Fine and Coarse Granular Argument Classification before Clustering

Lorik Dumani
dumani@uni-trier.de
Trier University
Trier, Germany

Tobias Wiesenfeldt
twiesenf@students.uni-mainz.de
Johannes Gutenberg University
Mainz, Germany

Ralf Schenkel
schenkel@uni-trier.de
Trier University
Trier, Germany

ABSTRACT

Computational argumentation and especially argument mining together with retrieval enjoys increasing popularity. In contrast to standard search engines that focus on finding documents relevant to a query, argument retrieval aims at finding the best supporting and attacking premises given a query claim, e.g., from a predefined collection of arguments. Here, a claim is the central part of an argument representing the standpoint of a speaker with the goal to persuade the audience, and a premise serves as evidence to the claim. In addition to the actual retrieval process, existing work has focused on (1) classifying polarities of arguments into supporting or opposing, (2) classifying arguments by their frames (such as *economic* or *environmental*), and (3) clustering similar arguments by their meaning to avoid repetitions in the result list. For experiments, either hand-made argument collections or arguments extracted from debate portals were used. In this paper, we extend existing work on argument clustering, making the following contributions: First, we introduce a novel pipeline for clustering arguments. While previous work classified arguments either by polarity, frame, or meaning, our pipeline incorporates these three, allowing a more systematic presentation of arguments. Second, we introduce a new dataset consisting of 365 argument graphs accompanying more than 11,000 high-quality arguments that, contrary to previous datasets, have been generated, displayed, and verified by journalists and were published in newspapers. A thorough evaluation with this dataset provides a first baseline for future work.

CCS CONCEPTS

• **Information systems** → Information extraction; **Clustering and classification**; Summarization; *Test collections*.

KEYWORDS

argumentation, argument classification, argument framing, argument clustering, argument retrieval dataset

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482431

ACM Reference Format:
Lorik Dumani, Tobias Wiesenfeldt, and Ralf Schenkel. 2021. Fine and Coarse Granular Argument Classification before Clustering. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482431

1 INTRODUCTION

Argumentation exists as long as humans and in all possible verbal and non-verbal forms. People use arguments, in order to be able to form opinions to themselves to particular topics, e.g., which mobile phone to buy, or to persuade others of own, for instance, political points of view and to possibly even induce to certain actions. Often the literature defines an *argument* as a *claim* that is supported or attacked by a *premise* [34]. The claim describes a controversial point of view that can be accepted (or refuted) only if it is supported (or attacked) by premises [33]. In the literature, the terms premise and argument are often used synonymously. An example for a claim could be "*vaccinations should be mandatory*", an example for a supporting premise (*pro*) could be "*vaccinations protect us from deadly diseases*", and an example for a refuting premise (*con*) could be "*poor people cannot finance vaccinations*". So far researchers often used sentence-level arguments for convenience. This comes with the crucial disadvantage that both supporting and attacking parts may appear in the same premise, which is often used for rhetorical reasons. For example, it might be very difficult to classify the (main) stance of the sentence "*Vaccinations protect us from deadly diseases but poor people cannot finance them*" not only for systems, but also for humans. We therefore advocate to work with *argument units* [37]. These are usually small spans of text in sentences that clearly position themselves for or against a topic. Using these, we avoid the problem of some sentences being both for and against a topic. Since argument units are typically determined by mining, this works on both well-formed and user-generated texts. Note that it is not the intention of the paper to divide sentences into argument units, but merely to work with them. For the remainder of this paper, we sometimes use the terms argument and argument unit synonymously.

Modern argument search engines support users in finding arguments for their queries in form of questions, claims, or keywords [32, 39]. Such systems usually obtain their arguments by extracting them from various heterogeneous Web documents and store them in a preceding step for the sake of efficiency. Given a user query, the systems compute similarities to the precomputed arguments ad-hoc and display a ranked list of premises to the user. Very frequently, the engines divide the resulting arguments into different *stances*, i.e., pro or con, so that a user can weigh up before

Table 1: Example showing premises related to the claim "vaccinations should be mandatory" in different clusters with different frames. Positive argument units towards the claim are highlighted in blue, negative argument units towards the claim are highlighted in orange.

premise	cluster	*frame*
vaccinations protect us from deadly diseases	c_1	*health*
producing vaccinations can be very expensive	c_2	*economy*
poor people cannot finance vaccinations	c_3	*economy*
the lower class may not be able to afford vaccinations	c_3	*economy*

deciding on a particular opinion. Considering the fact that most existing argument search engines derive their arguments from various debate portals [10, 39] (such as debate.org or idebate.org), where the premises are actually answers in form of posts on questions (though they can also stem from other Web documents [32]), it is obvious that there can be many semantically similar premises for a query in the result set. Consequently, it makes sense to remove near duplicates or to cluster them by their meaning.

Although argument clustering is relatively under-researched, there has been some work on it [2, 10, 27]. In particular, the main approach here is to first generate (contextualized) embeddings (e.g. using BERT [9]) of the arguments, and then to perform an agglomerative clustering. In order to visualize this procedure, but also its limitations, let us examine the example in Table 1. Assuming that the table represents an argument base, current argument search engines would classify the last two premises as having the same meaning and show only one of them as a representative of this cluster. However, they fail in handling the parent theme in column "*frame*". More precisely, for a given particular stance, when an argument emphasizes a particular aspect of a topic while omitting other aspects, it is referred to as *framing* in the social sciences [1, 11]. These systems only return a set of final clusters (here c_1, c_2, and c_3), but miss an additional grouping of results by frame, which would make it easier for the user to understand the result list. For example, the first premise deals with the higher-level frame "*health*", while the last three may be assigned to the frame "*economy*". This is especially important because having (even a few) arguments of the "right" frame(s) might convince the audience better of a speaker's position than having a large number of arguments that the listener cannot relate to, even if they are scientifically proven [1]. Note that there is already work on frame classification (see Section 2), but not integrated into a pipeline for clustering arguments. Another limitation of current systems is the fact that they do not utilize argument units but longer texts as they often occur in debate portals. Table 2 shows the main drawback of not clustering argument units but whole sentences and emphasizes its importance. The example shows two premises addressing the topic "*vaccinations*" where it is impossible to cluster all relevant text spans and a representative is shown only once.

In this paper, we tackle these previously mentioned limitations and make the following contributions: (1) We define an argument clustering pipeline in which we first classify frames and stances of all arguments before we cluster these subsets. We will show that this outperforms state-of-the-art baselines which solely rely on

Table 2: Example showing a case where it is impossible to cluster all relevant arguments (each a sentence) for the claim "vaccinations should be mandatory" in different clusters with clear distinguished stances and showing all argument units with the same meaning without repetition. Positive argument units towards the claim are highlighted in blue, negative argument units towards the claim are highlighted in orange.

frames	premise
finance, health	*vaccinations can be expensive* *but* *reduce death rates*
health, health	*vaccinations reduce death rates* *but* *can have adverse reactions*

hierarchical clustering. (2) We present a new dataset consisting of 11,266 high-quality argument units on a total of 365 queries, based on De Argumentenfabriek, which was originally produced by journalists over a longer period of time in Dutch language. We translated this into English, validated it, and made it freely available in two languages to interested researchers.[1] Moreover, we present a focused dataset for evaluation with unified stances and frames.

