Boromir at Touché 2022: Combining Natural Language Processing and Machine Learning Techniques for Image Retrieval for Arguments

Notebook for the Touché Lab on Argument Retrieval at CLEF 2022

Thilo Brummerloh, Miriam Louise Carnot, Shirin Lange and Gregor Pfänder

Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany

Abstract

With the frequent information overload when scrolling the web, little information sticks with the reader. In argumentation, images are often used to leave a formative impression. Until now, there has been little research focusing on search engines specifically devoted to finding argumentative images. Argumentative images help the viewer to form an opinion by implicitly or explicitly giving an argument that either supports or invalidates a thesis. We built a search engine that assists users to overview a controversial topic with supporting and opposing images. With this goal in mind, we compare different techniques from the fields of Natural Language Processing and Machine Learning to cluster the images, extract the text from the images and evaluate the sentiment of the page the image appears on. The best retrieval system uses a BERT model to determine stance, query Preprocessing, optical character recognition, and image clustering to detect the image content. Over 50% of the images found by this retrieval system are relevant, argumentative, and assigned to the correct stance according to automatic and manual evaluation.

Keywords

argument retrieval, images, information retrieval, search engines

1. Introduction

The internet contains a vast assortment of opinions and arguments in social media posts, discussion forums, and news pages that are presented, challenged, and evaluated by contributors and readers. But mostly these textual argumentation are not structured as arguments [1]. Those arguments are commonly expressed verbally. However, it is possible to present some of them as images with the argument written on them or simply through visual communication e.g., symbolism [2]. A common example of visual arguments are memes which amongst others became popular as a method to influence the 2016 presidential primaries in the U.S. [3].

Our research addresses the above-stated problems with the implementation of an image-based argument search engine in the course of Shared Task 3 from Touché 2022. The topic of the task is "Image retrieval to corroborate and strengthen textual arguments and to provide a quick overview of public opinions on controversial topics" [4].

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

Our work pursues the fulfillment of the evaluation criteria defined by the Shared Task 3 [1]:

- topic relevance
- argumentativeness
- stance relevance

of the retrieved images regarding the query. These three evaluation criteria represent the assessment of the relevance of an image to a given query that should be considered for an image argument search engine.

Topic relevance indicates the extent to which an image matches the content of the entered search query. **Argumentativeness** states whether the image is suitable for defending a position within the debate on the searched topic. And finally, **stance relevance** evaluates if the image supports the stance (pro or con) it is assigned to [5]. For each of the three evaluation criteria, we compare several approaches: To retrieve a high number of topic relevant images our focus is on preprocessing the query and optical character recognition (OCR). OCR is also used to filter the most argumentative images together with clustering the images according to the type of image (e.g., statistic, text image, image with people). To assign the images to the correct stance we use sentiment analysis. We assume that the sentiment of the web page text the image appears on, correlates with supporting (positive sentiment) or opposing (negative sentiment) a controversial question. In Section 2, we review related work covering argument search, OCR, image clustering and sentiment analysis. Section 3 will present our methodology and Section 4 will discuss our results. Section 5 will summarize this work and point out the limitations of our approach.

2. Related Work

In this section, we clarify the context of Argument Search, Optical Character Recognition, Image Clustering and Sentiment Analysis. It should allow the reader to better comprehend the following sections.

2.1. Argument Search

Wachsmuth et al. [6] propose a framework for argument search. An argument is saved with an ID, the URL of the web page they are found on, and the page's full text. The index is created with Apache Lucene using the argument representations as input. Elasticsearch can be used via a REST API as a user interface to Apache Lucene to create the index and to search it. To allow fast search responses Elasticsearch searches an index instead of the whole text. It creates the so-called inverted index with keywords from the document's text using Apache Lucene [7]. To provide results ranked by relevance to the query Lucene has several standard ranking functions such as Okapi BM25 which relies on term frequency-inverse document frequency (TF-IDF). With slight adaptions, this framework can be applied to results represented as images.

[1] https://webis.de/events/touche-22/shared-task-3.html

2.2. Optical Character Recognition

OCR is used to retrieve text from images automatically. Hamad et al. [8] discuss OCR's challenges, important steps in the pipeline, use cases, and historical details. One of the major challenges is called scene complexity where the intricate image content makes it difficult to distinguish the text from the rest.

One widely-used OCR system was developed by Fedor et al. [9] in 2018. The so-called "Rosetta" is deployed on Facebook and Instagram to evaluate memes in real-time and at scale. First, the system detects text regions of an image using a Faster-R-CNN model [10], and afterwards recognizes letters contained in those regions by applying a CNN model. They point out that high-quality training data containing a variety of fonts, styles, and font sizes is very important for training the OCR model. Even though the procedure is described in detail, the source code has not been made available so we can not try its performance on our data set.

Memes are a very popular image type on the internet. They often contain an opinion and take stance with one side or the other [11]. Beskow et al. [12] worked on characterizing memes and elaborating families of memes. Among other things, they used meme-specific OCR for identifying the text written in the image. The biggest challenge using OCR algorithms is that most are trained on text with black font and white background which is usually not the case for images from the web. To solve the problem, Beskow et al. preprocess the images before using Google's Tesseract OCR [2]. As our dataset also contains many memes, we take their Preprocessing approach as a guideline and also decide for Google Tesseract.

2.3. Image Clustering

In our work we use image clustering to group together images of similar types such as bar plots, pie charts, images of persons demonstrating or persons doing sport. From these image types we derive the importance of the images as an argument depending on their type. Omran et al. [13] give a general overview of clustering methods and addresses some of the most important similarity measures and clustering algorithms. We use the k-Means clustering, which aims to minimize the intra-cluster distance, by adding new data points to the cluster whose centroid has the shortest distance to the new point. Since having to consider a few special things when working with images, Rahmani et al. [14] give a short introduction of how k-means clustering with image data works.

