CATALPA_EduNLP at PragTag-2023

Yuning Ding[1], Marie Bexte[1], and Andrea Horbach[1,2]
[1]CATALPA, FernUniversität in Hagen, Germany
[2]Hildesheim University, Germany

Abstract

This paper describes our contribution to the PragTag-2023 Shared Task. We describe and compare different approaches based on sentence classification, sentence similarity, and sequence tagging. We find that a BERT-based sentence labeling approach integrating positional information outperforms both sequence tagging and SBERT-based sentence classification. We further provide analyses highlighting the potential of combining different approaches.

1 Introduction

This paper describes the CATALPA_EduNLP entry to the First Shared Task on Pragmatic Tagging of Peer Reviews (Dycke et al., 2023). In this task, sentences within peer-reviews for academic articles from various domains are assigned a label expressing the pragmatic function of that sentence, namely *Recap*, *Strength*, *Weakness*, *Todo*, *Structure* or *Other* (Kuznetsov et al., 2022).

We experiment with various approaches presented in Section 3 and 4. As there is no clear winner among them (see results in Section 5), we further focus on comparing them to see under which conditions each setting works best (Section 6).

2 Datasets

We participated in all three evaluation setups of the Shared Task, which provided different amounts of training data. In the **full-data** setting, 117 reviews with 2326 sentences are provided, from which we split ten reviews to serve as our internal validation data. In the **low-data** setting, 33 reviews with 739 sentences are used (we take five of these reviews as our internal testing data or perform four-fold cross-validation). In the **no-data** setting, we use our internal test data from the **full-data** setting for evaluation.

The Shared Task provides two additional data sets: **F1000raw** contains unlabeled data (7423 reviews from the same domains as the training data. To use this data, we first extract the domain for each article via a lookup of the respective gateway on `https://f1000research.com`. Since a large number of articles cannot be assigned any domain, we only use articles for which we can assign a domain, yielding 269 additional *iscb*, 144 *rpkg*, 445 *diso*, 525 *case* and 227 *scip* reviews. The **ARR-22** dataset consists of 684 labeled reviews coming from a different domain and using a different annotation scheme (Dycke et al., 2022). While some of the mappings are straightforward (*paper summary* to *Recap*, *summary of strengths* to *Strength*, *summary of weaknesses* to *Weakness*), we mapped *comments, suggestions and typos* to *Todo* and found no correspondences for *Structure* and *Other*.

3 Approaches with Training Data

We explore three complementary approaches, following similar tasks of identifying sections in scientific articles or abstracts that cast the problem as one of sentence classification (Mullen et al., 2005; Teufel and Kan, 2009) or sequence labeling (Hirohata et al., 2008): A BERT-based sentence classification model (Liu et al., 2019), a Longformer-based sequence tagging model (Beltagy et al., 2020), and a SBERT-based model (Reimers and Gurevych, 2019) to compute semantic similarity between sentences. The total training and inference time was about 22 hours on a single GPU.

3.1 BERT-Based Sentence Classification

This set of approaches are extensions of the Roberta-based baseline released with the Shared Task training data. In the **full-data** setting, apart from experimenting with a different variant of pretrained models (*roberta-large*) (Liu et al., 2019), we also included positional information (**+ Pos.**), by providing either the absolute position of the respective sentence within a review and the relative position by normalizing the former by the number of sentences in that review. Besides, the

Proceedings of the 10th Workshop on Argument Mining, pages 197–201

one-hot-encoded review domain is also used as an additional feature (**+ Domain**). These additional features are concatenated to the sentence embedding as an array. The combined representation was used to train the classification layer. To provide contextual information, we append the full review text after the sentence to be classified after a special separator token in the **+ Context** setting.

Reviews often contain domain-specific words occurring mainly in one domain, but not the others such as “malaria" in the “diso" domain or “cytobrowser" in “iscb". To improve the cross-domain generalizability of the model, we compute for each word (in its original form) a metric inspired by tf-idf where we set the frequency in the domain (using the F1000raw dataset to have a broader data basis) in relation to its general frequency provided by the wordfreq Python package[1] in its default setting. We replaced words exceeding a certain threshold (Equation 1) with a special <term> token. In addition, tokens containing the string “http" were replaced by a special <link> token and tokens without any letters by a <non_letter> token. We named this approach as **+ Word Normalization**.

$$\frac{domain\ frequency}{general\ frequency + 0.5} > 1 \qquad (1)$$

Combining the approaches above, we made a domain-specific model selection where sentences from a certain domain are scored by the model that performed best on this domain during validation. The result is reported as **Best**.