Next, Section 2 introduces related work and elaborates the differences of prior works to ours. Then we present our pipeline in Section 3. In Section 4 we describe our dataset in more detail. Then, Section 5 outlines the implemented methods and presents the evaluation of our approach. Finally, we conclude the paper with Section 6 and provide an outlook to future work.

2 RELATED WORK

The information age and the associated increase in the amount of data flowing through the Internet, as well as the power of today's computers, have made the research field of computational argumentation highly important. Today, research focuses on several sub-areas, with argument mining [4, 17] and argument retrieval [10, 32, 39] still being the two largest research branches in this community. While the former deals with the extraction of arguments from natural language texts, the latter deals with information-seeking aspects which are typical in the IR area, such as finding arguments for user-specified query claims, as well as ranking or clustering them. Two more recently emerged branches of research deal with the quality of arguments [38, 40], i.e., the persuasiveness of arguments, on the one hand, and with the validation of arguments [35] on the other hand.

Since this paper addresses both a clustering technique for arguments based on existing work as well as a new dataset, we discuss related work on them and highlight the differences to our work.

Detecting Frames in Arguments. While the identification of frames in arguments has been considered before, the research towards aggregating arguments into frames is largely unstudied [1]. Besides the differently applied names for frames (e.g. aspects [27, 36], facets [19], or frames [1, 21]), there are also conceptually different approaches [1]. For example, one fundamental difference in existing works is the question whether to conduct this task with topic-specific [6, 11], generic (i.e., a fixed set that is not related to the topic) [5, 14], or combined frames [7, 21]. Closely related to this

[1] The dataset is available at the following Website: https://zenodo.org/record/4813727.

question is whether to use a supervised [21] or an unsupervised [1] approach to predict frames.

Misra et al. [19] introduce a pipeline by first extracting the most essential arguments for a given conversation and then putting their frames in relation. Their underlying dataset is the Internet Argument Corpus (IAC) [41]. Naderi and Hirst [21] work with generic frame classification not at the document level, but at the sentence level. They propose a supervised approach based on deep neural networks and distributional representations for classifying frames in news articles. They represent the meaning of the frames using Bi-LSTMs and gated recurrent networks. They reach an accuracy of 0.537 for 16 frames. Ajjour et al. [1] define the task of frame identification as splitting a set of arguments into a set of exclusive and non-overlapping frames. Their method first removes topical features from the arguments and then clusters the arguments into frames. Since we will use their method as one of two baselines, we will discuss it later in more detail (see Section 5).

Clustering of Arguments. Clustering arguments becomes crucial mainly for finding premises for a user query in a search scenario, since the arguments in current argument retrieval systems typically origin from heterogeneous sources such as debate portals [10, 32, 39] and the goal should be to present semantically similar arguments to a user exactly once. While there is already a large body of work on argument retrieval [10, 32, 39], argument clustering remains a rather poorly explored field [27].

To the best of our knowledge, Boltužić and Šnajder [2] conducted the first work on clustering arguments. They took arguments from debate portals, computed their WORD2VEC embeddings as features, and formed an agglomerative clustering to find similar arguments. Reimers et al. [27] built on this approach and extended it with contextualized embeddings, i.e., BERT [9] and ELMo [24], which in contrast to WORD2VEC [18] or GLOVE [23], take the context of a word into account, wherefore a homonym can be mapped to different vectors depending on its meaning. They show how to obtain a better (hierarchical) frame-based clustering for topic-dependent arguments using these. They also demonstrated how to use these embeddings to improve the classification of arguments. However, contrary to our work, they used two different datasets for classification and clustering. Moreover, they classify arguments on the one hand on sentence level and not in argument units, and on the other hand not in stance or frame as we do, but only in terms of relevance to a given topic. They also evaluate the dataset of Shnarch et al. [31], which divides arguments into evidence or no evidence. In our work, we rather refer to classifying frames within an argument, e.g. "*financial*" or "*environmental*". In general, the focus of their work is rather to get the maximum out of the contextualized embeddings. Moreover, since they outperform the approach of Misra et al. [19] with their supervised methods, we include a variant of their method as one of two baselines for both clustering arguments by frames and meanings. More precisely, we use the variant of our prior work [10] where we also studied the clustering of arguments from debate portals. Unlike Reimers et al. [27], we did not use a constant tree cut for the final cluster determination, but a dynamic one [16]. Apart from that, we used Sentence-BERT [26] to determine embeddings which uses a siamese and triple network and produces better embeddings than BERT when used out-of-the-box.

Datasets. As previously pointed out, argument mining and argument retrieval are the two largest research blocks in computational argumentation. Consequently, the available datasets are also targeted at these two directions. For argument mining, researchers tend to use structured texts, thus they often extract arguments from student essays. However, for argument clustering, student essays can only cover a small part of the task. The scientific focus on argument retrieval came closer at a later stage. Since argument mining methods were not yet mature at that time, researchers resorted - initially as an interim solution - to debate portals and used them as an underlying argument base. However, they have now become well established among researchers working on argument retrieval [20, 31, 39].

A significant drawback of the current use of debate portals is that researchers here often use entire posts as arguments, which are not always arguments in the sense of argumentation theory as they are noisy and sometimes very long, making it impossible to unambiguously cluster arguments by semantics.[2] While there is the option of assigning an argument to multiple clusters [13], we advocate clustering short argument units, such as the one presented by Trautmann et al. [37] who identified and divided argumentative text spans into pro and con text spans.

The second main imperfection of these datasets is the procedure of the similarity judgment of these arguments. Existing works mostly first form pairs of arguments for which either crowd-workers from Amazon Mechanical Turk are hired to assess similarity [27], or researchers do this themselves [10]. Both approaches are reasonable, especially because they involve several annotators and they average the scores at the end. Also, the robustness is usually measured with an inter-annotator agreement method such as Krippendorff's α [15]. However, since these are only robust but not perfect similarities, there is still some room for improvement (even if a perfect similarity can probably never be reached due to subjectivity), e.g. having a dataset where the similarity of the arguments was determined by journalists over a longer period of time.

