Dataset Independent Baselines for Relation Prediction in Argument Mining

Oana Cocarascu, Elena Cabrio, Serena Villata, Francesca Toni

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200490

Dataset Independent Baselines for Relation Prediction in Argument Mining

Oana COCARASCU [a,c], Elena CABRIO [b], Serena VILLATA [b], Francesca TONI [a]
[a] *Imperial College London, UK*
[b] *Université Côte d'Azur, CNRS, Inria, I3S, France*
[c] *King's College London, UK*

Abstract. Argument(ation) Mining (AM) is the research area which aims at extracting argument components and predicting argumentative relations (i.e., *support* and *attack*) from text. In particular, numerous approaches have been proposed in the literature to predict the relations holding between arguments, and application-specific annotated resources were built for this purpose. Despite the fact that these resources were created to experiment on the same task, the definition of a single relation prediction method to be successfully applied to a significant portion of these datasets is an open research problem in AM. This means that none of the methods proposed in the literature can be easily ported from one resource to another. In this paper, we address this problem by proposing a set of dataset independent strong neural baselines which obtain homogeneous results on all the datasets proposed in the literature for the argumentative relation prediction task in AM. Thus, our baselines can be employed by the AM community to compare more effectively how well a method performs on the argumentative relation prediction task.

Keywords. Argument Mining, Relation Prediction, Machine Learning Methods

1. Introduction

Argument(ation) Mining (AM) is "the general task of analyzing discourse on the pragmatics level and applying a certain argumentation theory to model and automatically analyze the data at hand" [16]. Two tasks are crucial in AM [22,6,20]: *1)* argument component detection within the input natural language text, aiming at the identification of the textual boundaries of the arguments and their classification (claim, premise); and *2)* relation prediction, aiming at identifying (support, attack) relations between argumentative components, possibly identified in the first stage. In this paper we focus on the second task. Despite the high volume of approaches tackling the relation prediction task with satisfying results (see [6] for an extensive list), a problem arises: these solutions heavily rely on the peculiar features of the dataset taken into account for the experimental setting and are hardly portable from one application domain to another. On the one side, this issue can be explained by the huge number of heterogeneous application domains where argumentative text may be analysed (e.g., online reviews, blogs, political debates, legal cases). On the other side, it represents a drawback for the comparison of the different approaches proposed in the literature, which are often presented as solutions addressing the relation prediction task from a dataset independent point of view. A side drawback for

	essays	micro	nk	db	ibm	com	web	cdcp	ukp	aif
# attacks	497	108	378	141	1069	296	1301	0	5935	9854
# supports	4841	263	353	179	1325	462	1329	1220	4759	7543

Table 1. Datasets' statistics.

the AM community is therefore a lack of large annotated resources for this task, as most available resources cannot be successfully reused, being highly context-based. Even the employment of pretrained language models (e.g., BERT [12]) does not address this issue.

In this paper, we tackle this issue by proposing a set of strong cross-dataset baselines based on different neural architectures. Our baselines are shown to perform homogeneously over all the datasets proposed in the literature for the relation prediction task in AM, differently from individual methods proposed in the literature. Our contribution is to bestow the AM community with a set of strong cross-dataset baselines to compare with in order to demonstrate how well a relation prediction method for AM performs.

We focus on two types of argumentative relations: *attack* and *support*, given that the majority of datasets target only these two types of relations. We define neural baselines to address the corresponding binary classification problem, analysing, to the best of our knowledge, all available datasets for this task, ranging from persuasive essays to user-generated content, to political speeches. Given two arguments, we are interested in determining the argumentative relation between the first, called *child* argument, and the second, called *parent* argument, using a neural model. For example, the child argument *People know video game violence is fake* may attack the parent argument *Youth playing violent games exhibit more aggression*. In our baselines, each of the two arguments is represented using embeddings as well as other features. We propose three neural network architectures for the classification task, two concerned with the way child and parent are passed through the network (*concat* model and *mix* model), and an attention-based model. We also explore BERT as an alternative to our baselines: although this is used successfully to boost performances for other tasks in Natural Language Processing, it is generally not competitive for relation prediction with the datasets we consider.