Clustering methods for images mostly use image feature vectors, which are created to represent the contents of a image in form of a multidimensional numerical vector, as input data. Those vectors can be created in multiple ways, for example using machine learning methods like VGG16 [15]. VGG16 is a convolutional network designed for image classification. It was the winning method of the 2014 ImageNet Large-ScaleVisual Recognition Challenge (ILSVRC) for the localization and classification tracks [16]. The net consists of 16 convolutional layers with filters of a receptive field size of 3 x 3. We use a pre-trained VGG16 model for our clustering approach to create image feature vectors.

Another noteworthy approach for creating image feature vectors is the Scale Invariant Feature Transform (SIFT) from Lowe [17]. This method finds and localizes key points in an image and

[2] https://github.com/tesseract-ocr/tesseract

transforms them into image features. Csurka et al. [18] and Yanai [19] use SIFT to generate a so-called bag-of-keypoints. A bag-of-keypoints can be seen as a histogram of the number of occurrences of particular image patterns in an image, similar to the bag-of-words representation for documents. The bag-of-keypoints can then be used to classify an image using a classifier method like SVM. The strength of bag-of-keypoints lies in object detection. We decided against a bag-of-keypoints approach because the aim of our clustering is to separate the different types of images (e.g. pie chart, bar plot, persons demonstrating, ...) rather then identifying the objects pictured on them.

In many cases, the scale of the image collection is too large for computing on a standard processor. Liu et al. [20] propose a method that allows nearest neighbor clustering for a large number of images with the help of parallel distributed hybrid spill trees which are implemented by combining MapReduce operations with a intelligent data partitioning. Even though it is not relevant for the amount of data the Touché Task involves, the aspect of scalability must be kept in mind when working with even larger data sets. As an illustration, modern image search engines can contain billions of images.

2.4. Sentiment Analysis

We determine an image's stance towards a topic by performing a sentiment analysis on the associated web page's text. We assume that sentiments reveal if the image supports or opposes a given query. Nielsen [21] proposes a lexicon based approach to determine the sentiment of a text. The proposed lexicon "AFINN" consists of words that are labeled with a certain sentiment score. The latest version of AFINN consists of 2477 unique words (including phrases) that were manually labeled with an integer value between -5 (very negative) and 5 (very positive). The overall sentiment of a text is the sum of sentiment scores of all words contained in that text.
Another approach to sentiment analysis uses a machine learning classifier with a pre-trained BERT-model (Bidirectional Encoder Representations from Transformers) introduced by Devlin et al. [22]. BERT is a language representation model that can be customized to a variety of tasks by only adding one extra output layer. Possible applications include answering questions or sentiment analysis[23].

3. Methodology

In this section, we describe our approach guided by the three evaluation criteria: topic relevance, argumentativeness, and stance relevance. We start with a summary of the data we used and the workflow. Afterward, we explain in depth the methods used, namely document and query Preprocessing, optical character recognition, image clustering, and sentiment analysis. At the end of the chapter, we propose an evaluation method for comparing the implemented techniques.

Table 1
Data used in our project

Name	Content	Usage
one file for each image in the dataset:		
image.png	Image in PNG format	OCR/Clustering
rankings.jsonl	JSON objects describing a query to Google that retrieved the image. This includes the rank the image got in the Google retrieval	Evaluation
text.txt	Text content of the webpage with the image	Indexing/Sentiment/ Retrieval
dom.html	html source of the webpage	Sentiment
one file for the entire dataset:		
topics.xml	Title, description, and narrative of Touché topics 1 to 50	Evaluation/Development

3.1. Data

The first step for solving Touché Task 3 is to build an understanding of the data. It is accessible under [3] and can be downloaded from there. The download contains data for 23,841 images. For each image, there are 12 different files that can be used to develop retrieval methods. The files are for example giving detailed information about the image itself, the webpage where the image is pictured or the google retrieval for that image. Additionally, with *topics.xml* and *training-qrels.txt*, there are 2 files containing topics and qrels for the entire data set.

As shown in Table 1, we do not need all files but only the following subset in our retrieval approach: We need *image.png* to perform OCR and clustering, *rankings.jsonl* for the evaluation, *text.txt* for indexing, retrieval and to perform sentiment analysis, *dom.html* for sentiment analysis and *topics.xml* for evaluation and development.

3.2. Workflow

Our workflow begins by building the index. As can be seen in Figure 1 we use the crawled web pages' title, visible text and the image that is associated with that webpage.

The title is fed into a Machine Learning algorithm to derive a sentiment that can either be positive or negative.

The pre-processed text of the webpage is used to calculate a sentiment, though here we use a dictionary based approach to determine the score of the text. Furthermore the text is also directly fed into the index to execute queries on it.

The image of a webpage is also used twofold. We scan it for text and put any readable text into the index to be queried with higher weighting than the text of the webpage. Additionally, we analyze the image with a neural network trained for image classification. We use the output of the network to assign the image to an image type.

[3] https://files.webis.de/corpora/corpora-webis/corpus-touche-image-search-22/

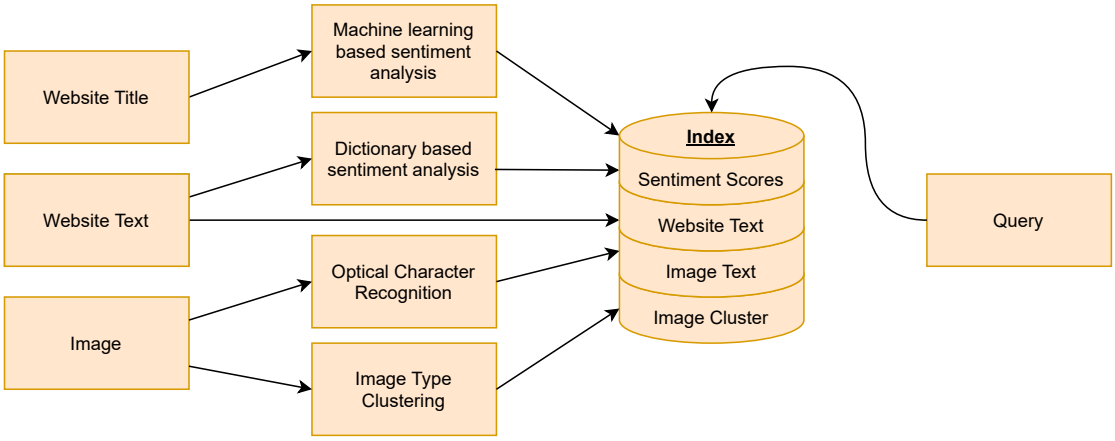


Figure 1: Methods to build the index and query it.

