Argument Mining Driven Analysis of Peer-Reviews

Michael Fromm,[1] **Evgeniy Faerman,**[1] **Max Berrendorf,**[1] **Siddharth Bhargava,**[2] **Ruoxia Qi,**[2] **Yao Zhang,**[2] **Lukas Dennert,**[2] **Sophia Selle,**[2] **Yang Mao**[2] and **Thomas Seidl**[1]

[1] Database Systems and Data Mining, LMU Munich, Germany
[2] LMU Munich, Germany
{fromm, faerman, berrendorf}@dbs.ifi.lmu.de

Abstract

Peer reviewing is a central process in modern research and essential for ensuring high quality and reliability of published work. At the same time, it is a time-consuming process and increasing interest in emerging fields often results in a high review workload, especially for senior researchers in this area. How to cope with this problem is an open question and it is vividly discussed across all major conferences. In this work, we propose an Argument Mining based approach for the assistance of editors, meta-reviewers, and reviewers. We demonstrate that the decision process in the field of scientific publications is driven by arguments and automatic argument identification is helpful in various use-cases. One of our findings is that arguments used in the peer-review process differ from arguments in other domains making the transfer of pre-trained models difficult. Therefore, we provide the community with a new peer-review dataset from different computer science conferences with annotated arguments. In our extensive empirical evaluation, we show that Argument Mining can be used to efficiently extract the most relevant parts from reviews, which are paramount for the publication decision. The process remains interpretable since the extracted arguments can be highlighted in a review without detaching them from their context.

Introduction

Argumentation is a process of bringing together and organizing reasons to convince a reasonable critic to accept or refuse a certain standpoint (Van Eemeren, Grootendorst, and van Eemeren 2004). It is an essential part of each rational decision-making process and after the decision is made, argumentation is important for its explanation and justification (Amgoud and Prade 2009). An important step in the argumentation process is the identification of arguments. Generally speaking, there is a difference between *argumentative* and *informative* content: Argumentative content expresses evidence or reasoning used to either oppose or support a given point. Informative parts often contain background information and describe how entities appear and act in the world.

In the last years, *Argument Mining* (AM) approaches have been applied in many fields and for different types of texts, such as encyclopedic articles (Aharoni et al. 2014), student essays (Stab and Gurevych 2014b), web discourse (Habernal and Gurevych 2016) or political speeches (Haddadan, Cabrio, and Villata 2019). AM techniques build the backbone of an IBM AI system *Project Debater*, which has the ambitious goal to debate humans on complex topics. This work aims to further extend the application of AM to the novel domain of scientific *peer reviews*. Peer reviewing is a cornerstone of today's academic editorial decision-making process in nearly all scientific disciplines. The peer-reviewers, who are usually not part of the editorial team, are experts in the corresponding research field and their task is the critical evaluation of the work proposed for publication. We argue that peer-reviewing can also be seen as an argumentation process, where the reviewers make up their minds about the examined publications and try to convince the editorial team by providing arguments in favor of or against acceptance. While the evaluation or review usually comprises different parts, such as a summary of the work or additional background information about the topic, the reviewers' pro and contra arguments are often the most relevant for making the final decision. Consequently, we envision that the automatic identification of argumentative content can improve and simplify different peer-review process phases. One possible use-case is to provide editors or meta-reviewers, co-responsible for the final decision, with an overview of arguments from all reviews and let them focus on the most relevant ones. For instance, after reading only the highlighted arguments in Figure 1, it is possible to get a good idea about the paper's strong and weak points. Another possible use-case is to support the reviewers by providing information about (missing) argumentation. For example, the author of the review in Figure 1 provides a detailed description of the empirical evaluation, but it is not completely clear from the text whether the reviewer is satisfied with the proposed evaluation criteria.

In this paper, we propose the application of AM to the domain of peer-reviewing. To this end, we collect a new dataset containing peer-reviews from different computer science conferences. We define a suitable AM annotation schema and annotate the dataset. We investigate the applicability of state-of-the-art AM techniques in an extensive empirical evaluation. Among others, we study the transferability of models trained on data from different domains to our task