We conduct experiments with a number of datasets, chosen either because they were specially created for relation prediction in AM or because they can be easily transformed to be used for this task. These are: Essays (essay) [33], Microtexts (micro) [29], Nixon-Kennedy (nk) [23], Debatepedia (db) [5], IBM (ibm) [1], ComArg (com) [3], Web-content (web) [7], CDCP (cdcp) [28], UKP (ukp) [34], AIFdb (aif) [2,10,18,31]. Datasets' statistics can be found in Table 1[1].

2. Neural baselines for relation prediction

We use four types of features: word embeddings, sentiment features, syntactic features, computed for both child and parent, and textual entailment from child to parent. We refer to the last three types of features as *standard* features. Word embeddings are distributed representations of texts in an n-dimensional space. Textual entailment represents the class (amongst entailment, contradiction, or neutral) obtained using AllenNLP[2], a textual entailment model based on decomposable attention [27]. The features related to

[1] For more details about the individual datasets, we refer the reader to the relevant publications.

[2] https://allennlp.org

sentiment are based on manipulation of SentiWordNet [15] and the sentiment of the entire (child and parent) texts analysed using the VADER sentiment analyser [17]. Every WordNet synset [24] can be associated to three scores describing how objective, positive, and negative it is. For every word in the (child and parent) texts, we select the first synset and compute its positive score and its negative score. In summary, the features related to sentiment for a text t that consists of n words, $W_i = 1 \ldots w_n$, are the following: (i) sentiment score ($\sum_{w_i} pos\_score(w_i) - neg\_score(w_i)$), (ii) number of positive/negative/neutral words in t, (iii) sentiment polarity class and score of t. Syntactic features consist of text statistics (e.g., number of words) and word statistics with respect to part-of-speech tags (i.e., number of words, nouns, verbs, first person singular, etc.) and lexical diversity (i.e., number of unique words divided by the total number of words in text t).

We describe the three neural architectures we propose for determining the argumentative relation (of attack or support) holding between child and parent. For all, we report only configurations of the architectures and number/size of the hidden layers which performed the best[3]. For our models, we use GRUs [11] as they take less time to train and are more efficient.

Concat model (C). In this model, each of the child and parent embeddings is passed through a GRU. We concatenate the standard features of the child and of the parent. The merged standard vector is then concatenated with the outputs of the GRUs. The resulting vector is passed through 2 dense layers (of 256 neurons and 64 neurons, with sigmoid as activation function), and then to softmax to determine the argumentative relation.

Mix model (M). In this model, we first concatenate the child and parent embeddings and then pass them through a GRU, differently from the concat model where we pass each embedding vector through a GRU first. We concatenate the standard features that we obtain for the child and for the parent. The merged standard vector is then concatenated with the output of the GRU. From this stage, the network resembles the concat model: the resulting vector is passed through 2 dense layers (of 256 neurons and 64 neurons, with sigmoid as activation function), to be then finally passed to softmax.

Attention model (A). Inspired by the demonstrated effectiveness of attention-based models [36,35], we combine the GRU-based model with attention mechanisms. Each of the child and parent embeddings is passed through a GRU and we compute attention in two directions. We concatenate the standard features of the child and of the parent. The merged standard vector is then concatenated with the outputs of the GRUs. The resulting vector is passed through a single dense layers (128 neurons, with sigmoid as activation function), that is then passed to softmax.

3. Experimental results

Non-neural baselines. For training we have used the larger datasets, *aif*, *essay*, *ibm* and *web*. We resampled the minority class from the *essay* dataset and used our models on the oversampled dataset. We did not use for training the *ukp* dataset as the parent is a topic instead of an argument. The models were then tested on the remaining datasets, with the average being computed on testing datasets. We report the F_1 performance of

[3] We also experimented with 1 and 2 hidden layers, and hidden layer sizes of 32, 64, 128, and 256, trying all possible combinations towards best configurations. We did not consider a higher number of hidden layers due to the small size of the data.

