Context-aware Argumentative Relation Mining

Huy V. Nguyen
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260
hvn3@pitt.edu

Diane J. Litman
Computer Science Department and
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260
dlitman@pitt.edu

Abstract

Context is crucial for identifying argumentative relations in text, but many argument mining methods make little use of contextual features. This paper presents context-aware argumentative relation mining that uses features extracted from writing topics as well as from windows of context sentences. Experiments on student essays demonstrate that the proposed features improve predictive performance in two argumentative relation classification tasks.

1 Introduction

By supporting tasks such as automatically identifying *argument components*[1] (e.g., premises, claims) in text, and the *argumentative relations* (e.g., support, attack) between components, argument (argumentation) mining has been studied for applications in different research fields such as document summarization (Teufel and Moens, 2002), opinion mining (Boltužić and Šnajder, 2014), automated essay evaluation (Burstein et al., 2003), legal information systems (Palau and Moens, 2009), and policy modeling platforms (Florou et al., 2013).

Given a pair of argument components with one component as the *source* and the other as the *target*, *argumentative relation mining* involves determining whether a relation holds from the source to the target, and classifying the argumentative function of the relation (e.g., support vs. attack). Argumentative relation mining - beyond argument component mining - is perceived as an essential step towards more fully identifying the argumentative structure of a text (Peldszus and Stede, 2013; Sergeant, 2013; Stab et al., 2014). Consider the second paragraph shown in Figure 1. Only detecting the argument components (a claim in sentence 2 and two premises in sentences 3 and 4) does not give a complete picture of the argumentation. By looking for relations between these components, one can also see that the two premises together justify the claim. The argumentation structure of the text in Figure 1 is illustrated in Figure 2.

> **Essay 73**. Topic: Is image more powerful than the written word?
> ...(1)Hence, *I agree only to certain degree that in today's world, image serves as a more effective means of communication*$_{[MajorClaim]}$.
> (2)Firstly, **pictures can influence the way people think**$_{[Claim]}$. (3)For example, nowadays horrendous images are displayed on the cigarette boxes to illustrate the consequences of smoking$_{[Premise]}$. (4)As a result, statistics show a slight reduction in the number of smokers, indicating that they realize the effects of the negative habit$_{[Premise]}$...

Figure 1: Excerpt from a student persuasive essay (Stab and Gurevych, 2014a). Sentences are numbered and argument components are tagged.

Our current study proposes a novel approach for argumentative relation mining that makes use of contextual features extracted from surrounding sentences of *source* and *target* components as well as from topic information of the writings.

[1]There is no consensus yet on an annotation scheme for argument components, or on the minimal textual units to be annotated. We follow Peldszus and Stede (2013) and consider "*argument mining as the automatic discovery of an argumentative text portion, and the identification of the relevant components of the argument presented there.*" We also borrow their term "*argumentative discourse unit*" to refer to the textual units (e.g., text segment, sentence, clause) which are considered as argument components.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1127–1137,
Berlin, Germany, August 7-12, 2016.

Prior argumentative relation mining studies have often used features extracted from argument components to model different aspects of the relations between the components, e.g., relative distance, word pairs, semantic similarity, textual entailment (Cabrio and Villata, 2012; Stab and Gurevych, 2014b; Boltužić and Šnajder, 2014; Peldszus and Stede, 2015b). Features extracted from the text surrounding the components have been less explored, e.g., using words and their part-of-speech from adjacent sentences (Peldszus, 2014). The first hypothesis investigated in this paper is that the *discourse relations* of argument components with adjacent sentences (called *context windows* in this study, a formal definition is given in §5.3) can help characterize the argumentative relations that connect pairs of argument components. Reconsidering the example in Figure 1, without knowing the content "*horrendous images are displayed on the cigarette boxes*" in sentence 3, one cannot easily tell that "*reduction in the number of smokers*" in sentence 4 supports the "*pictures can influence*" claim in sentence 2. We expect that such content relatedness can be revealed from a discourse analysis, e.g., the appearance of a discourse connective "*As a result*".