3.3. Document And Query Preprocessing

In our initial approach to pre-process we use the text of the web page as this document is used for building the index. We use the Natural Language Toolkit (NLTK) [24] for the desired adjustments: First, we convert the document to lowercase and remove URLs. We only keep letters and tokenize the text. Next, we lemmatize all tokens and remove tokens consisting of only one letter and tokens included in the stop word list of the NLTK library. Finally, we eliminate the tokens that appear only once or twice in the text, which is a method based on Zipf's law [25]. We find this step necessary because many web pages contain additional information like navigation or a footer. Words from those sections do not appear often on the web page, whereas words important to the main topic of the page will show up frequently. In the next step we pre-process the query. Not all words contained in a query have the same importance to the topic and thereby for the retrieval. Less important words can negatively influence the search. If web pages are found that use only these less important words, those pages might be less relevant to the topic. Therefore, we decide to create our own stop word list with selected words that do not contribute to the statement of the query e.g., "be", "for", "in". We eliminate these words from the query. Furthermore, we lemmatize the remaining words as we did with the text of the web pages. We believe that it is more effective to perform the retrieval when the words in the query and the page text are restored to their non-inflected form in the same way.

3.4. Optical Character Recognition

Our research team shares the impression that images containing text are in the majority of cases more argumentative than images without because the text directly indicates what the image should express. We expect images which contain text from a query to be more relevant than other images.
Optical character recognition is used to identify image text. We decide to build our pipeline using Google's Tesseract OCR [26] as it is one of the most popular freely-available OCR engines and is widely used [27][12]. The implemented text identification pipeline is depicted in Figure

2. First, we binarize the image as OCR techniques work better on black and white pictures [12]. Afterward, we extract the text using Tesseract. It often occurs that random symbols and letters are being extracted from parts of the image where there is no text. Therefore, we decide to keep only letters and check for each word of the extracted text if it is included in an English dictionary.

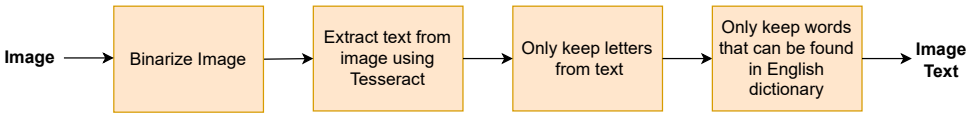


Figure 2: The constructed OCR pipeline.

There are clear limitations for handwritten text e.g., images containing demonstration posters. However, our goal is not to identify the text from each image perfectly but to get an impression of whether OCR can generally improve image retrieval for arguments. It is possible to use commercial OCR engines that claim to achieve more accurate results or to train neural networks for text extraction. The latter would be much more time-consuming and is an ongoing field of research on its own. Based on our results, Tesseract OCR is improving the retrieval and it would be interesting to find out in the future if different OCR systems could improve it even further. After extracting the text of each image, we add it to the index. For each query, we retrieve images based on the web pages text and the text extracted from the images via OCR. The image text gets a boost of five i.e. its score is multiplied by five. We choose five after having tried different numbers. For each image, we compare the boosted image text score with the page text score and use the higher one for the retrieval.

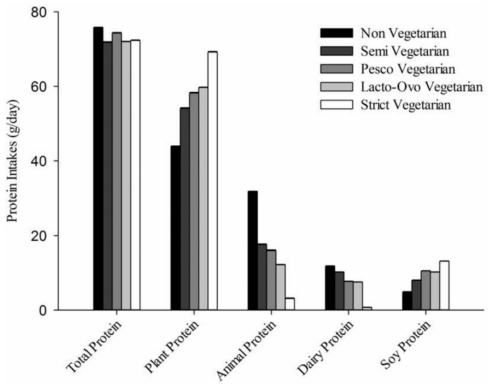
3.5. Image Clustering

Another approach to increase the argumentativeness of the results is to categorize images into classes of different image types. There are plots, memes, landscape photos, portraits, and many more possible classes. The idea behind using image clustering is, that some of the classes are more argumentative than others. Therefore images of these classes should get higher attention than images of less argumentative classes. This can be shown by the exemplary topic "Should people become vegetarian?". Figure 3 (a) shows a bar plot of the protein intake of vegetarians and non-vegetarians. Images like this can easily be used to argue for or against a topic. In comparison, Figure 3 (b) shows different kinds of fruits and vegetables in the shape of a heart. Compared to the argumentative benefit of a bar plot and other statistics, the argumentativeness of a symbolic image like this is limited, because it does not provide additional facts to support a textual argument with.

Our goal is to find clusters containing images with the same level of argumentativeness. We do this by using an image clustering method based on feature vectors and k-means clustering[4]. The detail of the processing is as follows:

1. Load a pre-trained VGG16 model for image classification

[4]cf. https://towardsdatascience.com/how-to-cluster-images-based-on-visual-similarity-cd6e7209fe34



(a) Bar plot of protein intake by diet (very argumentative)

(b) Image of fruits and vegetables in the shape of a heart (not as argumentative as Figure 1(a))

Figure 3: Retrieval candidates for the topic 'Should people become vegetarian?'

