Neville Longbottom at Touché 2023: Image Retrieval for Arguments using ChatGPT, CLIP and IBM Debater

Notebook for the Touché Lab on Argument and Causal Retrieval at CLEF 2023

Daria Elagina[1], Bernd-Albrecht Heizmann[1], Max Koch[1], Gustav Lahmann[1] and Christian Ortlepp[1]

[1]*Friedrich-Schiller University Jena, 07743, Jena*

Abstract

Argumentative search engines could benefit from images visualizing each side of the presented pro and con arguments. The objective of this task is to retrieve 10 images supporting and 10 images opposing a controversial topic from a given dataset of images. We describe a way to prompt ChatGPT for arguments backing each stance, which will then be used to find matching images. Our first approach uses BM25 on the website text to represent an image, while our second method uses OpenAI's CLIP to analyze the images themselves and match them with arguments. Both retrieval methods can then be refined using IBM Debater's stance detection capabilities to further increase stance awareness.

Keywords

CLIP, ChatGPT, IBM Debater, boilerpy

1. Introduction

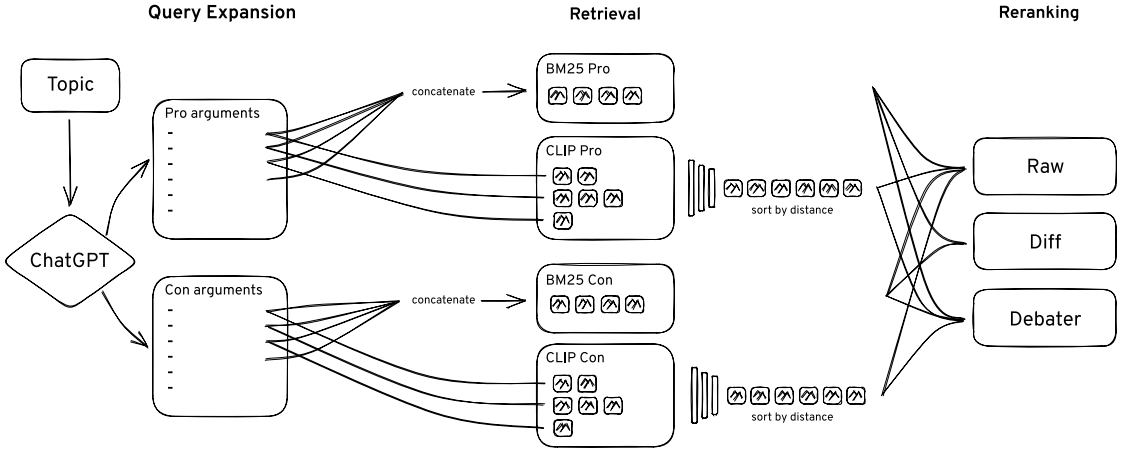
In order to find images conveying either a supporting (PRO) or opposing (CON) stance towards a controversial topic, we considered it being helpful to know what talking points each position commonly uses. Since this requires some outside knowledge, we wanted to make use of recent advances in generative large language models by using the knowledge embedded inside them, originating from their training data. As ChatGPT was trained on large portions of the Internet until 2021 and none of the 50 topics presented in this task are about a phenomenon only emerging after that time, we used it to generate a list of arguments supporting or opposing each topic.

Using these two sets of arguments, we built one pipeline for retrieving PRO images, and one for retrieving CON images for a topic. Our first approach uses the concatenation of all arguments of a stance for a topic to search in a BM25 index of the page contents in the dataset. The resulting images are ranked based on the BM25 score of the document they are found within. Our second approach uses OpenAI's pre-trained CLIP model[1] to retrieve images based on their actual content. The model was trained to map image data as well as the corresponding image description (short phrases of words) into the same 300-dimensional vector space. Using the

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

daria.elagina@uni-jena.de (D. Elagina); bernd-albrecht.heizmann@uni-jena.de (B. Heizmann); m.koch@uni-jena.de (M. Koch); gustav.lahmann@uni-jena.de (G. Lahmann); christian.ortlepp@uni-jena.de (C. Ortlepp)

CEUR Workshop Proceedings (CEUR-WS.org)



separate arguments as the description phrase we get a point in this vector space. By returning the images closest to this point we retrieve images whose content matches the argument. After doing this for every argument separately, the distance of each image to their respective argument point is used as the score to rank by.