While topic information in many writing genres (e.g., scientific publications, Wikipedia articles, student essays) has been used to create features for argument component mining (Teufel and Moens, 2002; Levy et al., 2014; Nguyen and Litman, 2015), topic-based features have been less explored for argumentative relation mining. The second hypothesis investigated in this paper is that features based on *topic context* also provide useful information for improving argumentative relation mining. In the excerpt below, knowing that '*online game*' and '*computer*' are topically related might help a model decide that the claim in sentence 1 supports the claim in sentence 2:

(1)**People who are addicted to games, especially online games, can eventually bear dangerous consequences**$_{[Claim]}$.

(2)Although it is undeniable that computer is a crucial part of human life$_{[Premise]}$, **it still has its bad side**$_{[MajorClaim]}$.[2]

Motivated by the discussion above, we propose *context-aware argumentative relation mining* – a novel approach that makes use of contextual features that are extracted by exploiting context sentence windows and writing topic to improve relation prediction. In particular, we derive features using *discourse relations* between argument components and windows of their surrounding sentences. We also derive features using an *argument and domain word lexicon* automatically created by post-processing an essay's topic model. Experimental results show that our proposed contextual features help significantly improve performance in two argumentative relation classification tasks.

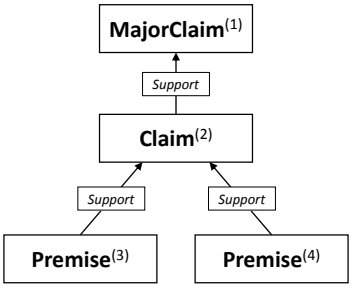

Figure 2: Structure of the argumentation in the excerpt. Relations are illustrated accordingly to the annotation provided in the corpus. Premises 3 and 4 were annotated for separate relations to Claim 2. Our visualization should not mislead that the two premises are linked or convergent.

2 Related Work

Unlike argument component identification where textual inputs are typically sentences or clauses (Moens et al., 2007; Stab and Gurevych, 2014b; Levy et al., 2014; Lippi and Torroni, 2015), textual inputs of argumentative relation mining vary from clauses (Stab and Gurevych, 2014b; Peldszus, 2014) to multiple-sentences (Biran and Rambow, 2011; Cabrio and Villata, 2012; Boltužić and Šnajder, 2014). Studying claim justification between user comments, Biran and Rambow (2011) proposed that the argumentation in justification of a claim can be characterized with discourse structure in the justification. They however only considered discourse markers but not discourse relations. Cabrio et al. (2013) conducted a corpus analysis and found certain similarity between Penn Discourse TreeBank relations (Prasad et al., 2008) and argumentation schemes (Walton et al., 2008). However they did not discuss how such similarity could be applied to argument mining.

Motivated by these findings, we propose to use features extracted from discourse relations be-

[2]In this excerpt, the Premise was annotated as an attack to the MajorClaim in sentence 2.

tween sentences for argumentative relation mining. Moreover, to enable discourse relation features when the textual inputs are only sentences/clauses, we group the inputs with their context sentences. Qazvinian and Radev (2010) used the term "context sentence" to refer to sentences surrounding a citation that contained information about the cited source but did not explicitly cite it. In our study, we only require that the context sentences of an argument component must be in the same paragraph and adjacent to the component.

Prior work in argumentative relation mining has used argument component labels to provide constraints during relation identification. For example, when an annotation scheme (e.g., (Peldszus and Stede, 2013; Stab and Gurevych, 2014a)) does not allow relations from claim to premise, no relations are inferred during relation mining for any argument component pair where the source is a claim and the target is a premise. In our work, we follow Stab and Gurevych (2014b) and use the predicted labels of argument components as features during argumentative relation mining. We, however, take advantage of an enhanced argument component model (Nguyen and Litman, 2016) to obtain more reliable argument component labels than in (Stab and Gurevych, 2014b).

Argument mining research has studied different data-driven approaches for separating organizational content (shell) from topical content to improve argument component identification, e.g., supervised sequence model (Madnani et al., 2012), unsupervised probabilistic topic models (Séaghdha and Teufel, 2014; Du et al., 2014). Nguyen and Litman (2015) post-processed LDA (Blei et al., 2003) output to extract a lexicon of argument and domain words from development data. Their semi-supervised approach exploits the topic context through essay titles to guide the extraction.