2. Cut off the original output-layer and use the model to calculate feature vectors, which are used to represent the contents of the images as numerical vectors
3. Reduce the dimension of the feature vectors with Principal Component Analysis (PCA) for faster computing
4. Make an elbow plot for k-means clustering to find possible number of clusters
5. Make k-means clustering with the chosen number of clusters

Figure 4 shows the elbow plot from which we derive the number of clusters to be found by k-means. The plot shows the Within Cluster Sum of Squared Errors (WCSS) when using different numbers of clusters [28]. In theory, the optimal number of clusters is a number where the WCSS stops decreasing significantly with the addition of one cluster. This point is usually recognizable by a clear inflection point in the displayed curve. Our plot shows no clear indication for one optimal number of clusters, but because of the flattening character of the curve in the range between 10 and 20, anywhere between 10 clusters and 20 clusters seems reasonable.

To find the final number we need a deeper inspection of the differences in the clustering results, when choosing different numbers of clusters. We repeat the k-means clustering for all possible numbers of clusters between 10 and 20. Then, for each possible number, we look into examples of images that are assigned to the different clusters and compare them to other results. After manually inspecting all possibilities between 10 and 20 clusters, we found that 14 clusters is a good number to build our image clustering up on.

After the identification of the clusters, we define a weight for each cluster, based on the level of argumentativeness of the image type each cluster represents. A strength of this approach is, that the determination of the weights is variable. However, it is difficult to find an objective heuristic for determining a cluster's weight, because the perceived level of argumentativeness

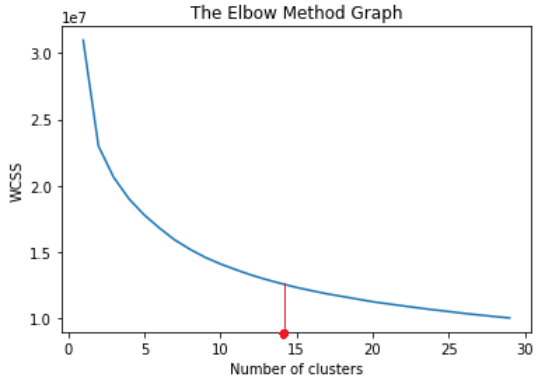


Figure 4: The elbow method graph indicates the optimal number of clusters of image types in the argumentative image dataset is between 10 and 20. After manually inspecting the clusters, we decided on 14 (red).

of each cluster is subjective. In our case we determine weights between 1 (for lowest level of argumentativeness) to 5 (for the highest level of argumentativeness) for each cluster. Table 2 gives a overview of the clusters and the weights we use in our retrieval system. It shows that we give the highest weight of 5.0 to the clusters *5,8,10* and *13* which contain a high amount of statistical graphics or other graphics with text on them. We give a weight of 3.0 to the clusters *3,6,9* and *12* which contain cartoons, memes or other graphics with text. Cluster 7 is the only one with a weight of 2.0 and contains photos of protesters with posters. The remaining clusters *0,1,2,4* and *11* get a weight of 1.0 and are mostly containing some sort of photos without text on them.

The information to which cluster each image belongs is added to the index. After that, when retrieving images, the weight for each cluster is implemented into the query in a way that it will be multiplied with the relevance score of the retrieval. Hence, the method will influence the relevance ranking of an image, depending on the cluster the image belongs to.

3.6. Sentiment Analysis

Our approach to stance classification assumes an underlying sentiment of the web page's content that can be used to determine whether the web page is supporting or opposing a query. In a first experiment, we use dictionary-based sentiment analysis on the whole text content of the web pages embedding the image. In a second experiment, we use a BERT-based method to label the titles of web pages that embed the images to positive and negative sentiments.

Dictionary-Based Approach: We use the complete textual content of the web page from the text.txt file to compare each word with the AFINN dictionary and get a corresponding score that is either positive (from 1 to 5), negative (from -1 to -5), or neutral (0). The sum of scores of all words gives the overall score of a page's content. We assume that a positive score represents a pro stance, a negative score a contra stance, and zero no stance. We add the scores to the index and modify the query to give two result sets, one for positive scores (higher than zero) as

Table 2
Identified clusters by k-means, the weights specified by us and the number N of images within each cluster

Cluster	Description	Weight	N
0	photos with round objects (e.g. pills, coins, bottle caps)	1.0	1,886
1	mostly photos but no real context identified	1.0	2,027
2	photos of objects people hold in their hands (e.g. guns, vapes, syringes)	1.0	1,631
3	cartoons, cartoon-like memes and maps	3.0	2,216
4	photos of groups of people (e.g. athletes, police, protesters)	1.0	1,452
5	graphics with text (e.g. memes, quotes, twitter posts)	5.0	860
6	statistical graphics (e.g. horizontal bar plots, line plots, scatter plots) and thesis covers	3.0	1,465
7	photos of protesters with posters	2.0	2,012
8	graphics with round forms and text (e.g. pie charts)	5.0	2,155
9	memes with faces as component	3.0	1,458
10	statistical graphics but with better quality as in 6 (e.g. bar plots, tables, line plots)	5.0	2,384
11	photos of children (... making homework, getting vaccinated, eating in canteen)	1.0	1,500
12	graphics with text (e.g. newspaper headlines, quotes, information graphics)	3.0	1,309
13	statistical plots (bar plots and line plots)	5.0	1,485

the pro side and one for negatives scores (lower than zero) as the con side. Images with the score zero are not considered in the results.

Machine-Learning-based method: For this approach, we fine-tune a pre-trained BERT model with movie reviews from the Internet Movie Database (IMDB) to specialize the model on sentiment [29]. This model accepts input phrases of up to 512 words only. Because the entire text of the web page is usually longer than the web page's title, we used the latter to determine the web page's sentiment. The assumption is that the title will represent a web page's content and thus, their sentiment. The BERT model we use is the BERT-base-uncased model from "Hugging Face".[5]

3.7. Evaluation Method

Corresponding to the three evaluation criteria (topic relevance, argumentativeness and stance), our evaluation method is threefold. The easiest case is evaluating if an image was relevant for the topic. The dataset is provided by the organizers of the shared task. They obtained the images by conducting Google Image searches for all topics from the provided topics.xml file in the dataset as queries. To verify if a retrieved image fits the entered query, we check from which Google image search query it was obtained. This information is given in the rankings.jsonl file. Our retrieval systems retrieve ten images supporting and ten images opposing a query.