Using the Additional ARR-22 data We experimented with the **ARR-22** dataset as additional training data (**+ ARR**), but found the label distribution to be very different from the main training data. (The majority class in ARR-22 is “weakness", while “Todo" is the dominant class in the **full data**.) Therefore, we sampled the mapped elements in ARR-22 dataset according to the class distribution in the full-data. No further filtering or normalization was applied to this dataset.

3.2 Longformer-based Sequence Tagging

This approach follows Ding et al. (2022) to inherently integrate a sentence’s context into the prediction. We applied it on the **full data** setting. Since it shows no advantage compared to the other sentence classification approaches, we didn’t apply it to other settings. It utilizes tokens with gold-standard annotation represented by Inside-Outside-Beginning (IOB) tags. For example, the gold-standard annotation ***Recap**: “The paper proposes ..."* will be represented as ***B-Recap**: The,* ***I-Recap**: paper,* ***I-Recap**: proposes, ...* These labeled tokens are input into a pretrained Longformer language model (longformer-large-4096) for token classification. We trained for 10 epochs and then used the model with the best performance on the validation data to predict a label for each token in the test data. Each sentence got the most frequent token label assigned. We also tested the **+ Word Normalization** approach from the sentence classification in this setting.

3.3 SBERT-Based Sentence Classification

In this approach, we follow the similarity-based content scoring methodology described in Bexte et al. (2022) and Bexte et al. (2023), making predictions based on the most similar reference examples and fine-tuning an SBERT model (Reimers and Gurevych, 2019) for 10 epochs with a batch size of 8, otherwise sticking to default values.

In the **full-data** setting, we train eight separate models and take their majority vote to obtain predictions on the test data. Five of these models are experts for one of the five domains in the dataset. These are therefore trained on the respective subset of the training data (fine-tuning the *All-MiniLM-L6-v2* base model). The remaining three models are trained across all domains: An **overall** model builds training pairs across all training instances, while the training instances of two **within-domain** models (one based on *All-Mini-LM-L6-v2*, the other on *All-MiniLM-L12-v2*) are restricted to pairs of sentences from the same domain.

We pursue the same similarity-based approach in the **low-data** setting: First, we train a single model on our internal split of the limited training data. We then further pursue a 4-fold cross-validation. We found it beneficial to augment the training data using the auxiliary data from F1000Research. For each of our models from the cross-validation, we select additional reference sentences in the following way: For each target label, we include the 15 nearest neighbors, i.e., those we find the highest similarity to an existing reference answer to. This is done for three rounds, after which the resulting extended set of reference data is used to make predictions on the test data by taking the label of

[1] https://pypi.org/project/wordfreq/

the most similar reference element[2] To prepare our submission to the challenge, we again perform a majority voting, taking the four votes of the augmented models from our cross-validation and that of the model trained on our internal train-test split.

4 Zero-Shot Approaches

This section describes our **no-data** approaches.

4.1 Clustering

Using a pretrained SBERT model (*All-MiniLM-L12-v2*), we encode representations of the target labels to serve as the centroids of clusters. These representations are derived from the label descriptions the challenge organizers gave and a set of at most three keywords per label (see Appendix A.1). Each answer from the testing data is then assigned to the label representation with the highest cosine similarity, thus predicting the respective label for this test instance.

4.2 GPT

We also explore using large commercial language models in a zero-shot setting. We prompt the GPT3.5 through the openai API by providing label definitions in the Shared Task description. As a post-processing step, we replace labels not corresponding to one of the six categories provided with *Other*.

5 Results

Following the evaluation scheme in the Shared Task, we report macro-averaged F1-scores per domain for our own data split and only an overall F1-score for the challenge test set.

Table 1 shows the results of our internal splits of the data. For the **full-data** setup, we see that adding additional information like position, domain, or context to the BERT-based model does only improve the results for individual domains but leads to performance drops on others, so there is no substantial improvement overall (column *mean*). However, if we select per domain the setup performing best on the training data, we see an overall improvement on the test data (.88 vs .82 for the baseline model.) Adding the ARR data as additional training data led to decreased performance, although sampling the ARR data to a similar distribution to the main

[2]We also experimented with additional fine-tuning using this augmented training set but found this not helpful.