Example Review

Summary: As the title suggests the paper focusses mostly on a negative result: Mutual information (MI) estimators obtained by variational methods have severe limitations that make them potentially not useful for down stream tasks. Besides highlighting the problems with variational MI estimators the authors suggest a modification to slightly improve the performance of MI estimators based on partition functions by reducing their variance when MI is high. **The authors give a good overview / introduction of various approaches to variational MI estimation by discriminative and generative methods.** Generally, MI estimation involves the estimation of the KL divergence between the joint distribution and the product of the marginals. The authors present a unifying view on the different approaches that optimizes the log density ratio required for the KL divergence over the space of log density ratios. Discriminative approaches model the density ratio directly (through e.g. neural network models) and generative approaches model the separate densities (as generative models where it is possible to evaluate the (conditional) probabilities / likelihoods of the data generating process). The authors prove that discriminative approaches that are based on the partition function approach suffer from high variance where mutual information is high (Theorem 2). The estimator based on a finite sample has high variance even if the density ratio approximation is correct. (The partition function approach is a way of staying constrained to the log density ratio function space.) This high variance problem is something that has previously been observed empirically and is the main theoretical point that is being made about limitations of MI estimators. In order to slightly alleviate the problem of high variance the authors suggest a way of biasing MI estimators by clipping the density ratio estimates through a constant chosen as a hyper-parameter. They prove that their clipping approach reduces variance and therefore introduces a bias variance tradeoff. In their later experiments the clipped version of the discriminative approach performs much better in terms of variance than without clipping and also better than a generative approach. In order to empirically evaluate the quality of MI estimators the authors suggest three criteria that they call self-consistency: (i) independence, (ii) data processing, (iii) additivity Self-consistency is evaluated experimentally on images where mutual information is computed between original image and image with part covered. The authors claim and experimentally show that discriminative approaches fail in (iii) and generative approaches fail in (i), (ii). Overall, variational MI approaches do not satisfy self-consistency. Evaluation: I suggest to accept the paper. **The theoretical contribution of showing the variance limitation of discriminative approaches seems significant.** That insight leads to the idea that clipping can be a useful bias that significantly reduces variance without making the already biased anyways results much worst in the experiments. However, I also feel like - *the paper is not yet as focused as it could be. It contains many concepts that could need a little bit more space.*

- Suggestions:

- Page 2: Nitpick, but in the definition of pseudo-formula using pseudo-formula twice is not super readable on the first read

- Page 2: In the definition of pseudo-formula clearify whether pseudo-formula is a marginal or a joint density (as pseudo-formula is the cumulative joint)

Figure 1: Example review for an ICLR'20 submission with labeling: Arguments in favor of acceptance are shown in green; red denotes arguments against it.

and the generalization across different conferences. Furthermore, we empirically validate our assumption about the importance of arguments for the decision-making process in academic publishing.

Related Work

Argument Mining

Argument Mining (AM) is the task of recognizing argument components (Palau and Moens 2009; Habernal and Gurevych 2016; Stab and Gurevych 2017; Hua and Wang 2017; Nguyen and Litman 2015) and their relations (Stab and Gurevych 2017; Nguyen and Litman 2016). The basis of AM are argumentation schemes that define the structure of the argument components and the relations between them. There is no universally accepted theory of argumentation (Van Eemeren, Grootendorst, and Kruiger 2019), and over time, argumentation schemes of varying complexity have been suggested in the literature (Toulmin 1958; Walton 2012; Freeman 2011; Stab and Gurevych 2014b). The original model by Toulmin (1958) comprises *claims* as an assertion for general acceptance, *data* (also often called premises) as the source of evidence to establish the claim, a *warrant* to justify the inference from a premise to a claim, *backing* (facts behind the warrant), a *qualifier* (degree of certainty for the inference) and *rebuttals*. The model has often been adopted in literature and most of the time, only premises and claims are used as argument components. However, it was observed that arguments in many text types have a more straightforward structure, e.g., models trained on a single dataset to identify *claims* do not generalize well to other document types (Daxenberger et al. 2017). Furthermore, annotating a dataset crawled from heterogeneous text sources leads to a low agreement among annotators (Habernal and Gurevych 2016; Miller, Sukhareva, and Gurevych 2019). Also, specific argument components (backing, warrant) appearing in the Toulmin-Scheme (Toulmin 1958) are often stated implicitly (van Eemeren et al. 2003; Habernal and Gurevych 2016). An argumentative scheme recently proposed by Stab, Miller, and Gurevych (2018) omits these components and simply distinguishes between (*supporting/opposing*) arguments and non-argumentative text parts. Its reasonableness is confirmed on the one hand by relatively high agreement among reviewers, and on the other hand by the model performance on texts from heterogeneous sources, see e.g. (Fromm, Faerman, and Seidl 2019). Furthermore, it was observed that the distinction between *supporting* and *opposing* arguments is more challenging than the distinction between argumentative and non-argumentative parts (Trautmann et al. 2020b,a; Fromm, Faerman, and Seidl 2019).

The development of models for the identification of argument components according to an argumentative scheme is similar to other NLP disciplines. Previous approaches rely on feature engineering (Habernal and Gurevych 2016; Lawrence and Reed 2015; Stab and Gurevych 2014a), more recent methods apply neural networks models. Guggilla, Miller, and Gurevych (2016) were the first to apply recurrent neural networks for AM. The state-of-the-art performance

in AM is achieved with pre-trained transformer-based architectures (Fromm, Faerman, and Seidl 2019; Trautmann et al. 2020a; Reimers et al. 2019).