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
non-neural baselines	RF	F_1 A	0.32	0.24	0.40		0.33	0.38	-	0.39	0.55	0.44	0.381	0.508
		F_1 S	0.57	0.74	0.64		0.67	0.59	0.85	0.53	0.59	0.54	0.636	
	RF	F_1 A	0.57	0.40	0.45	0.53	0.43		-	0.60	0.52	0.57	0.509	0.490
		F_1 S	0.44	0.47	0.52	0.41	0.57		0.51	0.45	0.50	0.38	0.472	
	SVM	F_1 A		0.34	0.36	0.33	0.29	0.38	-	0.42	0.42	0.40	0.368	0.503
		F_1 S		0.71	0.67	0.65	0.67	0.59	0.84	0.57	0.56	0.49	0.639	
	SVM	F_1 A	0.49	0.35	0.39	0.39	0.38		-	0.56	0.57	0.520	0.456	0.498
		F_1 S	0.50	0.54	0.52	0.59	0.60		0.67	0.46	0.47	0.500	0.539	

Table 2. Experimental results for non-neural baselines with F_1 for **A**ttack and for **S**upport. The blanks indicate the training dataset. The Average (Avg) and the Macro (Macr) Avg exclude the results for the training datasets.

the *attack* class (A) and the *support* class (S) for the non-neural baselines in Table 2. We used Random Forests (RF) [4] with 15 trees in the forest and gini impurity criterion and SVM with linear kernel using LIBSVM [9], obtained as a result of performing a grid search, as it is the most commonly used algorithm in the works that experiment on the datasets we considered [1,3,8,23,25]. On top of the *standard* features used for our neural models, for the baselines we added the following features: TF-IDF, number of common nouns, verbs and adjectives between the two texts as in [23], a different sentiment score $\frac{\text{nr\_pos}-\text{nr\_neg}}{\text{nr\_pos}+\text{nr\_neg}+1}$ as in [1], with all features being normalized.

Neural baselines with non-contextualised word embeddings. Table 3 shows the best baselines for relation prediction in AM. We experimented with GloVe (300-dimensional) embeddings [30], using pre-trained word representations in all our models. We used 100 as the sequence size as we noticed that there are few instances with more than 100 words. We used a batch size of 32 and trained for 10 epochs (as a higher number of epochs led to overfitting). We report the results using embeddings and *syntactic* features and the results with *all* the features presented in Section 2. We also conducted a feature ablation experiment (with embeddings being always used) and observed that syntactic features contribute the most to performance, with the other types of features bringing small improvements when used together only with embeddings. In addition, we have run experiments using two datasets for training to test whether combining two datasets improves performance. During training, we used one of the large datasets (*aif*, *essay*, *ibm*, *web*) and one of the remaining datasets (represented as blanks in the table).

Amongst the proposed architectures, the attention model generally performs better. Using only a single dataset for training, the model that performs the best is the *mix* model using *all* features and trained on the *essay* dataset. The best results are obtained when using another dataset along one of the larger datasets for training. This is because combining data from two domains we are able to learn better the types of argumentative relations. When using *syntactic* features, adding *micro*, *cdcp*, and *ukp* does not improve the results compared to using a single dataset for training. Indeed, *cdcp* has only one type of relation (i.e. support) resulting in an imbalanced dataset, and in *ukp*, the parent argument is a topic, which does not improve the prediction task. When using *all* features, *micro*, *com*, *ukp*, and *nk* do not contribute to an increase in performance. The best performing model is the attention mechanism trained on the *web* and *essay* datasets using *syntactic* features (0.544 macro average F_1).

Neural baselines with contextualised word embeddings. Contextualised word embeddings such as the Bidirectional Encoder Representations from Transformers (BERT) em-

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
embed. + syntactic	C G	F_1 A		0.35	0.43	0.48	0.31	0.45	-	0.58		0.43	0.433	0.526
		F_1 S		0.71	0.68	0.58	0.70	0.54	0.77	0.47		0.50	0.619	
	A G	F_1 A		0.37	0.58		0.53	0.53	-	0.61	0.59	0.55	0.537	0.526
		F_1 S		0.61	0.60		0.42	0.50	0.72	0.43	0.38	0.47	0.516	
	A G	F_1 A		0.36	0.48	0.43	0.39		-	0.52	0.45	0.51	0.449	**0.544**
		F_1 S		0.75	0.66	0.62	0.68		0.79	0.52	0.56	0.54	0.640	
all features	M G	F_1 A		0.37	0.43	0.43	0.40	0.46	-	0.71	-	0.46	0.466	0.532
		F_1 S		0.71	0.64	0.61	0.70	0.55	0.78	0.11	0.78	0.51	0.599	
	A G	F_1 A		0.36	0.54		0.50	0.51	-	0.59	0.59	0.55	0.520	0.535
		F_1 S		0.67	0.63		0.49	0.51	0.74	0.47	0.41	0.49	0.551	
	A G	F_1 A		0.43	0.54	0.49	0.46		-	0.59	0.63	0.63	0.539	0.539
		F_1 S		0.68	0.55	0.57	0.56		0.65	0.46	0.38	0.47	0.540	

Table 3. Experimental results with F_1 for **A**ttack and for **S**upport for the **C**oncat, **M**ix, and **A**ttention architectures, with **G**loVE embeddings. The blanks indicate the training datasets. The Average (Avg) and the Macro (Macr) Avg do not include the results for the training datasets.