In contrast to previous work dealing with argument clustering, we do not use a dataset that uses entire posts from debate portals as arguments, but a corpus consisting of high quality argument units, which was created by Dutch journalists in a visual fashion and translated by us into English and converted into a machine-readable format. Another dataset of interest is KIALO, which is of high quality but does not come with frame labels.[3]

3 ARGUMENT CLUSTERING PIPELINE

In this section, we define our two-stage pipeline for clustering arguments. The key idea, which is grounded in heuristics, is to split the set of arguments into disjoint subsets based on their *stances* and *frames* using two classifiers, and then cluster these subsets. The arguments in such disjoint subsets will therefore never be assigned to the same final cluster. This avoids a key problem of state-of-the-art methods for argument clustering by hierarchical clustering, which may assign dissimilar arguments in terms of stances or frames the same cluster. Figure 1 shows an overview of

[2] An example for a very long post on a debate portal can be seen at the following link: https://www.debate.org/debates/abortion-debate/2/.
[3] https://www.kialo.com/

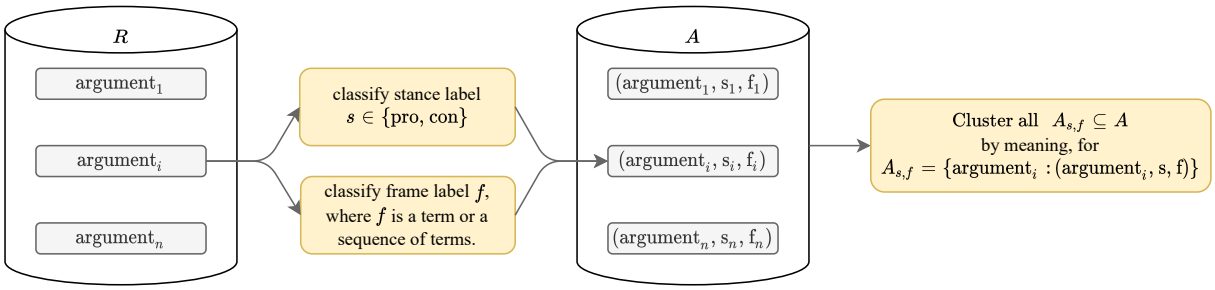

Figure 1: Pipeline for argument classification and clustering.

our pipeline. The input is a query q and a set of relevant argument units $R = \{\text{argument}_1, \text{argument}_2, \ldots, \text{argument}_n\}$ retrieved by an argument search engine. The first step consists of the two substeps *stance classification* and *frame classification*, which can be executed in parallel as they are independent from each other. For the *stance classification*, we determine the stance s_i in $\{pro, con\}$ of argument$_i$ to q. Note that we abstain from a neutral stance here because such statements are not argumentative by nature. The idea here is that if two arguments already have different stances, they ought never be in the same final cluster. For the *frame classification*, we determine the frame f_i from argument$_i$, which consists of a term or a sequence of terms. Although this would theoretically allow all possible sequences of words of any length for a frame, we do not impose a mandatory restriction here, but we do advocate that it should simply state the frame of an argument in one word if possible such as "*finance*" or "*health*", but also multiple terms such as "*job satisfaction*". Now we can form the tuple $(\text{argument}_i, s_i, f_i)$ for argument$_i$ and pass it into the set of argument units A to cluster for the query. We define a clustering of arguments by meaning as a set of similar premises that all have the same stance and frame. In this context, "similar" means that each of the premises mean the same. As an example, consider Figure 2 with the cluster label "*Social innovation keep care affordable*" and its associated specifications. Given all disjoint subsets $A_{s,f} \subseteq A$, where $A_{s,f} = \{\ \text{argument}_i : (\text{argument}_i, s, f)\ \}$ is the set of arguments with the stance s and the frame f as determined by the first step, we then cluster each subset $A_{s,f}$ independently to get our final clustering. For this purpose, we will incorporate agglomerative clustering (see Section 5).

4 DATASET

In this section, we present acquisition and processing of our novel dataset.

Drawbacks of Existing Datasets. Although argument clustering has been considered before, most researchers in argumentation, however, have been working with impure, hand-made, or incomplete datasets for a while. Frequently, such datasets include user-generated posts extracted from debate portals [27]. While they not only occasionally contain nonsense and insults, they also come with the disadvantage that very often they are not real arguments at all, i.e., do not correspond to arguments as defined in argumentation theory [10]. As it is undeniable that these posts contain valuable arguments, but are not in their pure form, we also advocate instead working with their argumentative units, which undoubtedly exist there. Otherwise, it can have a negative impact on the development of computational procedures, considering that machine learning methods, for example, may be sensitive to errors and non-argumentative statements. We argue that entire posts are rather suitable for pioneering work when one is still in the early stages and applying methods for the first time in order to see where the journey is going. However, such datasets might not be considered as the one absolute truth for learning methods.

Dataset of De Argumentenfabriek. This paper presents a dataset that meets the most stringent requirements. Instead of crowdworkers [27] or researchers in computer science [10], experts working in the field of journalism, spent weeks in constructing these argument graphs. The journalists' goal in creating the maps was to keep readers informed on the issues without requiring them to spend a lot of time re-reading past articles. Therefore, new arguments were transparently added to a map from time to time.

Our dataset consists of 365 unique argument graphs containing 11,266 arguments in English (and in Dutch), all of which were originally downloaded as PDFs from the Dutch website De Argumentenfabriek and manually converted to CSV files.[4] The graphs address various regional and national issues such as the "*environment*", "*old-age poverty*", or "*Brexit*". Figure 2 visualizes an except of such a graph. As we can infer from the figure, each map contains a query.[5] The left and the right side of the argument map visualize the stances. In this case, the left side contains the pro arguments while the right side contains the con arguments. Furthermore, each frame has its own branch, which originates from a stance. Additionally, the frames have branches to the individual final clusterings with their corresponding arguments. Often, the final clusterings have labels summarizing its arguments.

Processing of the Dataset. We initially downloaded 165 PDFs, from which we extracted 369 argument graphs. As mentioned before, the original text of these graphs was mainly written in Dutch. Consequently, after all graphs were extracted and saved to one large CSV file, we automatically translated all columns of the CSV into English using DeepL.[6] [7] We manually validated and, if necessary,

[4] The argument graphs were downloaded from the following website: https://www.argumentenfabriek.nl/nl/producten-kopen-of-gratis-downloaden/.
[5] This differs to the definition in Section 1 as an argument consists of a claim and a premise but is deployed in many works [10, 39].
[6] https://www.deepl.com/.
[7] We also found an argument map written in Portuguese and some maps in English language.