A popular real-life application of AM techniques are argument search engines such as *argumenText*[1] (Stab et al. 2018) and *args*[2] (Wachsmuth et al. 2017) which allow argument retrieval according to a user-defined topic. AM is applied in the preprocessing step, where arguments are extracted from documents before they are indexed by a search engine.

Application of NLP for Peer-reviewing Process

So far, AM for scientific peer-reviews has received little attention. Hua et al. (2019) introduce a dataset with propositions in scientific reviews. The annotation schema is comprised of components that often appear in reviews such as *requests, facts, evaluations* or *quotes*. The dataset is annotated on a sentence level and the main focus is to study the usage of different propositions across venues. In our application, we are interested in arguments directly affecting the decision process, and therefore, the stance of the argument bears essential information. Since this information is missing in Hua et al. (2019), this annotation schema is not suitable for our application. Closely related is Xiao et al. (2020) work, where the goal is to automatically detect the problem description in peer-reviews. However, although the problems can also be considered opposing arguments, it is crucial to consider both positive and negative arguments for our application.

Other related works deal with different aspects of the peer-reviewing process. In Plank and van Dalen (2019), the authors introduce a dataset with scientific reviews and analyze it based on the title, abstract, and review text on how well the citation impact of a paper can be predicted. Gao et al. (2019) study the effect of author replies in the rebuttal phase. *Argumentative zoning* (Teufel, Siddharthan, and Batchelor 2009) analyzes the rhetorical and argumentative structure of scientific papers with intending to convince reviewers that the knowledge claim of the paper is valid.

Dataset

We use the OpenReview[3] platform and the OpenReview-Crawler[4] to retrieve peer-reviews. We collect all reviews from six computer-science conferences listed in Table 1. The annotated dataset [5] and the code [6] is available.

There, we additionally provide basic statistics about conferences and collected reviews.

Preprocessing

In a first preprocessing step, we replace URLs, escape sequences, encapsulated mathematical formulas, Unicode symbols and markdown with a corresponding type placeholder token respectively, e.g. `<URL>` for URLs. Furthermore, we remove multiple consecutive whitespaces and split review texts into sentences using the `PunktSentenceTokenizer` from NLTK.[7] To further improve the sentence splitting results, we provide the tokenizer with a set of idioms and abbreviations commonly used in scientific texts to avoid sentence splitting in the middle or after them.[8] Finally, we remove all sentences with less than three tokens and go through the dataset manually and remove non-interpretable sentences.

From 12,135 collected reviews, we sample 77 for the annotation. To this end, we first sample a conference uniformly at random and then a review from the conference.[9] We use stratified sampling to ensure that sampled reviews reflect the following three characteristics of original review distribution for each conference: Review-Rating (1-4), Paper-Decision (acceptance / rejection), and Review-Length.

Annotation

Scheme We use a simple argumentation scheme proposed in Stab, Miller, and Gurevych (2018), which distinguishes between non-arguments, supporting arguments and attacking arguments, which we denote as `NON/PRO/CON` accordingly. While this simple scheme grasps argumentative context, the annotation is easier since annotators are not required to consider complex relationships between argumentative components. Furthermore, it is also flexible enough to capture argumentative parts that are not attributable to the single argument type. For instance, in our dataset, we often observe rhetorical questions that criticize the paper's vagueness under review. The annotation scheme can also be interpreted as a flat version of the *claim-premise* model: There is a single *claim*, *"The paper should be accepted"*, and arguments are premises that either attack or support the claim.

Annotation Process In total, we have seven annotators, all of whom are graduate-level computer science students. The annotation is made token-wise and when presented a review, an annotator chooses argumentative text spans and assigns labels with the argument type to it. The document parts which are not explicitly annotated are considered to be non-argumentative. We refer to this annotation as *token-level* annotation.

Each review is randomly assigned to three different annotators. We resolve situations when a token is assigned with different labels by different annotators with a majority vote. In case a token is assigned with three different labels, we ask a independent fourth annotator who did not previously annotate the review to make the final annotation decision.

To obtain *sentence-level* annotations from annotated tokens, we mainly follow the procedure described in Trautmann et al. (2020a). Sentences without argumentative tokens are annotated with the label `NON`. For sentences con-

[1] www.argumentsearch.com

[2] www.args.me

[3] https://openreview.net/

[4] https://openreview-py.readthedocs.io/en/latest/getting_data.html

[5] https://zenodo.org/record/4314390

[6] https://github.com/fromm-m/aaai2021-am-peer-reviews

[7] https://www.nltk.org/

[8] The manually defined set contains e.g. "e.g", "i.e.", "et al.", "Fig.", etc.