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
BERT + syntactic	4B	F_1 A			0.55	0.47	0.53	0.50	-	0.56	0.48	0.45	0.506	0.526
	2D	F_1 S			0.61	0.57	0.59	0.49	0.69	0.47	0.48	0.46	0.545	
	4B	F_1 A		0.36	0.48	0.40	0.45	0.42	-	0.53		0.37	0.430	0.525
	1D	F_1 S		0.69	0.67	0.61	0.62	0.57	0.79	0.50		0.50	0.619	
	4B	F_1 A	0.50	0.36	0.46		0.50		-	0.52	0.47	0.50	0.473	**0.537**
	2D	F_1 S	0.61	0.62	0.59		0.61		0.74	0.52	0.50	0.61	0.600	
all features	4B	F_1 A			0.53	0.50	0.54	0.51	-	0.59	0.51	0.49	0.524	0.529
	1D	F_1 S			0.59	0.56	0.55	0.47	0.67	0.45	0.48	0.49	0.533	
	3B	F_1 A	0.48	0.34	0.48		0.45		-	0.45	0.50	0.54	0.463	0.532
	2D	F_1 S	0.57	0.65	0.60		0.64		0.73	0.55	0.52	0.54	0.600	

Table 4. Experimental results with F_1 for **A**ttack and for **S**upport relations. XB stands for the number of BERT layers used (X=3,4) and YD stands for the number of dense layers (Y=1,2) used before the final layer that predicts the class. The blanks indicate the training datasets. The Average (Avg) and the Macro (Macr) Avg do not include the results for the training datasets.

beddings [12] analyse the entire sentence before assigning embeddings to individual words. We employ BERT embeddings to test whether they bring any improvements to the classification task. While for GloVe vectors we do not need the original, trained model in order to use the embeddings, for the BERT embeddings we require the pre-trained language models that we can then fine tune using the datasets of the downstream task. We try different combinations: using 3 or 4 BERT layers and using 1 dense layer (of 64 neurons) or 2 dense layers (of 128 and 32 neurons, respectively) before the final layer that determines the class. Table 4 shows the results with BERT embeddings instead of GloVe, using feature ablation (*syntactic* vs *all* features) and two datasets for training to test whether this can improve performance. The best results are obtained using 4 BERT layers and 2 dense layers (0.537 macro average F_1). However, this best BERT baseline does not outperform the best results with the attention model and GloVe embeddings.

4. Related work

In terms of results reported on the datasets we have conducted our experiments on, most works perform a cross-validation evaluation or, in the case of datasets consisting of several topics, the models proposed are trained on some of the topics and tested on the remaining topics. For *essay*, an Integer Linear Programming model was used to achieve 0.947 F_1 for *support* and 0.413 F_1 for *attack* on the testing dataset using cross-validation to select the model [33]. Using SVM, 0.946 F_1 for *support* and 0.456 F_1 for *attack* were obtained [33]. Using a modification of the Integer Linear Programming model to accommodate the lack of some features used for *essay* but not present in *micro*, 0.855 F_1 was obtained for *support* and 0.628 F_1 for *attack*. On *micro*, an evidence graph model was used to achieve 0.71 F_1 using cross-validation [29]. On *nk*, 0.77 F_1 for *attack* and 0.75 F_1 for *support* were obtained using SVM and cross-validation [23]. SVM accuracy results on the testing dataset using coverage (i.e. number of claims identified over the number of total claims) were reported in [1] as follows: 0.849 accuracy for 10% coverage, 0.740 accuracy for 60% coverage, 0.632 accuracy for 100% coverage. RF were evaluated on *web* and *aif* using cross-validation, achieving 0.717 F_1 and 0.831 F_1, respectively [8]. Structured SVMs were evaluated in a cross-validation setting on *cdcp* and *ukp* using various types of factor graphs, full and strict [25]. On *cdcp*, F_1 was 0.493 on the full graph and 0.50 on the strict graph, whereas on *ukp*, F_1 was 0.689 on the full graph and 0.671 on the strict graph. No results on the two-class datasets were reported for *db*, *com*, and *ukp*. The results on *ukp* treat either supporting and attacking arguments as a single category or consider three types of relations: *support*, *attack*, *neither*. The latter type of reporting results on three classes is also given on the *com*.