[5] cf. https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671

Table 3
Inter-annotator Agreement

coefficient name	Fleiss' kappa	Krippendorff's Alpha	AC1 (Gwet's) identity weights	AC2 (Gwet's) quadratic weights
coefficient value	5.9%	10.6%	13.2%	84.9%
confidence interval	(-0.265, 0.384)	(-0.218, 0.431)	(-0.066, 0.33)	(0.613, 1)
p-value	0.638	0.414	0.137	0.00057

We perform the retrieval for the 50 given topics and calculate the percentage of fitting images based on their ranking in the Google Image search results for the same query.

There is no information given if an image is argumentative or if it has the correct stance. Therefore, we evaluate those two evaluation criteria manually. We pick out five topics to evaluate. The team members evaluate the retrieved images for the five topics independently. For each run, we indicate how many of the images on the pro side are argumentative and how many of those are on the correct side. We repeat the same procedure for the con side. We then form the average across all topics and individual ratings and calculate the percentages of retrieved images that are argumentative respective stance relevant.

To get an idea of our overall consensus in the evaluation, we calculate several statistics used for ratio data and more than two raters: Fleiss' Kappa, Krippendorff's Alpha, and Gwet's coefficients AC1 and AC2 [30].

Table 3 shows the calculated coefficients. The percent agreement increases from left: Fleiss' kappa with 5.9% to right: AC2 with 84.9%. The general representation of percent agreement is shown in (1) and can be summarized with: the actual agreement that was not caused by chance divided by perfect agreement not caused by chance.

$$\frac{A_w - A_c}{1 - A_c} \tag{1}$$

where A_w = weighted percent agreement
A_c = percent chance agreement

It is known that Fleiss' kappa, Krippendorff's alpha, and AC1 "overstate the percent chance agreement" resulting in an understated percent agreement [30]. We can also see that the p-values of those methods are higher than a 5% alpha-value while the p-value of AC2 with 0.00057 is even lower than a 1% alpha-value. That is why we assume an 84.9% agreement within our evaluation.

4. Results

The following chapter compares the performance of the implemented methods. We aim to find out which methods work best together, considering the three evaluation criteria and the overall

performance.
We opt for six different retrieval systems which we compare to the method proposed by Kiesel et al. [5]. We replicate their approach which expands the query with the term "pro" for the supporting side and the term "anti" for the opposing side. This retrieval system is denoted with the number 0 in the following graphics. System number 1 uses only the dictionary-based sentiment analysis for stance allocation. The second system extends system 1 by usage of OCR, whereas the number 3 uses image clustering on top of the dictionary-based sentiment analysis. System number 4 combines the same sentiment analysis, the OCR, and the image clustering. Additionally, system number 5 adds query Preprocessing. For retrieval system number 6 we use the same components as number 5 but exchange the dictionary-based sentiment analysis with the machine-learning-based sentiment analysis.

We deployed our systems within the "TIRA" platform that was set up by the workshop officials [31]. There, we submitted five different approaches at first, that can be recognized by their timestamps in table 4. We included a short description of every system. A sixth system was submitted, that we did not anticipate to perform as well as it did. This sixth combination of methods ultimately performed best out of all in Touché Task 3. We did not analyze this system further because we did not expect that it would perform better without the clustering.

Table 4
Systems submitted as runs on the TIRA-platform

No.	Description	Timestamp
1	Afinn-Sentiment, OCR	2022-02-26-21-13-41
2	Afinn-Sentiment, Clustering	2022-05-03-07-54-56
3	Afinn-Sentiment, OCR, Clustering	2022-02-26-21-45-32
4	Afinn-Sentiment, OCR, Clustering, Query Preprocessing	2022-02-26-21-59-50
5	Afinn-Sentiment, OCR, Clustering, Query Preprocessing	2022-02-27-18-02-37
6	Afinn-Sentiment, OCR, Query Preprocessing	2022-06-17-21-01-29

Figure 5 reveals the results of the seven retrieval systems regarding the topic relevance of the retrieved images. Only the replicated system 0 obtains less than 80% of topic-relevant images. Especially the retrieval systems 1, 2, 3 and 5 perform best with over 84%. However, the distances are not large. Retrieval systems number 6 perform better for the con-side than any other retrieval systems.

The next evaluation criterion is to retrieve argumentative images. Figure 6 shows how well the retrieval systems perform regarding this criterion. It is important to note that in our point of view, only topic-relevant images can also be argumentative for this topic. This means that the reached percentage of topic-relevant images corresponds to 100% in the evaluation of the argumentativeness. By doing so, we can assess the evaluation criteria independently from each other and find out which retrieval systems are best regarding each evaluation criterion separately. The retrieval system's performances for retrieving argumentative images are far apart. As to be expected the retrieval systems 0 and 1 perform poorly, both of them are based on techniques made for boosting the topic relevance and not the argumentativeness. The more techniques we add to the retrieval systems the better the results. The retrieval systems 4 and 6 perform best. We record an improvement from 64% with retrieval system 0 to 89% with system