[9] We end up with 15 reviews for iclr20, 14 reviews for iclr19 and 12 per each other conference

Conference	Number of Papers	Number of Reviews	Acceptance rate	avg words
ICLR'19	1,419	4,332	35 %	403
ICLR'20	2,213	6,722	27 %	409
MIDL'19	59	178	80 %	362
MIDL'20	144	544	55 %	255
NeuroAI'19	62	174	68 %	305
GI'20	65	174	82 %	507
Total	3,962	12,135	-	368

Table 1: Dataset statistics

	PRO	CON	NON	Total
number of tokens	3,259 (12%)	10,559 (34%)	14,684 (54%)	28,502
number of sentences	203 (14%)	640 (46%)	558 (40%)	1,401

Table 2: The table shows the distribution of the classes in the datasets. The distribution of the labels in the token-level dataset is skewed towards NON, and in the sentence-level dataset towards CON.

taining argumentative tokens, we count the number of argumentative segments, which overlap with it. An argumentative segment is comprised of a sequence of tokens with the same argumentative label without interruption. The sentence is assigned with the label of the majority of segments. If the number of segments with both labels is the same, we count the number of tokens with argumentative labels and assign the most frequent token label. As a result, we get 28,502 annotated tokens and 1,401 sentences. Table 2 presents the resulting class distribution.

Agreement

The agreement among annotators is an important criterion for the reliability of the annotation. Since our annotations are done on a token level and we have more than two annotators per review, we use the Krippendorff's alpha (Krippendorff et al. 2016) family of measures to assess the annotation quality. Each annotation can be seen as a set of annotated segments $(start, stop, label)$, where $start$ and $stop$ denote the segment's bounds and $label$ its class. We include all three classes for the computation of agreement.[10] Krippendorff's alpha now considers all pairs of overlapping segments and compares the expected and the observed disagreements in the annotations. For better comparability we follow recent related work (Trautmann et al. 2020a) and compute the following two variants: ${}_{cu}\alpha$ only considers the agreement in the label, while ${}_{u}\alpha$ additionally takes the length of the overlap into account. For both variants, the perfect agreement corresponds to the value of 1, the score for a random agreement is zero and negative values are possible if the agreement is worse than random. For our annotation, we obtain ${}_{u}\alpha = 0.568$ and ${}_{cu}\alpha = 0.861$, which is comparable to related work (Trautmann et al. 2020a).

Another possibility to assess the agreement is to compute the Macro F_1 metric for individual annotators. In terms of the Macro F_1 score, the quality of our annotations is better than of comparable datasets (Trautmann et al. 2020a; Reimers et al. 2019), see Human Performance in Table 3. Thus, we conclude that our annotation is reliable for further experiments.

[10]The score also accounts for imbalanced classes, see e.g. (Artstein and Poesio 2008).

Experimental Setup

In the following, we discuss our experimental setup. The description applies for both token-level and sentence-level evaluation unless noted otherwise.

Problem Setting Our goal is to identify *supporting* and *opposing* arguments in scientific peer-reviews and separate them from non-argumentative text. To get a detailed analysis of the models' performance and possible bottlenecks, we first decouple the problem of *argument identification* from *stance detection* and solve them separately. Afterward, we jointly solve both problems by a single model and obtain a model performance for our desired application. Therefore, we define the following tasks:

1. Argumentation Detection: A binary classification of whether a text span is an argument. The classes are denoted by ARG and NON, where ARG is the union of PRO and CON classes.
2. Stance Detection: A binary classification whether an argumentative text span is supporting or opposing the paper acceptance. The model is trained and evaluated only on argumentative PRO and CON text spans.
3. Joint Detection: A multi-class classification between the classes PRO, CON and NON, i.e. the combination of argumentation and stance detection.

Evaluation

We split our dataset sentence-wise 7:1:2 into training, validation and test sets stratified by class, i.e. keeping the same ratio among classes in all three subsets. The validation set is used for hyperparameter optimization and early stopping, whereas the test set is only used to evaluate the final model performance reported in the result section. We report the macro F_1 score. The F_1 is defined as the harmonic mean of

precision and recall and Macro F_1 is the mean over the class-individual scores. Since Macro F_1 weights classes equally independently of class' size, it is insensitive to the class imbalance problem. We train each model ten times with different random seeds and report the mean performance.[11] To check the significance of our results, we use a two-sided t-test with a significance level 1%.

Methods

Since transfer learning achieves state-of-the-art results for AM on different datasets (Reimers et al. 2019; Fromm, Faerman, and Seidl 2019; Trautmann et al. 2020a) we also apply it for our task. We employ a transformer (Vaswani et al. 2017) based BERT model (Devlin et al. 2019) with fine-tuning on different datasets. We include the following model variants in our evaluation:

Majority Baseline The majority baseline labels the instances with the most frequent class.