Finally, prior research has explored predicting different argumentative relationship labels between pairs of argument components, e.g., attachment (Peldszus and Stede, 2015a), support vs. non-support (Biran and Rambow, 2011; Cabrio and Villata, 2012; Stab and Gurevych, 2014b), {implicit, explicit}×{support, attack} (Boltužić and Šnajder, 2014), verifiability of support (Park and Cardie, 2014). Our experiments use two such argumentative relation classification tasks (Support vs. Non-support, Support vs. Attack) to evaluate the effectiveness of our proposed features.

3 Persuasive Essay Corpus

Stab and Gurevych (2014a) compiled the Persuasive Essay Corpus consisting of 90 student argumentative essays and made it publicly available.[3] Because the corpus has been utilized for different argument mining tasks (Stab and Gurevych, 2014b; Nguyen and Litman, 2015; Nguyen and Litman, 2016), we use this corpus to demonstrate our context-aware argumentative relation mining approach, and adapt the model developed by Stab and Gurevych (2014b) to serve as the baseline for evaluating our proposed approach.

Three experts identified possible argument components of three types within each sentence in the corpus (*MajorClaim* - writer's stance toward the writing topic, *Claim* - controversial statements that support or attack MajorClaim, and *Premise* - evidence used to underpin the validity of Claim), and also connected the argument components using two argumentative relations (*Support* and *Attack*). According to the annotation manual, each essay has exactly one MajorClaim. A sentence can have one or more argument components (*Argumentative* sentences). Sentences that do not contain any argument component are labeled *Non-argumentative*. Figure 1 shows an example essay with components annotated, and Figure 2 illustrates relations between those components. Argumentative relations are directed and can hold between a Premise and another Premise, a Premise and a (Major-) Claim, or a Claim and a MajorClaim. Except for the relation from Claim to MajorClaim, an argumentative relation does not cross paragraph boundaries. The three experts achieved inter-rater accuracy 0.88 for component labels and Krippendorff's α_U 0.72 for component boundaries. Given the annotated argument components, the three experts obtained Krippendorff's α 0.81 for relation labels. The number of relations are shown in Table 1.

4 Argumentative Relation Tasks

4.1 Task 1: Support vs. Non-support

Our first task follows (Stab and Gurevych, 2014b): given a pair of source and target argument components, identify whether the source argumentatively supports the target or not. Note that when a support relation does not hold, the source may attack or has no relation with the target compo-

[3] www.ukp.tu-darmstadt.de/data/argumentation-mining

Label	#instances
Within-paragraph constraint	
Support	989
Attack	103
No paragraph constraint	
Support	1312
Attack	161

Table 1: Data statistics of the corpus.

nent. For each of two argument components in the same paragraph[4], we form two pairs (i.e., reversing source and target). In total we obtain 6330 pairs, in which 989 (15.6%) have Support relation. Among 5341 Non-support pairs, 103 have Attack relation and 5238 are no-relation pairs.

Stab and Gurevych (2014b) split the corpus into an 80% training set and a 20% test set which have similar label distributions. We use this split to train and test our proposed models, and directly compare our models' performance to the reported performance in (Stab and Gurevych, 2014b).

4.2 Task 2: Support vs. Attack

To further evaluate the effectiveness of our approach, we conduct an additional task that classifies an argumentative relation as *Support* or *Attack*. For this task, we assume that the relation (i.e., attachment (Peldszus, 2014)) between two components is given, and aim at identifying the argumentative function of the relation. Because we remove the paragraph constraint in this task, we obtain more Support relations than in Task 1. As shown in Table 1, of the total 1473 relations, we have 1312 (89%) Support and 161 (11%) Attack relations. Because this task was not studied in (Stab and Gurevych, 2014b), we adapt Stab and Gurevych's model to use as the baseline.

5 Argumentative Relation Models

5.1 Baseline

We adapt (Stab and Gurevych, 2014b) to use as a baseline for evaluating our approach. Given a pair of argument components, we follow (Stab and Gurevych, 2014b) by first extracting 3 feature sets: structural (e.g., word counts, sentence position), lexical (e.g., word pairs, first words), and grammatical production rules (e.g., *S→NP,VP*).

[4]Allowing cross-paragraph relations exponentially increases the number of no-relation pairs, which makes the prediction data extremely skewed (Stab and Gurevych, 2014b).