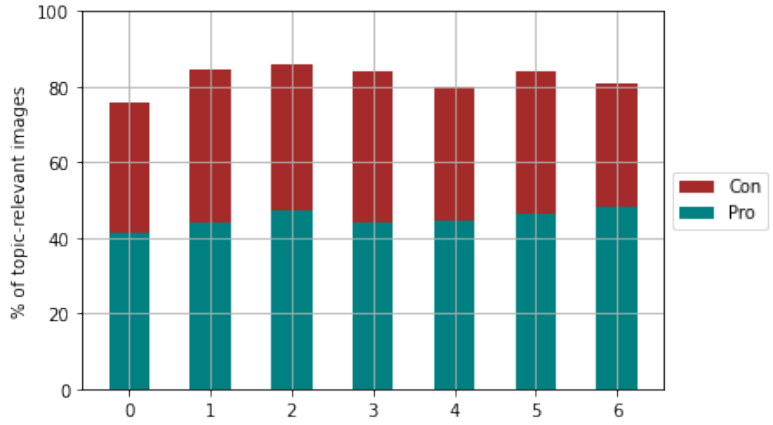


Figure 5: Percentage of how many topic-relevant images are retrieved by the different retrieval systems for the pro (blue) and the contra (red) side. (0 - query expansion, 1 - dictionary-based sentiment analysis, 2 - dictionary-based sentiment analysis + OCR, 3 - dictionary-based sentiment analysis + Image Clustering, 4 - dictionary-based sentiment analysis + OCR + Image Clustering, 5 - dictionary-based sentiment analysis + OCR + Image Clustering + Query Preprocessing, 6 - machine-learning-based sentiment analysis + OCR + Image Clustering + Query Preprocessing)

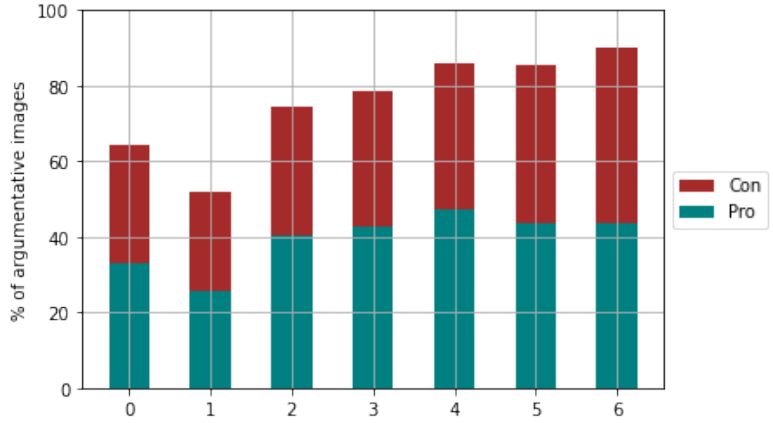


Figure 6: Percentage of how many argumentative images are retrieved by the different retrieval systems for the pro (blue) and the con (red) side. (0 - query expansion, 1 - dictionary-based sentiment analysis, 2 - dictionary-based sentiment analysis + OCR, 3 - dictionary-based sentiment analysis + Image Clustering, 4 - dictionary-based sentiment analysis + OCR + Image Clustering, 5 - dictionary-based sentiment analysis + OCR + Image Clustering + Query Preprocessing, 6 - machine-learning-based sentiment analysis + OCR + Image Clustering + Query Preprocessing)

6 - a clear sign that the applied methods have a positive effect on retrieving argumentative images.

The most difficult evaluation criterion to achieve is assigning the correct stance (pro or contra) to the images. As can be observed in figure 7 all retrieval systems are struggling to figure out whether an image is supporting the query or not. On the pro-side, the retrieval

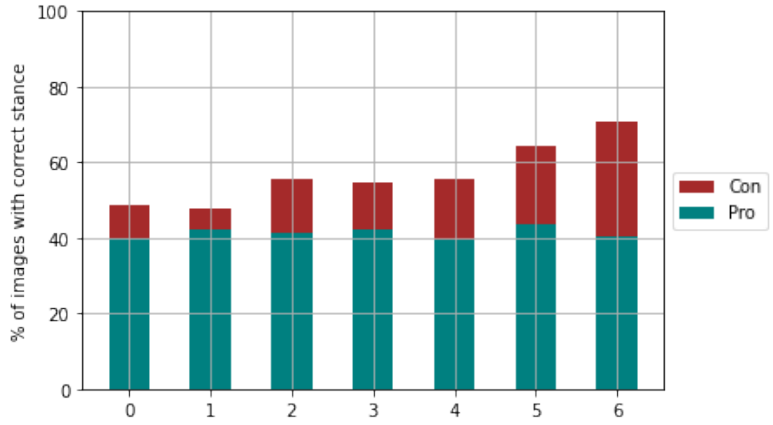


Figure 7: Percentage of how many correctly classified images are retrieved by the different retrieval systems for the pro (blue) and the con (red) side. (0 - query expansion, 1 - dictionary-based sentiment analysis, 2 - dictionary-based sentiment analysis + OCR, 3 - dictionary-based sentiment analysis + Image Clustering, 4 - dictionary-based sentiment analysis + OCR + Image Clustering, 5 - dictionary-based sentiment analysis + OCR + Image Clustering + Query Preprocessing, 6 - machine-learning-based sentiment analysis + OCR + Image Clustering + Query Preprocessing)

systems do similarly well with about four out of five correct assignments. But the images on the contra-side often do not oppose the query. Retrieval system 6 using the machine-learning-based sentiment analysis managed to classify a majority of contra-images correctly. We accomplish an improvement from 48% with system 0 to 71% using system 6.

Regarding all three evaluation criteria, we obtain the results depicted in Figure 8. To understand the plot, it is important to know that only images that correspond to the topic can be argumentative for that topic and only those images argumentative for a topic can be assigned a correct side. Images in the red part on the bottom of the plot achieve all three evaluation criteria, whereas the brown part in the middle is argumentative but does not have the correct stance. The beige top part contains images that match the topic but are not argumentative. Retrieval systems 0 and 1 which only use the query expansion respective the dictionary-based sentiment analysis perform weakest with around 26% and 21% of images fulfilling all evaluation criteria. The three following retrieval systems 2, 3, and 4 score 39%, 39%, and 42%. Finally, the best results are produced by retrieval systems 5 with 48% and 6 with 52%. Comparing the replicated system designed by Kiesel et al.[5] to our best retrieval system, we achieve an improvement of the retrieval performance by 26%, which doubles that of system 0.