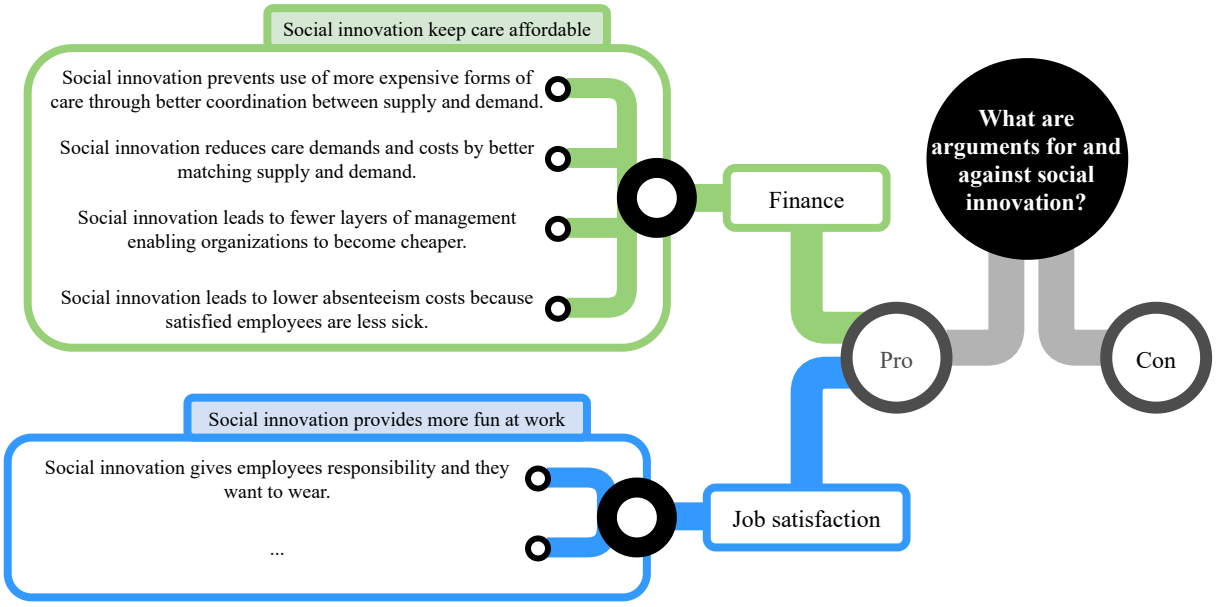

Figure 2: Visualization of an excerpt of an argument map.

corrected the translated texts with respect to their sense in the context of the query. In the process, we removed four (near-)duplicates to have unique graphs. We left out 77 maps because (1) they contained only graphs or tables as arguments, (2) they were by far too small, (3) they serve solely as overview maps, (4) the argument texts consisted only of bullet points, (5) there were several queries in one map, or (6) they were much too poorly structured. Note that not all graphs followed the stance-frame-cluster structure presented in Section 3. For example, in some graphs the stance was missing, in others there was no frame label or there were several of them. In the case of multiple frame labels, we clustered them in the form (frame_1; frame_2; ...; frame_n) to avoid omitting crucial information (e.g., for follow-up works).

Thus, our dataset consists of a total of 11,266 arguments spread over 365 queries. In total, we found 63 graphs that are divided into stances and 346 graphs that are divided into frames; 49 graphs come with both stances and frames. On average, there were 9.926 final clusters per graph, with a median of 27.

5 METHODS AND EVALUATION

In this section we present the methods used in the implementation of our pipeline as well as their evaluation. We first describe the classification of the argument units into their respective stances, and the determination of their frames. Then, we evaluate the isolated performance of these methods. After that, we show how these methods can be combined for clustering arguments. We then evaluate the end-to-end performance of the whole clustering pipeline.

5.1 Preprocessing of the Dataset

Since in this section we focus on frame detection in argumentative graphs, we only work with those 49 graphs with argument units divided into frames on the one hand and stances on the other hand. We determined the contextualized embeddings of the argument units, using the framework SBERT from Reimers and Gurevych [26]. For the concrete computation, we used the model `stsb-roberta-large` since it achieved the highest performance for the *Semantic Textual Similarity Benchmark* (STSB) task. An embedding in each layer contains 1,024 dimensions. Note that we work with the embedding in the last layer. To integrate the semantics of the query into the premise, as the premise can also depend on it (e.g. the premise's stance), we computed both the embedding of the premise (e_p) and the embedding of the query (e_q) and merged it pointwise via multiplication, addition and difference as follows:

$$(e_p, \quad e_q \cdot e_p, \quad e_q + e_p, \quad |e_q - e_p|) \tag{1}$$

Thus, each final embedding consists of $4{,}096 = 4 \cdot 1{,}024$ dimensions.

5.2 Predicting Stances

In order to determine a premise's stance to its query, we evaluated various classifiers that distinguish between pro and con stances. Since there are different notations for expressing stances, e.g., a positive stance in the dataset appears in the forms "*pro*", "*pros*", "*well*", "*for*", "*in front of*", etc., we manually unified them in a preceding step, resulting in only the two stances mentioned above. As there were only 17 unique stances in the whole database and those stances were unambiguous, the unification of these stances was easy to accomplish by hand. The distribution of the stances is relatively even (1,080 with stance pro and 1,106 with stance con).

For classification, we evaluated a total of five standard classifiers with their standard parameters and evaluated the predictions with leave-one-out cross validation, i.e., we have 49 folds.[8] Considering the fact that we have only 2,186 argument units here, it

[8]For the implementation we used the framework from scikit-learn [3], where we also took the default parameters from.

Table 3: Classifiers and their performance for predicting the argument units' stances.

classifier	accuracy (DeepL)	accuracy (Google Translate)
Support Vector Machine	**0.8632**	**0.8303**
Extreme Gradient Boosting	0.8495	0.817
Random Forest	0.8449	0.8129
Gaussian Naïve Bayes	0.8381	0.8124
Logistic Regression	0.8353	0.7777

Table 4: Classifiers and their performance for predicting the argument units' frames.

classifier	accuracy standard model	accuracy fine-tuned model
Logistic Regression	**0.6839**	**0.9602**
Extreme Gradient Boosting	0.6427	0.9515
Support Vector Machine	0.6327	0.7283
Random Forest	0.5933	0.9405
Gaussian Naïve Bayes	0.5119	0.5631

is not worthwhile to take 10 % (or five of 49 graphs) from this set as a development set to find suitable parameters. Table 3 shows the used classifiers and their performances in terms of accuracy. As we can observe from the table, classification with a Support Vector Machine performs best for predicting stances. Note that in a preliminary study we executed the translations of the graphs by Google Translate before calculating their embeddings. However, by utilizing DeepL we achieved a performance boost between 2.5 to 5.7 percentage points for the accuracy values, which is an indicative for the quality of the translations using DeepL. For the remainder of the paper we use the translations from DeepL.

5.3 Predicting Frames

Contrary to the stances, the unification of the frames is more complicated, because here we have much more frames, in total 133. Another and here even more significant factor for the difficulty in contrast to the stances are that frames can be differently similar or dissimilar to other frames. For example, the frame "*finance*" is more similar to "*economy*" than to "*health*".

5.3.1 Our Approach. Our approach aims to predict the general frames with classifiers and is very similar to the prediction of the stances above. The only difference is that here we do not use the 133 frames for the classifiers in their pure form. Instead, we manually group them by their similarity, obtaining 22 frames. This number is reasonable as, e.g., Naderi and Hirst [21] used a similar one, i.e., 16. Furthermore, online newspapers usually provide about 20 tags to find interesting articles. Examples of such newspapers are BBC News, The Independent, or Daily Mirror.[9] Still, the distribution of unified frames is clearly more unbalanced than for stances. While the most used frame deals with "*finance*" (577), the least used deals with "*persuasion*" (7). On average, each frame contains 99.4 arguments (median 57).