Because a sentence may have more than one argument component, the relative component positions might provide useful information (Peldszus, 2014). Thus, we also include 8 new component position features: whether the source and target components are the whole sentences or the beginning/end components of the sentences; if the source is before or after the target component; and the absolute difference of their positions.

Stab and Gurevych (2014b) used a 55-discourse marker set to extract indicator features. We expand their discourse maker set by combining them with a 298-discourse marker set developed in (Biran and Rambow, 2011). We expect the expanded set of discourse markers will represent better possible discourse relations in the texts.

Stab and Gurevych (2014b) used predicted label of argument components as features for both training and testing their argumentation structure identification model.[5] As their predicted labels are not available to us, we adapt this feature set by using the argument component model in (Nguyen and Litman, 2016) which was shown to outperform the corresponding model of Stab and Gurevych.

For later presentation purposes, we name the set of all features from this section *except word pairs and production rules* as the **common features**. While word pairs and grammatical production rules were the most predictive features in (Stab and Gurevych, 2014b), we hypothesize that this large and sparse feature space may have negative impact on model robustness (Nguyen and Litman, 2015). Most of our proposed models replace word pairs and production rules with different combinations of new contextual features.

5.2 Topic-context Model

Our first proposed model (TOPIC) makes use of **Topic-context** features derived from a lexicon of argument and domain words for persuasive essays (Nguyen and Litman, 2015). Argument words (e.g., '*believe*', '*opinion*') signal the argumentative content and are commonly used across different topics. In contrast, domain words are specific terminologies commonly used within the topic (e.g., '*art*', '*education*'). The authors first use

[5]Stab and Gurevych (2014b) reported that including gold-standard labels of argument component in both training and testing phases yielded results close to human performance. Our preliminary experiment showed that including gold-standard argument component labels in training did not help when predicted labels were used in the test set.

topic prompts in development data of unannotated persuasive essays to semi-automatically collect argument and domain seed words. In particular, they used 10 argument seed words: *agree, disagree, reason, support, advantage, disadvantage, think, conclusion, result, opinion*. Domain seed words are those in the topic prompts but not argument seed words or stop words. The seeds words are then used to supervise an automated extraction of argument and domain words from output of LDA topic model (Blei et al., 2003) on the development data. The extracted lexicon consists of 263 (stemmed) argument words and 1806 (stemmed) domain words mapped to 36 LDA topics.[6] All argument words are from a single LDA topic while a domain word can map to multiple LDA topics (except the topic of argument words). Using the lexicon, we extract the following Topic-context features:

Argument word: from all word pairs extracted from the source and target components, we remove those that have at least one word not in the argument word list. Each argument word pair defines a boolean feature indicating its presence in the argument component pair. We also include each argument word of the source and target components as a boolean feature which is true if the word is present in the corresponding component. We count number of common argument words, the absolute difference in number of argument words between source and target components.

Domain word count: to measure the topic similarity between the source and target components, we calculate number of common domain words, number of pairs of two domain words that share an LDA topic, number of pairs that share no LDA topic, and the absolute difference in number of domain words between the two components.

Non-domain MainVerb-Subject dependency: we extract MainVerb-Subject dependency triples, e.g., *nsubj(belive, I)*, from the source and target components, and filter out triples that involve domain words. We model each extracted triple as a boolean feature which is true if the corresponding argument component has the triple.

Finally, we include the **common feature set**.

To illustrate the Topic-context features, consider the following source and target components. Argument words are in boldface, and domain words are in italic.

Essay 54. Topic: museum and art gallery will disappear soon?

Source: **more** and **more people can** *watch exhibitions* through *television* **or** *internet* at *home* **due to** *modern technology*$_{[Premise]}$

Target: **some people think** *museums* and *art galleries* **will** *disappear soon*$_{[Claim]}$

An argument word pair is **people-think**. There are 35 pairs of domain words. A pair of two domain words that share an LDA topic is *exhibitions-art*. A pair of two domain words that do not share any LDA topic is *internet-galleries*.

5.3 Window-context Model

Our second proposed model (WINDOW) extracts features from discourse relations and common words between context sentences in the *context windows* of the source and target components.

Definition. *Context window of an argument component is a text segment formed by neighboring sentences and the covering sentence of the component. The neighboring sentences are called context sentences, and must be in the same paragraph with the component.*

In this study, context windows are determined using *window-size* heuristics.[7] Given a window-size n, we form a context window by grouping the covering sentence with at most n adjacently preceding and n adjacently following sentences that must be in the same paragraph.