While the retrieved images are already stance aware due to the stance specific arguments, we tried to further eliminate neutral, on-topic images by using IBM Debater to rerank the results of the BM25 or CLIP retrieval stage. Five surrounding sentences for each image are passed to the API alongside the topic, returning a score indicating whether they both convey the same stance.

2. Query expansion

To find arguments of a particular stance towards a given topic, we prompted ChatGPT using the following template, in which `$STANCE` is replaced by “for” for PRO and “against” for CON. The original query is passed as `$TOPIC`.

> Name a list of arguments `$STANCE "$TOPIC"`

The answers were always correctly formatted markdown lists (either enumerated or with bullet points), which could be easily parsed into a list of separate arguments. Since our project began with the start of the public beta of ChatGPT, no official API was available yet, so the answers were fetched from the browser interface of the ChatGPT version deployed at that time using an unofficial puppeteer script. For reproducibility, all answers for both stances and topics 51 to 100 are hard coded in a JSON file published with our source code. [1]

3. Retrieval

After expanding the topic for a given stance into a list of arguments, we try to find corresponding images from the dataset in two different ways. The first one uses the surrounding text of the image, whereas the second approach exclusively works on the image data itself.

[1] https://github.com/corite/ir-code

Name a list of arguments for "Should the US Electoral college be abolished?"

- The Electoral College is undemocratic and gives disproportionate weight to certain states.
- ...

For each of the arguments, what could be an image description of an image illustrating the issue?

- An image of a map of the United States with certain states highlighted to show how the Electoral College disproportionately affects the outcome of elections in those states.
- ...

Figure 1: Query expansion with PRO arguments for topic 98: ChatGPT conveniently answers using markdown formatted lists. The generated image description wasn't used in the end, as the argument phrases as input for both BM25 and CLIP yielded better results on our testing data.

3.1. BM25

In this approach, we assume the image's stance towards a topic is reflected by the text surrounding it. Following the idea that the text in closer spatial proximity to the image in the document is more likely to be connected to the image's content, we implemented an algorithm to extract paragraphs from the document starting at the image and then alternating between paragraphs directly above and below the image. These paragraphs will be concatenated and used as the document representation in the BM25 index.

To find nodes containing meaningful text in the HTML document, we used the `DefaultExtractor`[2] of the boilerpy3 library, because the default `ArticleExtractor` left out too many relevant parts of the document. The extractor employs some heuristics and marks nodes in the HTML document, which are likely to contain meaningful text, using a special tag. We then used the xPaths of the images provided in the image dataset to find image occurrences in the tagged HTML document. Starting with the alt text of the image, if available, text from below and above the image is extracted alternatingly.

Using the first 4096 characters of the concatenated paragraphs obtained this way, a BM25 index was built for the whole dataset using the default implementation of pyterrier. The document IDs are the ID of the image, whose containing HTML page was used for content extraction. Given a list of arguments for a certain stance, we concatenate all the arguments and use the resulting string as the input query for the BM25 search, whose results will be further processed by a reranker as described later.

3.2. CLIP

The CLIP [1] neural network was designed and trained by OpenAI to map images as well as short phrases of text describing images into the same 512-dimensional vector space. While projects like Stable Diffusion use this model to obtain vector representations for prompts, we

