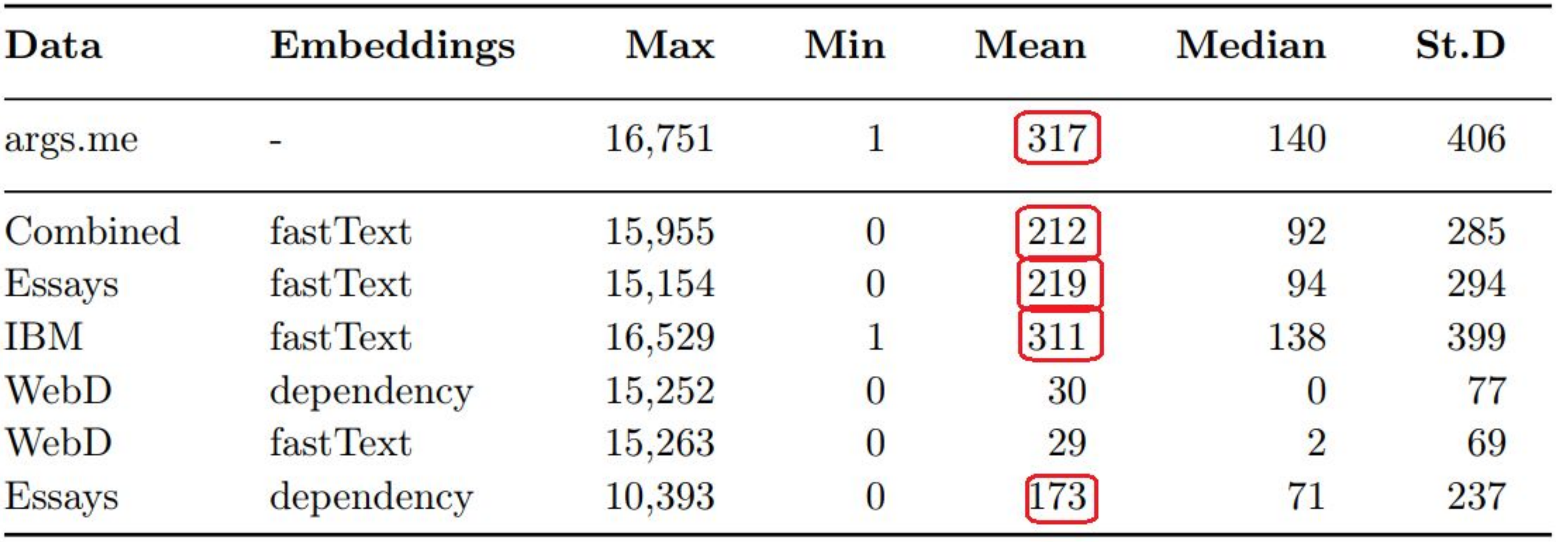
Overview of approaches that achieved most effective results in nDCG@5-score for each retrieval model and for each year.



	Retrieval Model	Transformed with	Unjudged removed	All documents		Ref. Table
			nDCG@5	nDCG@5	Bpref	
2020	BM25	DeepCT: With pools + Topics & Concl.	0.79	0.42	0.71	5.8
	BM25 with RM3	DeepCT: With pools + Topics & Concl.	0.82	0.40	**0.77**	5.11
	DLM	DeepCT: With pools + Topics & Concl.	0.79	**0.45**	0.68	5.5
	DLM with RM3	DeepCT: All documents + Conclusions.	**0.86**	0.42	0.71	5.14
	Best Touché	-	0.80	-	-	-
2021	BM25	DeepCT: Without pools + Topics & Concl.	0.72	0.61	0.74	5.9
	BM25 with RM3	DeepCT: With pools + Topics & Concl.	0.71	0.53	**0.74**	5.12
	DLM	DeepCT: All documents + Topics & Concl.	**0.73**	**0.61**	0.72	5.6
	DLM with RM3	DeepCT: All documents + Conclusions.	0.68	0.51	0.73	5.15
	Best Touché	-	0.72	-	-	-
2020+2021	BM25	DeepCT: All documents + Topics & Concl.	0.73	0.49	0.72	5.7
	BM25 with RM3	DeepCT: With pools + Topics & Concl.	0.73	0.47	**0.76**	5.10
	DLM	DeepCT: With pools + Topics & Concl.	**0.74**	**0.52**	0.70	5.4
	DLM with RM3	DeepCT: Without pools + Conclusions.	0.74	0.45	0.71	5.13

Evaluation

Overview of statistics of original args.me corpus and after extracting premises and claims via models deployed in TARGER. Values are given in the number of tokens.



Data	Embeddings	Max	Min	Mean	Median	St.D
args.me	-	16,751	1	317	140	406
Combined	fastText	15,955	0	212	92	285
Essays	fastText	15,154	0	219	94	294
IBM	fastText	16,529	1	311	138	399
WebD	dependency	15,252	0	30	0	77
WebD	fastText	15,263	0	29	2	69
Essays	dependency	10,393	0	173	71	237

Conclusions

- Fine-tuned DeepCT-based models are able to outperform most effective participant approaches in Touché track 1 in years 2020, 2021.
- Fine-tuned DeepCT-based models perform more effective as a pre-trained-based one.
- TARGER-based models can be used instead to reduce to required space to store the corpus, by saving only premises and claims.

Conclusions

- Fine-tuned DeepCT-based models are able to outperform most effective participant approaches in Touché track 1 in years 2020, 2021.
- Fine-tuned DeepCT-based models perform more effective as a pre-trained-based one.
- TARGER-based models can be used instead to reduce to required space to store the corpus, by saving only premises and claims.

Thank you!