The left side of Table 4 shows the results for predicting the argument units' frames. As shown in the table, logistic regression performs best for frame prediction. However, the performance here is not as good as for predicting the stances owing to the class diversity. Due to this moderate performance in frame prediction, we fine-tuned the underlying model 49 times.[10] To this end, for each graph we randomly picked both argument pairs with the same frame label and argument pairs with different frame labels. For equal and unequal argument frame pairs, we extracted at least 100 and at most 200 pairs. In fact, the algorithm stopped the random drawing of pairs once both conditions were met. Pairs with the same frame and pairs from different frames were labeled with 1 and 0, respectively. We trained the model with only one epoch and a batch size of 64. For each of the 49 argument graphs, we trained a new model with the other 48 graphs and predicted the frame labels for the remaining one. The right side of Table 4 shows the massive performance boost gained after fine-tuning. Note that similar to predicting the stances, the performance of the frame prediction improved with the use of DeepL instead of Google Translate. These improvements range from 0.9 to 6.1 percentage points for the accuracy for the standard model but are not displayed due to space limitations. Note also, that applying DeepL reduced the number of unique frames in the concentrated dataset from 138 to 133.

5.3.2 Baselines to Cluster Frames. For comparing our framing approach with others, we employ two baseline methods here.

The first follows the idea of Reimers and Gurevych [26], who apply agglomerative clustering on contextualized word embeddings to obtain a frame-based clustering. We however implement a variant here, namely that of our previous work [10], which instead of making a constant tree cut, computes a dynamic one [16]. The latter detects clusters in a dendrogram depending on their shape and thus also finds nested clusters. This is helpful because otherwise we would have to manually specify the cut height to get the number of clusters, which we do not know in advance and may vary from dataset to dataset and thus would be difficult to generalize.

The second baseline originates from Ajjour et al. [1] and clusters arguments into frames in three steps. We choose this approach since in their experiments they show the benefit of finding generic frames after removing topical features from the arguments. As they compare several variants in their work, here we restrict ourselves only to their best reported performing parameters.[11] In the first step, they remove all topical features. For this, they map each argument into a vector space, and then cluster them with k-means [12] using Euclidean distance. For the mappings into the vector spaces they use Latent Semantic analysis (LSA) [8] with 1,000 dimensions. In their evaluation, they find that clustering by topics with LSA works better by adding more contextual information. Since they work with posts from debate portals, they achieve this by using debates instead of individual arguments. In our implementation, we also extend the arguments with additional context information in form of query, argument graph headline and background information for the above-mentioned reason. Furthermore, we use x-means [22], instead of k-means, because the variable k is highly dependent on

[9] https://www.bbc.com/, https://www.independent.co.uk/, https://www.mirror.co.uk/.
[10] https://sbert.net/docs/training/overview.html.
[11] We used the libraries provided by scikit learn [3] and https://pyclustering.github.io.

the selected dataset and we use a different one here. In our case, for each argument graph the number of dimensions is smaller than 1,000 (about 40), because the argument graphs contain less words than whole debates in debate portals which can be overwhelming. In the second step, they remove the topic-specific features from the formed topic clusters. For this, they compute TF-IDF [30] for each word in each cluster and remove all terms whose TF-IDF value is over an empirically chosen threshold δ. They calculate IDF using the following formula: $\text{IDF}(v) = |\bar{A}| \;/\; |\{\bar{A}_j \in \bar{A} : \text{for each word } v \in \bar{A}_j\}|$, where $\bar{A}$ is the set of arguments on the same topic. Finally, after removing the terms from step 2, they again cluster all arguments with k-means as in step 1. We again apply a slightly modified implementation by using x-means.

5.4 Determination of the Final Clusters

Now after having determined both the stances and the frames, we can already use them to form clusters by placing those argument units in the same clusters where both stance and frame are identical. We call this approach PLAIN. In addition, we can take this further and run a final clustering procedure on these new disjoint subsets, as illustrated in the pipeline in Figure 1. Therefore, we apply hierarchical clustering in accordance with the state-of-the-art. We follow the procedure of our prior work [10] and use agglomerative clustering with Euclidean distance, the average linkage method, as well as a dynamic tree cut [16] for these subsets. We call this approach SEQUENTIAL. We call the third and last approach MERGE where we concatenate the cluster ids of the approaches PLAIN and the agglomerative (henceforth called: HIERARCHIC) clustering conducted on all arguments of a graph.

5.5 Evaluation Setup

We derive the *ground truth clustering* for the perfect labels from the dataset, since the correct labels for stance, frame and final clustering are already provided in the maps. We intend to evaluate two important aspects regarding the clustering: First, we measure the performance of the clusterings for the respective levels, i.e., stance and frame. Second, we evaluate the overall clustering given that (as shown in Figure 1) the output of the preceding layer is used as input for the current layer.

As a *baseline*, we use hierarchical (agglomerative) clustering of the contextualized embeddings of the argument units according to the state-of-the-art for clustering arguments. Please note that the arguments in our dataset do not contain semantic duplicates of arguments, yet this baseline is also used for that [10]. For computation of the agglomerative clustering, we use the implementation of our previous work [10]. This method was already introduced earlier. In addition to this baseline, we also added two simple comparison clusterings by (1) putting all argument units in the same cluster ($SIMPLE^{\,n\ units}_{1\ cluster}$), and (2) each argument unit in its own cluster ($SIMPLE^{\,n\ units}_{n\ clusters}$). For the rest, we implemented almost all of the methods described in the previous section except that we used the variant SEQUENTIAL only for the combinations that performed best, due to the fact that agglomerative clustering is time consuming. When computing the predictions for each graph, we always removed that graph from the corresponding training set.

5.6 Evaluation Measures

We evaluated the clusterings using external cluster evaluation methods. In contrast to internal cluster evaluation measures, which require only the information of the vectors of the dataset, the external evaluation measures are based on prior knowledge [28]. In this case, the prior knowledge comprises the labels provided by the ground truth. Since different cluster evaluation measures emphasize disparate issues, we compare our clusterings by using two measures, i.e., the *Purity* and the *Adjusted Rand Index* (Rand). We chose the adjusted variant for Rand here because the amount of arguments per graph is fairly small, and we want to eliminate randomness as a factor here to avoid producing true clusters by chance. We briefly explain the main features of these measures: Purity ranges from 0 to 1 and measures the extent to which clusters contain a single class. The value 1 implies a perfect clustering. The drawback here is that a high number of clusters result in a high purity. Actually, we get purity 1 simply by putting each element in its own cluster. The adjusted Rand index ranges from 0 to 1 and calculates the accuracy of the generated clusterings in comparison to the ground truth. Hereby, it penalizes false positives as well as false negatives with equal weights. The term adjusted implies that the values are adjusted for chance. In the context of clustering, we propose to favor the product of these two measures, i.e., prod(P,R) as we want to obtain a good balance between many pure clusters (following the Purity) and few impure clusters (following the Rand index).