	Domain					
	case	diso	iscb	rpkg	scip	mean
	Full-data					
BERT-based						
Roberta-large	.80	.87	.88	.75	.77	.82
+ Word Normalization	.87	**.88**	.94	.68	.56	.79
+ Pos.	.76	.85	.92	.74	.77	.81
+ Domain	**.89**	.79	.82	.69	.81	.80
+ Context	.87	.83	.83	.75	.76	.81
+ Pos., Context	.87	.82	**.94**	**.83**	.70	.83
+ Pos., Context, Domain	.83	.81	.88	.72	**.85**	.82
Best	.89	.88	.94	.83	.85	**.88**
+ ARR	.60	.72	.78	.61	.67	.68
+ ARR Sampled	.68	.66	.78	.60	.78	.70
Sequence Tagging	.67	.65	.72	.56	.51	.62
+ Word Normalization	.59	.65	.77	.56	.53	.62
SBERT-based						
ALL	.82	.78	.83	.64	.85	.78
ALL_large	.71	.74	.86	.67	.70	.74
ALL_cross	.84	.74	.77	.67	.77	.76
Domains	.75	.76	.66	.74	.80	.74
Voting	.88	.81	.84	.67	.77	.79
	Low-data					
BERT-based						
Roberta-large	.10	.17	.19	.11	.24	.16
SBERT-based						
Train-test split	.52	1.0	.70	.91	.68	.76
4-fold CV	.71	.71	.77	.66	.69	.71
4-fold CV + aux	.74	.72	.80	.65	.74	.73
	No-data					
SBERT-based	.19	.33	.17	.22	.15	.21
GPT	.53	.54	.46	.24	.42	.44

Table 1: F1 results on our internal validation split.

Setting	Submission	mean
Final	Roberta large + Pos., Text	.81
Full-data	Roberta large + Pos., Text	.82
Low-data	SBERT 4-fold voting	.75
No-data	SBERT clustering	.22

Table 2: F1 results on challenge test data.

training data helped somewhat. Both the SBERT-based model and the sequence tagging approach did not reach the performance of the BERT-based model in the full-data setup (.62 and .79 vs .88 in the best configuration).

However, the situation changes drastically when the amount of available training data is reduced (**low-data**). In this scenario, the BERT-based model could hardly learn anything while the SBERT-based model reached a performance close to the **full-data** setup. Note that the results are not directly comparable across the different dataset variants, as the test data is not identical. Performance in the **no-data** setting is unsurprisingly again reduced, with GPT outperforming our SBERT-based clustering method.

Table 2 shows the methods that led to the best

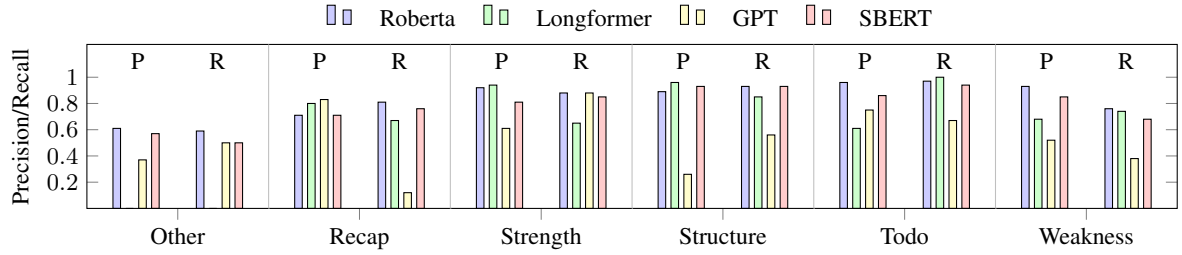

Figure 1: Per-label precision and recall of our different methods on our internal test data of the full-data setting.

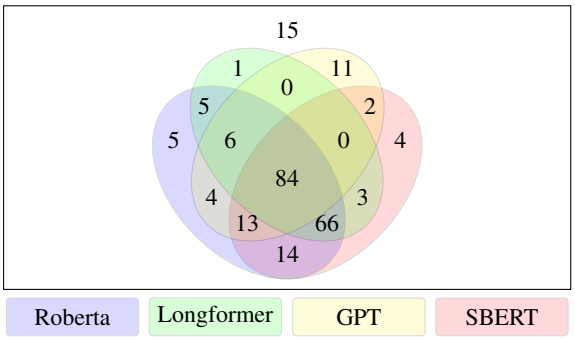

Figure 2: Venn diagram of how many sentences are classified correctly by which methods.

performance on the challenge test data in the different settings. Unfortunately, our **Best** approach on the validation set did not beat the **+Pos., Context** approach on the test data in the **full-data** setting. Therefore, we only submit the **+Pos., Context** approach in the final round.