5. Conclusion

Intending to retrieve argumentative images supporting or opposing an entered query, we built six retrieval systems using ElasticSearch. The implemented retrieval systems use different combinations of the following techniques: Query Preprocessing, Optical Character Recognition, Image Clustering, dictionary-based, and machine-learning-based sentiment analysis. All six retrieval systems were evaluated regarding topic relevance, argumentativeness, and stance

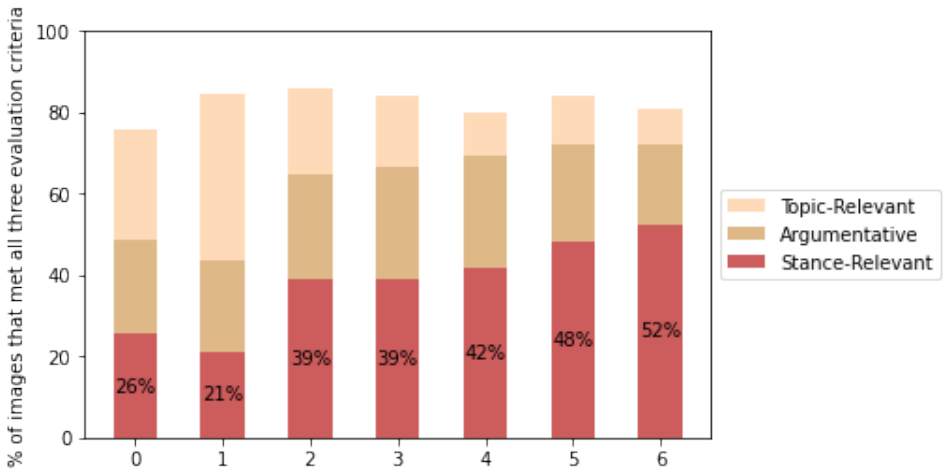


Figure 8: Percentage of how many images meet all three evaluation criteria (pink) retrieved by the different retrieval systems (0 - query expansion, 1 - dictionary-based sentiment analysis, 2 - dictionary-based sentiment analysis + OCR, 3 - dictionary-based sentiment analysis + Image Clustering, 4 - dictionary-based sentiment analysis + OCR + Image Clustering, 5 - dictionary-based sentiment analysis + OCR + Image Clustering + Query Preprocessing, 6 - machine-learning-based sentiment analysis + OCR + Image Clustering + Query Preprocessing)

relevance of the retrieved images. The last two evaluation criteria were evaluated manually by four independent annotators. Our best retrieval system used a combination of all mentioned techniques except the dictionary-based sentiment analysis. We compared our retrieval systems to the only search engine we found that was implemented for this sort of image retrieval designed by Kiesel et al. [5]. We replicated their approach. Our best retrieval system is able to improve it by 26% regarding all three evaluation criteria.

5.1. Limitations

Nevertheless, our approach is constrained by some limitations. We observed that several images were informative and thus useful to build an opinion on the topic but did not clearly represent either side (pro or con). This often is the case with statistics showing the results of a survey. Depending on the viewer's interpretation the image can support the pro or the con side.

As an example, Figure 9 presents people's opinions on abortion depending on the time of the abortion and the political opinion of the respondents. The plot opens up different interpretation possibilities. Particularly, the viewer's political opinion may influence if the plot is interpreted as supporting or opposing abortion. Nevertheless, this graph can contribute to answering the question of whether abortion should be legal as it contains a lot of information.

Another observation was that even though the retrieval systems work well on many topics some topics do not have as many supporting arguments in form of images as opposing them or the other way around. An example of this is the ban on bottled water. Many arguments support the ban such as environmental friendliness and the equal quality of tab and bottled water, while the opposing side does not reveal any relevant arguments. We conclude that not all

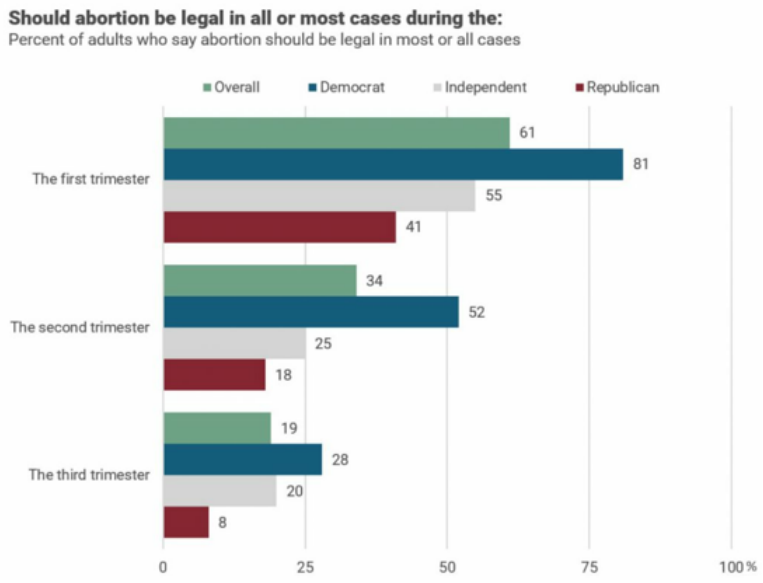


Figure 9: Legality of abortion depending on time of abortion and political opinion

arguments are equally represented on the internet thus in our database and results. This limits the representativeness of an argument search engine to topics and opinions that are frequently discussed with a diversity of arguments.
Also, more annotators would improve the significance of the results because the agreement would be more reliable. A more diverse group of annotators could help represent the view of a bigger part of society.