ArgBERT To assess the new dataset necessity, we evaluate the zero-shot learning performance of a BERT model fine-tuned on another AM dataset annotated on token and sentence level with the same scheme (Trautmann et al. 2020a). The other dataset comprises heterogeneous data found on the internet, and therefore, the resulting model is supposed to be universally applicable.

PeerBERT-ArgInit We initialize the model with the weights of ArgBERT and additionally fine-tune it on our new dataset. We hypothesize that the model can take advantage of the argumentative structure learned on another dataset.

PeerBERT Smaller BERT model with 110M parameters fine-tuned on our dataset (based on `bert-base-cased`).

PeerBERT-L Larger BERT model with 340M parameters fine-tuned on our dataset (based on `bert-large-cased`).

Human Performance An interesting experiment for assessing the applicability of the proposed solution is the comparison with the human performance on the task. To compute the human performance, we treat each annotator analogously to the model. Therefore, we compare labels produced by each annotator to the final annotations and compute the Macro F_1 score. The reported score is the mean among scores of all annotators.[12]

Training

We use a weighted cross-entropy loss to tackle the class imbalance problem, where the weight is given as the reciprocal of the number of samples of this class. The class weights are defined individually for each task and dataset. The models are trained using either `bert-base-cased` or `bert-large-cased`, with training batch size 100 for bert-base and 32 for bert-large. We use the AdamW optimizer with a learning rate of 10^{-5} for all models and early stopping with a patience of 3.

Results

In this section, we present the results of our experiments, which we have designed to answer the following research questions:

1. How well does the automatic mining of arguments work for peer-reviews?
2. Can we transfer knowledge from pre-existing annotated argumentation datasets?
3. How well does the approach generalize across different conferences?
4. How relevant are arguments in the decision making process for scientific publications?

Automatic Mining of Arguments

The results for the three AM tasks and all methods are summarized in Table 3. Our most important observation is that automatic argument extraction performs close to human performance and can be relied upon in the peer-review domain. Surprisingly, the detection of the stance in the peer-review domain appears to be considerably easier than identifying arguments. For other datasets annotated with the same scheme, we observe an inverse effect, see Table 4. Although there is no explicit stance detection experiment in the other works, we can infer it from the inferior results of joint detection compared against the argument detection results.

When comparing our results to other datasets on the token level, we observe that our results are substantially better, with a difference of about 10 % points. A reason might be that we operate on a single domain while other datasets contain heterogeneous documents covering multiple domains. However, we observe a significant performance difference when comparing our results on sentence and token level. To identify the reasons, we analyze the label ambiguity within sentences in our dataset. We found out that 22% of sentences for the argumentation detection task and 23% of those for the stance detection task contain tokens annotated with both classes. Therefore, we conclude that while it is still possible to achieve acceptable performance on the sentence level, the difference to the token level is more evident in our dataset.

Finally, the experiment regarding knowledge transfer from another AM dataset reveals transfer difficulties. The zero-shot performance is better than the majority vote only on the simpler stance detection task, but it is clearly outperformed by the models directly trained on our dataset. The additional intermediate fine-tuning step on the other AM dataset does not bring significant improvement either compared to directly fine-tuning on our dataset, cf. PeerBERT.

Training Set Size Figure 2 presents the model performance for different training set sizes. We can observe that pretraining on the other AM dataset does not help, even if the training set is small. The performance saturates when about

[11] To avoid the clutter, we provide the variance across the different runs in the appendix

[12] The resulting score should be seen as the upper bound for human performance since we use the same annotations for ground-truth.

Detection	Argument		Stance		Joint	
Level	Sentence	Token	Sentence	Token	Sentence	Token
Majority Baseline	0.351	0.350	0.423	0.434	0.234	0.233
ArgBERT	0.316	0.353	0.719	0.644	0.203	0.241
PeerBERT-ArgInit	0.718	0.877	0.852	0.862	0.734	0.796
PeerBERT	**0.789**	0.896	0.893	0.849	0.728	0.808
PeerBERT-L	0.763	**0.900**	**0.936**	**0.930**	**0.757**	**0.839**
Human Performance	0.885	0.873	0.978	0.980	0.881	0.860

Table 3: Overview of the results for different Argument Mining tasks on token and sentence level. We show results in terms of Macro F_1 for different BERT model variants, as well as the majority baseline and human performance estimate. In bold font, we highlight the best performance of our models per task and level.

Detection	Argument		Joint	
Level	Sentence	Token	Sentence	Token
UKP	0.810	-	0.690	-
AURC	-	0.782	0.725	0.743
Ours	0.789	0.900	0.757	0.839

Table 4: Comparison of maximum Macro F_1 values obtained for different datasets from literature, UKP (Stab, Miller, and Gurevych 2018; Fromm, Faerman, and Seidl 2019) and AURC (Trautmann et al. 2020a).