5.7 Evaluation Results

Tables 5, 6, 7 display the measured performance in terms of external cluster evaluation methods after classifying (or clustering) by stance, frames, or meaning, respectively. All tables are sorted by prod(P,R) in descending order.

Evaluation of Clustering by Stance. Since the classifier Support Vector Machine (SVM) performs best for predicting stances (see Table 3) and also achieves a better clustering (see Table 5), we only show this classifier in comparison to the baseline and the comparison values. Anyway, the other classifiers' performances were worse. We can observe from the table that a SVM for stance prediction (henceforth: $\text{SVM}^{\text{stance}}$) achieves the best performance here. The hierarchical clustering is obviously and as expected not suitable for this task. Just as one might have guessed, even perfect clustering by frame is of no use if we want to cluster by stance. In the following we will see that it is the same the other way around. This supports the thesis that it makes sense to cluster by stance and by frames because they do not get in each other's way. In order to incorporate correction for multiple tests, we performed Tukey's HSD (honestly significant difference) tests [29] with $p = .05$ on the purity and adjusted Rand values on the 49 folds showing significant differences between $\text{SVM}^{\text{stance}}$ and the other methods in Table 5. Hence, $\text{SVM}^{\text{stance}}$ for stance prediction especially outperforms the baseline comprising hierarchical clustering but is still significantly worse than the ground truth. The observed difference between $\text{SVM}^{\text{stance}}$ and the other classifiers shown in Table 3 is not significant.

Evaluation of Clustering by Frame. Table 6 shows the clustering performance when we take gold standard frames as ground truth. For a better overview, we only show the best performing

Table 5: Table showing performance for clustering when stance is the only ground truth. The highest values unequal to the comparison values are marked in bold.

method	mean Purity	mean Rand	prod(P,R)
SVM^{stance}	**0.8542**	**0.524**	**0.4476**
HIERARCHIC	0.7364	0.0343	0.0253
ground $truth_{frame}$	0.6033	0.0	0.0
SIMPLE $^{n\ units}_{1\ cluster}$	0.5397	0.0	0.0
SIMPLE $^{n\ units}_{n\ clusters}$	1.0	0.0	0.0

variant for the baseline LSA^{frame} here. It should be no surprise that the classifiers trained on the fine-tuned dataset (NB^{frame}_{tuned} and LR^{frame}_{tuned}) perform better than those of the standard model (NB^{frame} and LR^{frame}). However, the surprise is that NB^{frame}_{tuned}, which was noticeably worse at correctly assigning frame labels to the classifiers (see Table 4), now even outperforms LR^{frame}_{tuned}. After examining the predicted labels more closely for both NB^{frame}_{tuned} and LR^{frame}_{tuned}, we can report that although NB^{frame}_{tuned} often predicted the presumed "wrong" label, it was able to assign the same frame labels more often to arguments that have the same ground truth labels (see Section 5.8). Therefore, it might be primarily important that argument units with the same frame are placed in the same cluster, no matter whether the correct frame was predicted; predicting frame labels is a task for future work. Considering the classifiers which run on the embeddings of the standard model, we observed that the performance agrees with Table 4. Furthermore, we can infer from the table that the two baselines, HIERARCHIC and LSA^{frame} perform worst here. Presumably, the dynamic tree cut is not well suited for cluster determination after agglomerative clustering. LSA^{frame} probably performs poorly because we have fewer terms in the graphs here than in the debate portals that were used in the original work [1].

As we did with the stances, here we also conducted Tukey's HSD tests with $p = .05$ on the purity and adjusted Rand values of the 49 folds. The observed difference after applying the fine-tuned models is significant for both measures. We did not find significant differences between LR^{frame}, LSA^{frame}, and HIERARCHIC for the Rand index, but we did for Purity. Moreover, we observed a significant difference between NB^{frame}_{tuned} and LR^{frame}_{tuned} for both measures.

Evaluation of Final Clustering. Table 7 shows the performance for the final clusterings. In the following, a triple (S, F, C) implies that the methods S, F, and C were executed sequentially, where S is the stance prediction and F is the frame prediction. C describes the final clustering, which is skipped in the case of SKIP. Again, for the sake of clarity, we chose the best combinations of the methods performing out of Tables 5 and 6. Anyway, the values of the other methods were not better than the ones shown here. As we can see in the table, we defeat the baseline by classifying the stances of the argument units with SVM^{stance} beforehand and then classifying the frames with NB^{frame}_{tuned}. Users interested in more pure clusters can combine the cluster ids with MERGE as this method retrieves the intersection of the clusters. While the combinations using the standard model without fine tuning as well as the baseline LSA^{frame} perform poorer,

Table 6: Table showing performance for clustering when frame is the only ground truth. The highest values unequal to the comparison values are marked in bold.

method	mean Purity	mean Rand	prod(P,R)
NB^{frame}_{tuned}	**0.849**	**0.6982**	**0.5928**
LR^{frame}_{tuned}	0.7833	0.5653	0.4428
LR^{frame}	0.5175	0.0974	0.0504
NB^{frame}	0.4577	0.062	0.0284
HIERARCHIC	0.5394	0.0317	0.0171
LSA^{frame}	0.3905	0.0244	0.0095
ground $truth_{stance}$	0.3773	0.0	0.0
SIMPLE $^{n\ units}_{1\ cluster}$	0.3519	0.0	0.0
SIMPLE $^{n\ units}_{n\ clusters}$	1.0	0.0	0.0

all combinations using SEQUENTIAL perform worst in terms of the Rand index. This might be due to the graphs having relatively small sizes, but may work better for larger graphs. As with stance and frame, here we also performed Tukey's HSD tests. For both Purity and Rand, the observed differences between the methods (SVM^{stance}, NB^{frame}_{tuned}, SKIP) and (SVM^{stance}, NB^{frame}_{tuned}, MERGE) to the others were significant. Furthermore, we observed significant differences between these two methods for both measures.

5.8 Evaluation on Another Dataset

In order to evaluate the performance of our methods in a more generalized way, we applied them to another dataset, that is, AURC-8 [37]. This dataset consists of eight topics each with 1,000 user-generated sentences originating from the Common Crawl archive.[12] A key difference to our dataset, besides presumably different appearing frames as well as topics being used as queries, is the larger size of AURC-8. Some of the sentences are divided into several argument units, others are not argumentative at all. Overall, there are 4,967 argument units. Among others, the topics include abortion, net neutrality, or gun laws. With this dataset, we evaluate stance prediction, framing, and final clustering. In all experiments, we shuffled the data and disguised each information that could bias the annotator's decision (such as the used classifier names).