5.2. Future Work

In the future, we want to add more approaches to our list and work on the limitations. One idea is to train Convolutional Neural Networks to perform the OCR to improve text identification. Another idea is to analyze an image's colors and their distribution to find out how argumentative the image is and which stance it supports. We want to introduce an additional stance category besides pro and con, that contains informative images that can not be assigned to either pro or con. We would also like to have more people conduct the manual evaluation to increase the validity of the results.
We would expect the retrieval performance to improve further by adding synonyms to the query. Thereby, we could also retrieve images that appear on web pages treating the same topic but use different words. Instead of applying sentiment analysis only on the title or the entire webpage we want to use it also on the image text or only on an excerpt of the webpage that is of higher importance to the image. Also, we would like to find out, if the retrieval performance could be enhanced by classifying the images into predefined classes instead of clustering them.

Acknowledgments

We would like to thank the Webis research group for giving helpful advice and always being available for upcoming questions. A special thanks goes to Theresa Elstner, who took time for us every week to discuss the current status.

References

[1] I. Rahwan, F. Zablith, C. Reed, Laying the foundations for a world wide argument web, Artificial Intelligence 171 (2007) 897–921. doi:10.1016/j.artint.2007.04.015.

[2] J. A. Blair, The possibility and actuality of visual arguments, in: Groundwork in the Theory of Argumentation, Springer, Dordrecht, 2012, pp. 205–223. doi:10.1007/978-94-007-2363-4_16.

[3] B. Heiskanen, Meme-ing electoral participation, European journal of American studies 12 (2017). doi:10.4000/ejas.12158.

[4] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of touché 2022: Argument retrieval, in: Advances in Information Retrieval. 44th European Conference on IR Research (ECIR 2022), Springer, 2022.

[5] J. Kiesel, N. Reichenbach, B. Stein, M. Potthast, Image Retrieval for Arguments Using Stance-Aware Query Expansion, in: K. Al-Khatib, Y. Hou, M. Stede (Eds.), 8th Workshop on Argument Mining (ArgMining 2021) at EMNLP, Association for Computational Linguistics, 2021, pp. 36–45. URL: https://aclanthology.org/2021.argmining-1.4/. doi:10.18653/v1/2021.argmining-1.4.

[6] H. Wachsmuth, M. Potthast, K. Al Khatib, Y. Ajjour, J. Puschmann, J. Qu, J. Dorsch, V. Morari, J. Bevendorff, B. Stein, Building an argument search engine for the web, in: Proceedings of the 4th Workshop on Argument Mining, Association for Computational Linguistics, 2017, pp. 49–59. doi:10.18653/v1/w17-5106.

[7] M. S. Divya, S. K. Goyal, Elasticsearch: An advanced and quick search technique to handle voluminous data, Compusoft, An international journal of advanced computer technology 2 (2013) 171–175.

[8] K. A. Hamad, K. A. Y. A. Mehmet, A detailed analysis of optical character recognition technology, International Journal of Applied Mathematics Electronics and Computers (2016) 244–249.

[9] F. Borisyuk, A. Gordo, V. Sivakumar, Rosetta: Large scale system for text detection and recognition in images, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018) 71–79.

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[11] L. Shifman, Memes in a digital world: Reconciling with a conceptual troublemaker, Journal of computer-mediated communication 18 (2013) 362–377.

[12] D. M. Beskow, S. Kumar, K. M. Carley, The evolution of political memes: Detecting and

characterizing internet memes with multi-modal deep learning, Information Processing & Management 57 (2020). doi:10.1016/j.ipm.2019.102170.

[13] M. G. Omran, A. P. Engelbrecht, A. Salman, An overview of clustering methods, Intelligent Data Analysis 11 (2007) 583–605. doi:10.3233/IDA-2007-11602.

[14] M. K. I. Rahmani, N. Pal, K. Arora, Clustering of image data using k-means and fuzzy k-means, International Journal of Advanced Computer Science and Applications 5 (2014) 160–163. doi:10.14569/IJACSA.2014.050724.

[15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. URL: http://arxiv.org/pdf/1409.1556v6.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[17] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

[18] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on statistical learning in computer vision, ECCV, volume 1, Prague, 2004, pp. 1–2.

[19] K. Yanai, Image collector iii: a web image-gathering system with bag-of-keypoints, in: Proceedings of the 16th international conference on World Wide Web, ACM Press, 2007, pp. 1295–1296. doi:10.1145/1242572.1242816.

[20] T. Liu, C. Rosenberg, H. Rowley, Clustering billions of images with large scale nearest neighbor search, in: 2007 IEEE Workshop on Applications of Computer Vision (WAC V '07), IEEE, 2007, pp. 28–28. doi:10.1109/WACV.2007.18.

[21] F. Å. Nielsen, A new ANEW: Evaluation of a word list for sentiment analysis in microblogs, 2011.

[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL: https://arxiv.org/pdf/1810.04805.

[23] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification?, ???? URL: http://arxiv.org/pdf/1905.05583v3.

[24] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, "O'Reilly Media, Inc.", 2009.

[25] H. Saif, M. Fernandez, Y. He, H. Alani, On stopwords, filtering and data sparsity for sentiment analysis of twitter (2014).

[26] R. Smith, An overview of the tesseract ocr engine, Ninth international conference on document analysis and recognition (ICDAR 2007) Vol. 2. (2007).

[27] A. Amalia, A. Sharif, F. Haisar, D. Gunawan, B. B. Nasution, Meme opinion categorization by using optical character recognition (ocr) and naïve bayes algorithm, in: 2018 Third International Conference on Informatics and Computing (ICIC), IEEE, 2018, pp. 1–5. doi:10.1109/iac.2018.8780410.

[28] C. Yuan, H. Yang, Research on k-value selection method of k-means clustering algorithm, J 2 (2019) 226–235. doi:10.3390/j2020016.

[29] S. Alaparthi, M. Mishra, Bert: a sentiment analysis odyssey, Journal of Marketing Analytics 9 (2021) 118–126. URL: https://link.springer.com/article/10.1057/s41270-021-00109-8.

doi:10.1057/s41270-021-00109-8.
[30] K. L. Gwet, Handbook of Inter-Rater Reliability, 4th Edition: The Definitive Guide to Measuring The Extent of Agreement Among Raters, Advanced Analytics, LLC, 2014.
[31] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.