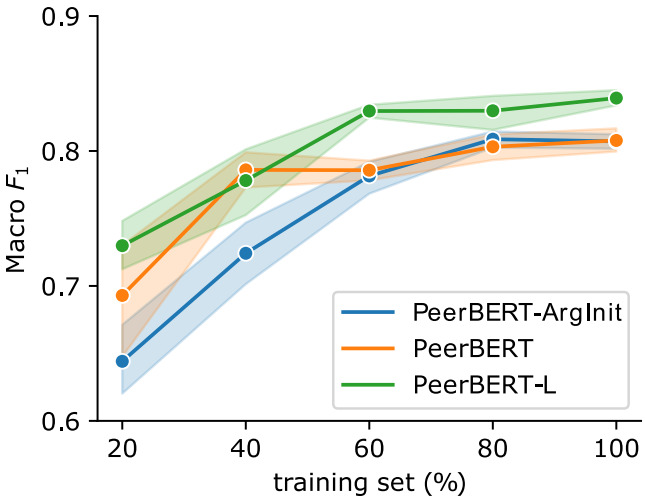

Figure 2: The Macro-F_1 evaluated on the task of joint prediction on the token level. The shaded areas indicate confidence intervals across ten runs with different random seeds.

Detection	Argument	Joint
ALL	0.891	0.823
NO-GI	0.873	0.791

Table 5: Comparison of Macro F_1 values for sentences from GI-20 reviews, when training with/without sentences from reviews from GI-20. All tasks are done on token-level.

60% of the training set is used. Therefore, we conclude that we have collected enough annotations. Similar behavior has been observed for the other tasks at both sentence and token level.

Generalization Across Conferences

In this section, we study the model's generalization to peer-reviews for papers from other (sub)domains. To this end, we reduce the test set to only contain reviews from the GI'20 conference. The focus of the GI'20 conference is Computer Graphics and Human-Computer Interaction, while the other conferences are focused on Representation Learning, AI and Medical Imaging. We consider the GI'20 as a subdomain since all conferences are from the domain of computer science. As a model, we choose our PeerBERT-L model and train on two different training sets:

NO-GI The original training dataset with all sentences from reviews of GI'20 removed.

ALL A resampling of the original training dataset of the same size as NO-GI, with sentences from all conferences.

Table 5 presents the experimental results. We observe a small performance decrease on both tasks, about two points on argument detection and three on joint detection tasks. At the same time, we also observe similar behavior when comparing results obtained on the whole test set (Table 3) and only on GI'20 reviews by the ALL model. Therefore the more considerable drop is not necessary due to the worse generalization and can be explained by the more challenging task. Overall, the drops are relatively small, and we conclude that the model generalizes well across subdomains.

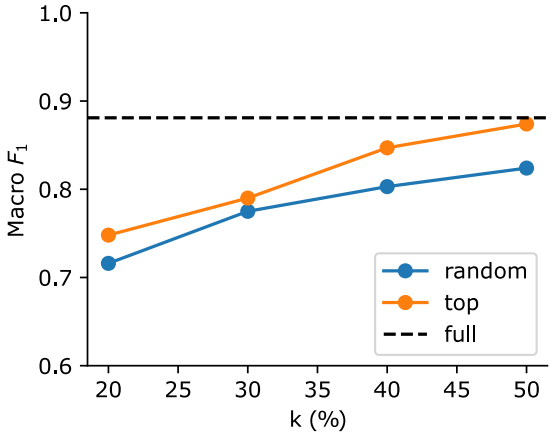

Figure 3: Evaluation of acceptance classification performance in F_1-measure based on different sentence selection methods. Using the top k% sentences according to argumentativeness likelihood results in superior performance compared to random selection. With 50% of the text, almost the same performance is reached as with the full review.

Relevance for Decision-making

In previous experiments, we have shown that peer-reviews contain arguments and these arguments can be identified automatically. In this section, we want to verify the usefulness of the extracted arguments for the decision making process. As a proxy to evaluate the usefulness, we design an experiment where the acceptance/rejection decisions made solely by considering arguments are compared to the decisions supported by taking full reviews into account. Therefore, we use the unannotated rest of our dataset and assign a probability to be an argument to each sentence with our best performing PeerBERT-L model. Now, we can compare three different settings for the decision-making process:

Full The decision-makers are allowed to see all reviews completely. This particularly includes decision suggestions often encountered in reviews that are not annotated as arguments in our dataset.

Top-K Arguments The decision-makers are only allowed to see the $k\%$ sentences with the highest probability to be arguments from each review. Note that the high probability to be identified as an argument does not necessarily correlate with the strength of the argument.

Random-K Decision-makers are only allowed to see $k\%$ randomly selected sentences from each review. We do not exclude explicit decision suggestions here.

We consider sentence level in this experiment despite the better performance of our model on the token-level. The main reason is a fair comparison with the Random-k setting, random sampling of words would result in large gaps and meaningless texts, especially for small k.