The test data in the **final** setting contains unseen data from a "secret" domain, which might explain the slight performance drop (.82 vs. .81). But our approach reaches the second-best performance on the data from the "secret" domain with an F1-score of 0.79 on the leaderboard, indicating its good generalization ability.

In the **low-data** setting, our SBERT-based method performs better than the BERT-based methods, which consists of the results observed on the validation set. The **no-data** performance of our SBERT-based method is slightly better on the test set than the average on our validation splits. (Following the competition rules regarding reproducibility, we did not submit our GPT results since the model requires a paid API.)

6 Analysis

The different approaches produce results in the same ballpark so that one may wonder if they can be used interchangeably. To investigate this we compare the results by checking four conditions: The percentage of sentences that **all** four models judge correctly, the percentage that **none** of the models classified correctly, which proportion is classified correctly in a **majority** setting and the percentage of correctly classified sentences that could be reached in an **oracle** condition if we knew to which model a sentence should be passed, i.e. the percentage of sentences judged correctly by at least one model.

For this analysis, we use the respective best-performing model variant on our internal split of the data provided for the full data setting. We analyze all four approaches we took: Sentence classification using Roberta, similarity-based classification with SBERT, sequence tagging using the longformer architecture, and zero-shot application of GPT. Figure 2 gives an overview of how many sentences are correctly classified by which method. The **oracle** condition sums up to 94% of test instances being assigned the correct label, meaning that the remaining six percent are classified correctly by **none** of the methods. About a third (36%) of the data is correctly solved by **all** four models, and a **majority** voting over their predictions comes up to 83% accuracy, which is 1% lower than what Roberta achieves on its own.

Overall, GPT seems the most distinct from the other methods: It has the highest number of 11 sentences that none of the other methods can classify correctly. Such sentences often have the label *Other*, for example “Dear Authors". However, there are 66 sentences for which all other methods except GPT predict the correct label. GPT rarely labeled instances of “Recap" correctly and often mislabeled “Structure" as “Other", such as “Reviewer response for version 1". Figure 1 breaks down performance for the individual labels, revealing GPT to be much worse in both precision and recall when it comes to *Structure*, and showing especially low recall for *Recap*. All methods have the most difficulty with sentences labeled *Other*, with our se-

quence tagging approach having both precision and recall of zero. The overall best-performing Roberta method especially shows superiority in terms of high and balanced precision and recall values for the labels *Strength*, *Todo*, and *Weakness*.

7 Conclusion

We have presented experiments using a variety of very different approaches. The comparison shows that they behave quite differently and that a sensible combination of approaches yields further improvements. Future work therefore has to determine which approach is most suitable for a given item to be classified.

Acknowledgements

This work was partially conducted at "CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics" of the FernUniversität in Hagen, Germany.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. Similarity-based content scoring - How to make S-BERT keep up with BERT. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1892–1903.

Yuning Ding, Marie Bexte, and Andrea Horbach. 2022. Don't drop the topic - the role of the prompt in argument identification in student writing. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 124–133, Seattle, Washington. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2022. NLPEER: A unified resource for the computational study of peer review. *arXiv preprint arXiv:2211.06651*.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. Overview of PragTag-2023: Low-resource multi-domain pragmatic tagging of peer reviews. In *Proceedings of the 10th Workshop on Argument Mining*, Singapore. Association for Computational Linguistics.

Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, and Mitsuru Ishizuka. 2008. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Computational Linguistics*, 48(4):949–986.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tony Mullen, Yoko Mizuta, and Nigel Collier. 2005. A baseline feature set for learning rhetorical zones using full articles in the biomedical domain. *ACM SIGKDD Explorations Newsletter*, 7(1):52–58.

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)*. Association for Computational Linguistics.

Simone Teufel and Min-Yen Kan. 2009. Robust argumentative zoning for sensemaking in scholarly documents. In *Natural Language Processing for Digital Libraries Workshop*, pages 154–170. Springer.

A Appendix

A.1 Keywords for Zero-Shot Clustering Label Assignment

'Todo': ['should', 'could', 'need'], 'Strength': ['good', 'strength', 'clear'], 'Weakness': ['weakness', 'shortcoming', 'flaw'], 'Structure': ['reviewer'], 'Recap': ['authors', 'describe', 'article'], 'Other': ['other']