[2] https://github.com/jmriebold/BoilerPy3/blob/master/boilerpy3/extractors.py

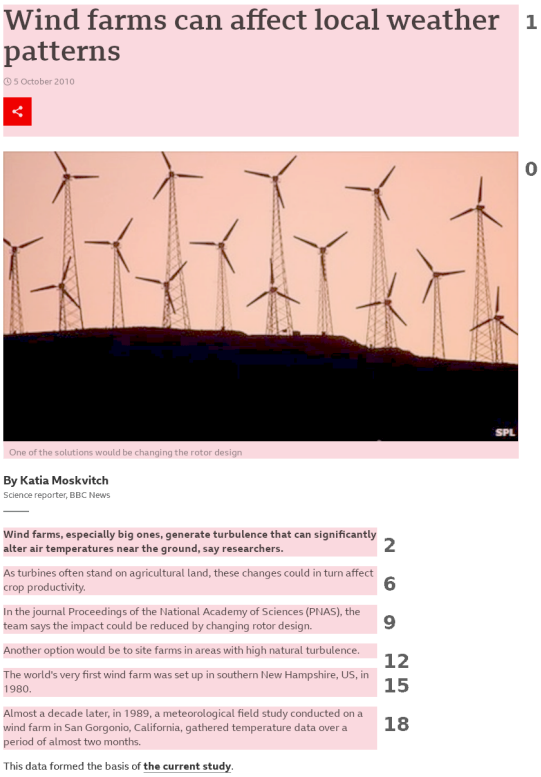


Figure 2: Nodes marked by boilerpy3's `Default Extractor` are highlighted in pink. The numbers indicate the order in which the content of the nodes are concatenated to form the document representation. Missing numbers correspond to nodes not found in the DOM when creating this visualization.

wanted to find images representing a certain topic, or even argument.

For every image in the dataset, we used the pre-trained `openai/clip-vit-base-patch32`[3] model to infer the corresponding vector representation, which was then normalized to length one and added to a `BallTree` index from sklearn. This enables us to use a k-nearest-neighbor search, which results in the same ranking as when sorting by cosine similarity.

Based on a list of separate arguments, we looked up the 100 nearest images to the vector representing the argument text for each argument. The retrieval result is sorted by the distance to the respective argument, so images obtained for different argument phrases will share the top ranks. If an image appears in the 100 nearest neighbors of multiple arguments, the lowest of the distances is used as the score.

[3] https://huggingface.co/openai/clip-vit-base-patch32

4. Reranking

While the retrieval operations discussed before can be efficiently applied on the whole dataset and result in on-topic images most of the time, we attempted to further refine the results towards a certain stance. For this, we used the top 500 images obtained using either CLIP or BM25 and passed them to our reranking stage.

The parameters used in both rerankers were chosen using a grid search on the 2022 dataset on topics 12, 13, 20, 46 and 49, for which we manually judged the union of the top 1000 CLIP and BM25 results using the original topic as the input.

4.1. Diff

Often on-topic images appeared in the top results of both stances. Therefore, in this reranking we subtracted 0.5 times the BM25 score of the same document for the opposite stance from the actual BM25 score, in an attempt to lower the score of neutral but on-topic images.

4.2. Debater

We used IBM's Project Debater Early Access API [2] to calculate a new score for each image based on the first 5 sentences of our document representation, i.e. the sentences closest to the image on the page, including the image's alt text.

For this, we used the "pro-con" endpoint, which given two sentences returns the stance of the first sentence relative to the second sentence as a number between -1 (CON) and 1 (PRO). We calculated this score for each sentence paired with the original topic text and averaged them to get the new image score. When ranking for CON instead of PRO, the score was simply inverted.

5. Evaluation

We submitted a single Docker image to TIRA [3], which can run any of the pipeline combinations when passing the corresponding name as an environment variable. The arguments generated by ChatGPT as well as the scores from the IBM Debater api were fetched in advance and are mounted into the container to allow for reproducible runs without Internet access.

Finding on-stance images is the hardest part of this task. When looking at the `on stance` score, the `raw` pipelines without the reranking stage outperforming both the `diff` and `debater` rerankers for BM25 as well as CLIP retrieval, show they are less reliable than our retrieval stage on its own.

run	on topic	argumentative	on stance
clip_chatgpt_args.raw	0.785	0.338	0.222
clip_chatgpt_args.debater	0.684	0.341	0.216
bm25_chatgpt_args.raw	0.572	0.274	0.166
bm25_chatgpt_args.diff	0.442	0.240	0.150
bm25_chatgpt_args.debater	0.416	0.201	0.128

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, 2021. arXiv:2103.00020.

[2] R. Bar-Haim, Y. Kantor, E. Venezian, Y. Katz, N. Slonim, Project debater apis: Decomposing the ai grand challenge, 2021. arXiv:2110.01029.

[3] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. URL: https://doi.org/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.