To avoid manual expenditure, we decide to apply a language model as a decision-maker. Since we also have a decision for each paper in our dataset, we train models to make an acceptance/rejection decision for the different settings described above. The standard BERT model is not directly applicable for this task since combining the reviews for a single paper often exceeds the input length restriction of at most 512 tokens. Therefore, we employ ToBERT (Pappagari et al. 2019), a model proposed for the classification of the long texts. It splits texts into multiple segments and individual segments are first used for the finetuning of the BERT model. In a second step, a second transformer model on the top combines representations of the segments and makes the final decision.

The results in terms of F_1-measure are given in Figure 3. We observe that selecting according to argumentativeness likelihood improves classification performance consistently in terms of F_1, compared to the random selection baseline, if at least a third of the review text is taken into consideration. The fraction of argumentative sentences in the annotated part of our dataset is 60%, cf. Table 2. We can achieve almost the same performance as the classifier trained on the full reviews while only considering 50% of the review. This is particularly impressive considering that reviews often already contain decision suggestions. Therefore, we conclude that arguments, which can be automatically extracted from reviews, are essential for the decision making process.

Conclusion

In this work, we have presented a new Argument Mining based approach for the assistance of different actors in the peer-review process. We have demonstrated that arguments are present in peer-reviews and that their identification with different stances can be made automatically. We have also shown that the peer-review domain is different from other previous Argument Mining applications, and therefore, there is a need for a new dataset. We have presented a new dataset that we make available for the community and have performed an extensive evaluation. We have also analyzed the editorial decision-making process and have empirically demonstrated that it is driven by argumentation.

In future work, we plan to address the problem of automatic determination of argument strength. Ranking arguments, according to their strength, is an undoubtedly useful feature for the potential application. For this purpose, we intend to extend our decision-making model and analyze single arguments' influence on the final decision.

Another useful feature, especially for the editorial team, would be identifying similar arguments in different reviews of the same paper.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and by the Deutsche Forschungsgemeinschaft (DFG) within the project Relational Machine Learning for Argument Validation (ReMLAV), Grant Number SE 1039/10-1, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999). The infrastructure for the course was provided by the Leibniz-Rechenzentrum. The authors of this work take full responsibilities for its content.

References

Aharoni, E.; Polnarov, A.; Lavee, T.; Hershcovich, D.; Levy, R.; Rinott, R.; Gutfreund, D.; and Slonim, N. 2014. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics. In *Proceedings of the First Workshop on Argumentation Mining*, 64–68. Baltimore, Maryland: Association for Computational Linguistics. doi:10.3115/v1/W14-2109. URL https://www.aclweb.org/anthology/W14-2109.

Amgoud, L.; and Prade, H. 2009. Using arguments for making and explaining decisions. *Artificial Intelligence* 173(3-4): 413–436.

Artstein, R.; and Poesio, M. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34(4): 555–596.

Daxenberger, J.; Eger, S.; Habernal, I.; Stab, C.; and Gurevych, I. 2017. What is the Essence of a Claim? Cross-Domain Claim Identification. *CoRR* abs/1704.07203.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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, Volume 1 (Long and Short Papers)*, 4171–4186.

Freeman, J. B. 2011. *Argument Structure: Representation and Theory*. Springer.

Fromm, M.; Faerman, E.; and Seidl, T. 2019. TACAM: Topic And Context Aware Argument Mining. In *2019 IEEE/WIC/ACM Int. Conf. on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*, 99–106.

Gao, Y.; Eger, S.; Kuznetsov, I.; Gurevych, I.; and Miyao, Y. 2019. Does My Rebuttal Matter? Insights from a Major NLP Conference. *CoRR* abs/1903.11367. URL http://arxiv.org/abs/1903.11367.

Guggilla, C.; Miller, T.; and Gurevych, I. 2016. CNN- and LSTM-based Claim Classification in Online User Comments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2740–2751. Osaka, Japan: The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1258.

Habernal, I.; and Gurevych, I. 2016. Argumentation Mining in User-Generated Web Discourse. *CoRR* abs/1601.02403.

Haddadan, S.; Cabrio, E.; and Villata, S. 2019. Yes, we can! Mining Arguments in 50 Years of US Presidential Campaign Debates. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4684–4690. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1463. URL https://www.aclweb.org/anthology/P19-1463.

Hua, X.; Nikolov, M.; Badugu, N.; and Wang, L. 2019. Argument Mining for Understanding Peer Reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2131–2137. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1219. URL https://www.aclweb.org/anthology/N19-1219.

Hua, X.; and Wang, L. 2017. Understanding and Detecting Supporting Arguments of Diverse Types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 203–208. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-2032. URL https://www.aclweb.org/anthology/P17-2032.

Krippendorff, K.; Mathet, Y.; Bouvry, S.; and Widlöcher, A. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity* 50(6): 2347–2364.