Some other works have started investigating the dataset independence in AM. [26] showed how models may overlook textual content when provided with the context surrounding the span by relying on contextual markers for predicting relations and tested their method on the *essay* dataset. [21] integrated (claim and other domain) lexicon information into neural networks with attention tested on *ukp*. [19] experimented with span representations, originally developed for other tasks, on the *essay* dataset. Other works have used contextualised word embeddings for relation prediction in AM [13,32]. More recently, [14] proposed and tested on *ukp* an argument retrieval system.

5. Discussion and Conclusion

Dataset independence is one of the biggest challenges in AM. An AM model for relation prediction trained on every individual dataset we considered in this paper would perform better than any general baseline on that dataset. We believe an AM model would require leveraging a diverse corpus to be of use in a real-world system. Most works have previously focused on a moderate-sized corpus distributed across a small set of topics [14]. This paper is a step towards the applicability of AM techniques across datasets. Our baselines perform homogeneously in terms of average over all existing datasets for relation prediction in AM while using generic features. We propose as baseline the model that performed the best, with the baseline using attention mechanism with GloVe embeddings and syntactic features trained on the *web* and *essay* datasets (0.544 macro average F_1). The results for the *attack* class are generally worse than those for *support* as

the datasets that are used in training (e.g. *essay*, *ibm*) have fewer instances for the *attack* class than for *support* (see Table 1). The datasets differ at granularity: some consist of pairs of sentences (e.g., *ibm*) whereas others include pair of multiple-sentence arguments (e.g., *nk*). Additionally, the argumentative on relations can be domain-specific and their semantic nature may vary between corpora (e.g., *com*). We considered the unified task of determining *support* or *attack* between any two texts.

Embeddings represent the differentiating feature for the models we experimented with. Whilst word embeddings are often used as the first data processing layer in a deep learning model, we employed TF-IDF features for the non-neural models that we considered as baselines. Other works that address the task of relation prediction make use of features specific to the single dataset of interest, making it difficult to test those models on other datasets. For instance, for the *essay* dataset, [33] use structural features such as number of preceding and following tokens in the covering sentence, number of components in paragraph, number of preceding and following components in paragraph, relative position of the argument component in paragraph. For the other datasets, [34] use topic similarity features (as the *parent* argument is a topic), [23] use the position of the topic and similarity with other related/unrelated pair from the dataset, keyword embeddings of topics from the dataset. We have used only general purpose features that are meaningful for all datasets addressing the relational AM task. Surprisingly, BERT embeddings (achieving state-of-the-art performances in many tasks [12]) do not bring improvements here, compared to non-contextualised word embeddings.

To conclude, several resources have been built recently for the task of argumentative relation prediction, covering different topics like political speeches, Wikipedia articles, persuasive essays. Given the heterogeneity of these different kinds of text, it is hard to compare cross-dataset the different proposed approaches. We addressed this non-portability issue by making a broad comparison of different deep learning methods using both non-contextualised and contextualised word embeddings for a large set of datasets for the argumentative relation prediction task, an important and still widely open problem. We proposed a set of strong dataset-independent baselines based on several neural architectures and have shown that our models perform homogeneously over all existing datasets for relation prediction in AM.

References

[1] Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A., Slonim, N.: Stance classification of context-dependent claims. In: Proceedings of EACL. pp. 251–261 (2017)
[2] Bex, F., Modgil, S., Prakken, H., Reed, C.: On logical specifications of the argument interchange format. Journal of Logic and Computation 23(5), 951–989 (2013)
[3] Boltužić, F., Šnajder, J.: Back up your stance: Recognizing arguments in online discussions. In: Proceedings of the 1st Workshop on Argumentation Mining. pp. 49–58 (2014)
[4] Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
[5] Cabrio, E., Villata, S.: Node: A benchmark of natural language arguments. In: Proceedings of COMMA. pp. 449–450 (2014)
[6] Cabrio, E., Villata, S.: Five years of argument mining: a data-driven analysis. In: Proceedings of IJCAI. pp. 5427–5433 (2018)
[7] Carstens, L., Toni, F.: Towards relation based argumentation mining. In: Proceedings of the 2nd Workshop on Argumentation Mining. pp. 29–34 (2015)
[8] Carstens, L., Toni, F.: Using argumentation to improve classification in natural language problems. ACM Transactions on Internet Technology 17(3), 30:1–30:23 (2017)