Evaluation of the Stance Predictions. To evaluate the stance predictions, we trained the SVM classifier on our dataset and then generated the predictions for AURC-8 (after calculating the embeddings applying Equation 1). Since AURC-8 already comes with gold labels for stances, we could easily determine the accuracy, which is 0.7244. The classifier predicted 2,241 times a pro stance and 2,728 times a con stance. However, since stance prediction is not the focus of this paper, we did not make any further effort to boost this value.

Evaluation of the Frame Predictions. As we already saw in Table 6 actually only the classifiers for frame prediction, once trained on the fine-tuned embeddings, perform well. Thus, here we consider only those classifiers trained on the embeddings of the fine-tuned model on the initial dataset. To be precise, we evaluate here only

[12] http://commoncrawl.org/2016/02/february-2016-crawl-archive-now-available/.

Table 7: Table showing performance for clustering when the final clustering is the only ground truth. The highest values unequal to the comparison values are marked in bold.

method	mean Purity	mean Rand	prod(P,R)
(SVM^{stance}, NB^{frame}_{tuned}, SKIP)	0.5424	**0.3293**	**0.1786**
(SVM^{stance}, NB^{frame}_{tuned}, MERGE)	**0.7848**	0.1732	0.1359
(SVM^{stance}, LR^{frame}, MERGE)	0.7033	0.0805	0.0566
(SVM^{stance}, LSA^{frame}, MERGE)	0.5804	0.0768	0.0446
(SVM^{stance}, LR^{frame}, SKIP)	0.4066	0.1011	0.0411
(SVM^{stance}, LSA^{frame}, SKIP)	0.2624	0.0711	0.0187
HIERARCHIC	0.3525	0.046	0.0162
(SVM^{stance}, NB^{frame}_{tuned}, SEQUENTIAL)	0.997	0.0006	0.0006
ground $truth_{frame}$	0.4102	0.3184	0.1306
ground $truth_{stance}$	0.2403	0.1283	0.0308
SIMPLE $^{n\ units}_{n\ clusters}$	1.0	0.0	0.0
SIMPLE $^{n\ units}_{1\ cluster}$	0.1327	0.0	0.0

the frame predictions of NB^{frame}_{tuned} and LR^{frame}_{tuned}, because the latter performed better in predicting the correct frame labels (see Table 4), whereas the former detected more correct frame clusters, even though the latter assigned them to different frame names. Besides, we disregard LSA^{frame} and HIERARCHIC as they do not provide frame labels which we could evaluate here. Moreover, LSA^{frame} needs to specify several parameters and thresholds.

We evaluate two aspects: (i) we investigate which of the two classifiers can find more correct frame labels, and (ii) which classifier is better at clustering into frames of arguments. Note that similar to the ground truth, the predictions are relatively unbalanced for both classifiers, while the distributions are similar to each other and to the ground truth to some extent. In terms of (i), we randomly pick ten premises for each of the eight topics for both classifiers ($160 = 10 \cdot 8 \cdot 2$). Then, an annotator determined scores on a scale from 1 to 3 whether the output frame labels are absolutely reasonable (score 3), tenable to an extent (score 2), or not reasonable at all (score 1). We purposely chose not to let the annotator assign labels here because we do not want to penalize the classifiers if they assign labels to the premises that are reasonable but different to the ground truth. For example, for the argument unit "*the death penalty is a deterrent to crime*" one classifier predicted the frame "*safety*" while the other predicted "*effectiveness*" which are both acceptable. The mean average scores for NB^{frame}_{tuned} and LR^{frame}_{tuned} are 2.275 and 2.15, respectively. Thus, both classifiers predict tenable frame labels. Similar to our dataset, NB^{frame}_{tuned} performs slightly better. In terms of (ii), for each of the eight topics we draw 50 premise pairs and let an annotator decide for each of the 400 pairs on the same scale from 1 to 3 whether the two premises could be placed in the same frame cluster or not. After that, we computed the accuracy values for the two classifiers. Considering premise pairs as similar with a minimum score of 2, the accuracy for NB^{frame}_{tuned} and LR^{frame}_{tuned} is 0.648 and 0.593, respectively. With a minimum score of 3, the accuracy values yield 0.62 and 0.6. Once again, the former performs better.

Evaluation of the Clustering Predictions. We employed the following approach to evaluate whether our findings in the initial dataset corresponds to this one: For each of the eight topics, we computed the clusterings of the four methods (SVM^{stance}, NB^{frame}_{tuned}, SKIP), (SVM^{stance}, NB^{frame}_{tuned}, MERGE), (SVM^{stance}, LR^{frame}_{tuned}, MERGE), and HIERARCHIC (see Table 7). Existing works often evaluate argument clusters by assessing the similarity of pairs of arguments, for example, on a scale of 1 to 3. Given the interpretive freedoms, we take a more robust approach here: we choose two arguments from each cluster and also add a randomly picked one from another cluster (with the same topic). The annotator now has to decide which of the three arguments is least fitting to the cluster. For each topic and method, we picked 10 pairs from the clusterings and asked an annotator to spot the intruding premise. Thus, we have a total of 314 triples to be evaluated.[13] The methods (SVM^{stance}, NB^{frame}_{tuned}, SKIP), (SVM^{stance}, NB^{frame}_{tuned}, MERGE), (SVM^{stance}, LR^{frame}_{tuned}, MERGE), and HIERARCHIC yielded mean average precision values of 0.663, 0.638, 0.663, and 0.663, respectively. The mean average cluster sizes are 16.5, 310.88, 367.13, and 115.75. Overall, we can conclude that the final clustering step yields a more fine granular clustering, yet it does not seem to improve the state-of-the-art applying agglomerative clustering at least in this dataset. Most likely, this is due to the vast amount of premises to be clustered here, therefore requiring more intermediate steps.

6 CONCLUSION AND FUTURE WORK

Clustering arguments to help users identifying the best arguments quickly is an important yet difficult task in argument retrieval. In this paper, we demonstrated that the clustering of arguments can be enhanced by dividing them by their stances as well as their frames before performing a final clustering on them. We make our new dataset consisting of high quality argument graphs published in newspapers for classifying by stances, frames and final clusters available to the computational argumentation community.

In future work, we plan to extend the pipeline to include argument schemes such as those of Walton et al. [42]. We plan also to cluster argument units by their factual validity as well as their argument quality dimensions [38], i.e., we then address whether the units are logically conclusive or just evoke emotions. Since we now have a gold standard for high quality arguments and their clustering, we may use this to predict the cluster labels. For example, the transformer model T5 [25] is suitable for this purpose as it allows fine-tuning. As each argument graph consists of a query and arguments along with labels for stance, frame, and cluster, we can use it primarily as an argument base for IR systems, where we could, for example, use these queries to train retrieval for arguments.

ACKNOWLEDGMENTS

We would like to thank Damian Trilling from the University of Amsterdam for his input during a workshop where he introduced us to the dataset this paper is based on.

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the projects ReCAP and ReCAP-II, Grant Number 375342983 - 2018-2024, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

[13] The method (SVM^{stance}, NB^{frame}_{tuned}, SKIP) delivered for the topic minimum wage only four instead of ten triples to evaluate since this classifier often predicted the frame "finance" for these premises.