Lawrence, J.; and Reed, C. 2015. Combining Argument Mining Techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, 127–136. Denver, CO: Association for Computational Linguistics. doi:10.3115/v1/W15-0516. URL https://www.aclweb.org/anthology/W15-0516.

Miller, T.; Sukhareva, M.; and Gurevych, I. 2019. A Streamlined Method for Sourcing Discourse-level Argumentation Annotations from the Crowd. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1790–1796. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1177. URL https://www.aclweb.org/anthology/N19-1177.

Nguyen, H.; and Litman, D. 2015. Extracting Argument and Domain Words for Identifying Argument Components in Texts. In *Proceedings of the 2nd Workshop on Argumentation Mining*, 22–28. Denver, CO: Association for Computational Linguistics.

Nguyen, H.; and Litman, D. 2016. Context-aware Argumentative Relation Mining. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1127–1137. Berlin, Germany: Association for Computational Linguistics.

Palau, R. M.; and Moens, M.-F. 2009. Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In *Proc. of the 12th Int. Conf. on Artificial Intelligence and Law*, ICAIL '09, 98–107. New York, NY, USA: ACM. ISBN 978-1-60558-597-0.

Pappagari, R.; Zelasko, P.; Villalba, J.; Carmiel, Y.; and Dehak, N. 2019. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 838–844. IEEE.

Plank, B.; and van Dalen, R. 2019. CiteTracked: A Longitudinal Dataset of Peer Reviews and Citations. In *BIRNDL@ SIGIR*, 116–122.

Reimers, N.; Schiller, B.; Beck, T.; Daxenberger, J.; Stab, C.; and Gurevych, I. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 567–578. Florence, Italy:

Association for Computational Linguistics. doi:10.18653/v1/P19-1054. URL https://www.aclweb.org/anthology/P19-1054.

Stab, C.; Daxenberger, J.; Stahlhut, C.; Miller, T.; Schiller, B.; Tauchmann, C.; Eger, S.; and Gurevych, I. 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: demonstrations*, 21–25.

Stab, C.; and Gurevych, I. 2014a. Annotating Argument Components and Relations in Persuasive Essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 1501–1510. Dublin, Ireland: Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-1142.

Stab, C.; and Gurevych, I. 2014b. 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. URL http://aclweb.org/anthology/D/D14/D14-1006.pdf.

Stab, C.; and Gurevych, I. 2017. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics* 43(3): 619–659.

Stab, C.; Miller, T.; and Gurevych, I. 2018. Cross-topic Argument Mining from Heterogeneous Sources Using Attention-based Neural Networks. *CoRR* abs/1802.05758.

Teufel, S.; Siddharthan, A.; and Batchelor, C. 2009. Towards Domain-Independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 1493–1502. Singapore: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D09-1155.

Toulmin, S. E. 1958. *The Uses of Argument*. Cambridge University Press.

Trautmann, D.; Daxenberger, J.; Stab, C.; Schütze, H.; and Gurevych, I. 2020a. Fine-Grained Argument Unit Recognition and Classification. In *AAAI*.

Trautmann, D.; Fromm, M.; Tresp, V.; Seidl, T.; and Schütze, H. 2020b. Relational and Fine-Grained Argument Mining. *Datenbank-Spektrum* 1–7.

van Eemeren, F.; Blair, J.; Willard, C.; and Henkemans, A. 2003. *Anyone Who Has a View: Theoretical Contributions to the Study of Argumentation*, volume 8. Kluwer Academic Publishers. ISBN 978-1-4020-1456-7. doi:10.1007/978-94-007-1078-8.

Van Eemeren, F.; Grootendorst, R.; and van Eemeren, F. H. 2004. *A systematic theory of argumentation: The pragma-dialectical approach*. Cambridge University Press.

Van Eemeren, F. H.; Grootendorst, R.; and Kruiger, T. 2019. *Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies*, volume 7. Walter de Gruyter GmbH & Co KG.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wachsmuth, H.; Potthast, M.; Al-Khatib, K.; Ajjour, Y.; Puschmann, J.; Qu, J.; Dorsch, J.; Morari, V.; Bevendorff, J.; and Stein, B. 2017. Building an Argument Search Engine for the Web. In *Proceedings of the 4th Workshop on Argument Mining*, 49–59. Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/W17-5106. URL https://www.aclweb.org/anthology/W17-5106.

Walton, D. 2012. Argument Mining by Applying Argumentation Schemes. *Studies in Logic* 4.

Xiao, Y.; Zingle, G.; Jia, Q.; Shah, H. R.; Zhang, Y.; Li, T.; Karovaliya, M.; Zhao, W.; Song, Y.; Ji, J.; et al. 2020. Detecting Problem Statements in Peer Assessments. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 704–709.