[9] Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology TIST 2(3), 27:1–27:27 (2011)
[10] Chesñevar, C.I., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G.R., South, M., Vreeswijk, G., Willmott, S.: Towards an argument interchange format. Knowledge Eng Review 21(4), 293–316 (2006)
[11] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In: Proceedings of EMNLP. pp. 1724–1734 (2014)
[12] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
[13] Durmus, E., Ladhak, F., Cardie, C.: Determining relative argument specificity and stance for complex argumentative structures. In: Proceedings of ACL. pp. 4630–4641 (2019)
[14] Ein-Dor, L., Shnarch, E., Dankin, L., Halfon, A., Sznajder, B., Gera, A., Alzate, C., Gleize, M., Choshen, L., Hou, Y., Bilu, Y., Aharonov, R., Slonim, N.: Corpus wide argument mining - A working solution. In: Proceedings of AAAI. pp. 7683–7691 (2020)
[15] Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. pp. 417–422 (2006)
[16] Habernal, I., Gurevych, I.: Argumentation mining in user-generated web discourse. Computational Linguistics 43(1), 125–179 (2017)
[17] Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of ICWSM (2014)
[18] Iyad, R., Reed, C.: The argument interchange format. In: Argumentation in Artificial Intelligence, pp. 383–402. Springer (2009)
[19] Kuribayashi, T., Ouchi, H., Inoue, N., Reisert, P., Miyoshi, T., Suzuki, J., Inui, K.: An empirical study of span representations in argumentation structure parsing. In: Proceedings of ACL. pp. 4691–4698 (2019)
[20] Lawrence, J., Reed, C.: Argument mining: A survey. Computational Linguistics 45(4), 765–818 (2019)
[21] Lin, J., Huang, K.Y., Huang, H., Chen, H.: Lexicon guided attentive neural network model for argument mining. In: Proceedings of the 6th Workshop on Argument Mining. pp. 67–73 (2019)
[22] Lippi, M., Torroni, P.: Argumentation mining: State of the art and emerging trends. ACM Transactions on Internet Technology 16(2), 10 (2016)
[23] Menini, S., Cabrio, E., Tonelli, S., Villata, S.: Never retreat, never retract: Argumentation analysis for political speeches. In: Proceedings of AAAI. pp. 4889–4896 (2018)
[24] Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38, 39–41 (1995)
[25] Niculae, V., Park, J., Cardie, C.: Argument mining with structured SVMs and RNNs. In: Proceedings of ACL. pp. 985–995 (2017)
[26] Opitz, J., Frank, A.: Dissecting content and context in argumentative relation analysis. In: Proceedings of the 6th Workshop on Argument Mining. pp. 25–34 (2019)
[27] Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of EMNLP. pp. 2249–2255 (2016)
[28] Park, J., Cardie, C.: A corpus of eRulemaking user comments for measuring evaluability of arguments. In: Proceedings of LREC (2018)
[29] Peldszus, A., Stede, M.: Joint prediction in MST-style discourse parsing for argumentation mining. In: Proceedings of EMNLP. pp. 938–948 (2015)
[30] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of EMNLP. pp. 1532–1543 (2014)
[31] Reed, C., Wells, S., Devereux, J., Rowe, G.: AIF+: dialogue in the argument interchange format. In: Proceedings of COMMA. pp. 311–323 (2008)
[32] Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: Proceedings of ACL. pp. 567–578 (2019)
[33] Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. Computational Linguistics 43(3), 619–659 (2017)
[34] Stab, C., Miller, T., Schiller, B., Rai, P., Gurevych, I.: Cross-topic argument mining from heterogeneous sources. In: Proceedings of EMNLP (2018)
[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of NeurIPS. pp. 6000–6010 (2017)
[36] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: Proceedings of NAACL HLT. pp. 1480–1489 (2016)