REFERENCES

[1] Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling Frames in Argumentation. 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 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 2922–2932. https://doi.org/10.18653/v1/D19-1290

[2] Filip Boltuzic and Jan Snajder. 2015. Identifying Prominent Arguments in Online Debates Using Semantic Textual Similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining, ArgMining@HLT-NAACL 2015, June 4, 2015, Denver, Colorado, USA*. The Association for Computational Linguistics, 110–115. https://doi.org/10.3115/v1/w15-0514

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. *CoRR* abs/1309.0238 (2013). arXiv:1309.0238 http://arxiv.org/abs/1309.0238

[4] Elena Cabrio and Serena Villata. 2018. Five Years of Argument Mining: a Data-driven Analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 5427–5433. https://doi.org/10.24963/ijcai.2018/766

[5] Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The Media Frames Corpus: Annotations of Frames Across Issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*. The Association for Computer Linguistics, 438–444. https://doi.org/10.3115/v1/p15-2072

[6] Dennis Chong and James N Druckman. 2007. Framing theory. *Annu. Rev. Polit. Sci.* 10 (2007), 103–126.

[7] Claes H De Vreese. 2005. News framing: Theory and typology. *Information design journal & document design* 13, 1 (2005).

[8] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[10] Lorik Dumani and Ralf Schenkel. 2020. Quality-Aware Ranking of Arguments. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 335–344. https://doi.org/10.1145/3340531.3411960

[11] Robert M Entman. 1993. Framing: Toward clarification of a fractured paradigm. *Journal of communication* 43, 4 (1993), 51–58.

[12] John A Hartigan and Manchek A Wong. 1979. A K-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 100–108.

[13] Ian Jeantet, Zoltán Miklós, and David Gross-Amblard. 2020. Overlapping Hierarchical Clustering (OHC). In *Advances in Intelligent Data Analysis XVIII - 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27-29, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12080)*, Michael R. Berthold, Ad Feelders, and Georg Krempl (Eds.). Springer, 261–273. https://doi.org/10.1007/978-3-030-44584-3_21

[14] Kristen Johnson, Di Jin, and Dan Goldwasser. 2017. Modeling of Political Discourse Framing on Twitter. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*. AAAI Press, 556–559. https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15671

[15] Klaus Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data.

[16] Peter Langfelder, Bin Zhang, and Steve Horvath. 2009. Dynamic Tree Cut: In-depth description, tests and applications. https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/BranchCutting/Supplement.pdf

[17] John Lawrence and Chris Reed. 2019. Argument Mining: A Survey. *Comput. Linguistics* 45, 4 (2019), 765–818. https://doi.org/10.1162/coli_a_00364

[18] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

[19] Amita Misra, Pranav Anand, Jean E. Fox Tree, and Marilyn A. Walker. 2015. Using Summarization to Discover Argument Facets in Online Idealogical Dialog. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar (Eds.). The Association for Computational Linguistics, 430–440. https://doi.org/10.3115/v1/n15-1046

[20] Amita Misra, Brian Ecker, and Marilyn A. Walker. 2016. Measuring the Similarity of Sentential Arguments in Dialogue. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*. The Association for Computer Linguistics, 276–287. https://doi.org/10.18653/v1/w16-3636

[21] Nona Naderi and Graeme Hirst. 2017. Classifying Frames at the Sentence Level in News Articles. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, Ruslan Mitkov and Galia Angelova (Eds.). INCOMA Ltd., 536–542. https://doi.org/10.26615/978-954-452-049-6_070

[22] Dan Pelleg and Andrew W. Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, Pat Langley (Ed.). Morgan Kaufmann, 727–734.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162

[24] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2227–2237. https://www.aclweb.org/anthology/N18-1202/

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[26] 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 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[27] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 567–578. https://doi.org/10.18653/v1/p19-1054

[28] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi, and Elvia M Quiroz. 2011. Internal versus external cluster validation indexes. *International Journal of computers and communications* 5, 1 (2011), 27–34.

[29] Tetsuya Sakai. 2018. *Laboratory Experiments in Information Retrieval - Sample Sizes, Effect Sizes, and Statistical Power*. The Information Retrieval Series, Vol. 40. Springer. https://doi.org/10.1007/978-981-13-1199-4

[30] Gerard Salton, A. Wong, and Chung-Shu Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18, 11 (1975), 613–620. https://doi.org/10.1145/361219.361220

[31] Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it Blend? Blending Weak and Strong Labeled Data in a Neural Network for Argumentation Mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 599–605. https://doi.org/10.18653/v1/P18-2095

[32] Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching for Arguments in Heterogeneous Sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 2-4, 2018,*

Demonstrations, Yang Liu, Tim Paek, and Manasi S. Patwardhan (Eds.). Association for Computational Linguistics, 21–25. https://doi.org/10.18653/v1/n18-5005

[33] Christian Stab and Iryna Gurevych. 2014. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 46–56. https://www.aclweb.org/anthology/D14-1006/

[34] Manfred Stede, Stergos D. Afantenos, Andreas Peldszus, Nicholas Asher, and Jérémy Perret. 2016. Parallel Discourse Annotations on a Corpus of Short Texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. http://www.lrec-conf.org/proceedings/lrec2016/summaries/477.html

[35] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 809–819. https://doi.org/10.18653/v1/n18-1074

[36] Dietrich Trautmann. 2020. Aspect-Based Argument Mining. In *Proceedings of the 7th Workshop on Argument Mining*. Association for Computational Linguistics, Online, 41–52. https://www.aclweb.org/anthology/2020.argmining-1.5

[37] Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. Fine-Grained Argument Unit Recognition and Classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 9048–9056. https://aaai.org/ojs/index.php/AAAI/article/view/6438

[38] Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. Computational Argumentation Quality Assessment in Natural Language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, Mirella Lapata, Phil Blunsom, and Alexander Koller (Eds.). Association for Computational Linguistics, 176–187. https://doi.org/10.18653/v1/e17-1017

[39] Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an Argument Search Engine for the Web. In *Proceedings of the 4th Workshop on Argument Mining, ArgMining@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, Ivan Habernal, Iryna Gurevych, Kevin D. Ashley, Claire Cardie, Nancy Green, Diane J. Litman, Georgios Petasis, Chris Reed, Noam Slonim, and Vern R. Walker (Eds.). Association for Computational Linguistics, 49–59. https://doi.org/10.18653/v1/w17-5106

[40] Henning Wachsmuth and Till Werner. 2020. Intrinsic Quality Assessment of Arguments. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 6739–6745. https://doi.org/10.18653/v1/2020.coling-main.592

[41] Marilyn A. Walker, Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A Corpus for Research on Deliberation and Debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), 812–817. http://www.lrec-conf.org/proceedings/lrec2012/summaries/1078.html

[42] Douglas Walton, Chris Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.