To minimize noise in feature space, we require that context windows of the source and target components must be mutually exclusive. Biran and Rambow (2011) observed that the relation between a source argument and a target argument is usually instantiated by some elaboration/justification provided in a support of the source argument. Therefore we prioritize the context window of source component when it overlaps with the target context window. Particularly, we keep overlapping context sentences in the source window, and remove them from the target window. For example, with window-size 1, context windows of the Claim in sentence 2 in Figure 1 and the Premise in sentence 4 overlap at sentence 3. When the Claim is set as source component, its

[6] An LDA topic is simply represented by a number, and should not be misunderstood with essay topics.

[7] Due to the paragraph constraint and window overlapping, window-size does not indicate the actual context window size. However, window-size tells what the maximum size a window can have.

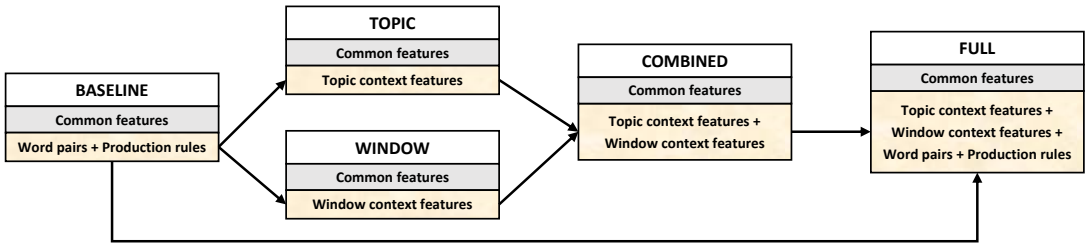

Figure 3: Features used in different models. Feature change across models are denoted by connectors.

context window includes sentences $\{2, 3\}$, and the Premise as a target has context window with only sentence 4. We extract three **Window-context** feature sets from the context windows:

Common word: as common word counts between adjacent sentences were shown useful for argument mining (Nguyen and Litman, 2016), we count common words between the covering sentence with preceding context sentences, and with following context sentences, for source and target components.

Discourse relation: for both source and target components, we extract discourse relations between context sentences, and within the covering sentence. We also extract discourse relations between each pair of source context sentence and target context sentence. Each relation defines a boolean feature. We extract both Penn Discourse Treebank (PDTB) relations (Prasad et al., 2008) and Rhetorical Structure Theory Discourse Treebank (RST-DTB) relations (Carlson et al., 2001) using publicly available discourse parsers (Ji and Eisenstein, 2014; Wang and Lan, 2015). Each PDTB relation has sense label defined in a 3-layered (class, type, subtype), e.g., *CONTINGENCY.Cause.result*. While there are only four semantic class labels at the class-level which may not cover well different aspects of argumentative relation, subtype-level output is not available given the discourse parser we use. Thus, we use relations at type-level as features. For RST-DTB relations, we use only relation labels, but ignore the nucleus and satellite labels of components as they do not provide more information given the component order in the pair. Because temporal relations were shown not helpful for argument mining tasks (Biran and Rambow, 2011; Stab and Gurevych, 2014b), we exclude them here.

Discourse marker: while the baseline model only considers discourse markers within the argument components, we define a boolean feature for each discourse marker classifying whether the marker is present before the covering sentence of the source and target components or not. This implementation aims to characterize the discourse of the preceding and following text segments of each argument component separately.

Finally, we include the **common feature set**.

5.4 Combined Model

While Window-context features are extracted from surrounding text of the argument components, which exploits the local context, the Topic-context features are an abstraction of topic-dependent information, e.g., domain words are defined within the context of topic domain (Nguyen and Litman, 2015), and thus make use of the global context of the topic domain. We believe that local and global context information represent complementary aspects of the relation between argument components. Thus, we expect to achieve the best performance by combining Window-context and Topic-context models.

5.5 Full Model

Finally, the FULL model includes all features in BASELINE and COMBINED models. That is, the FULL model is the COMBINED model plus word pairs and production rules. A summary of all models is shown in Figure 3.

6 Experiments

6.1 Task 1: Support vs. Non-support

Tuning Window-size Parameter

Because our WINDOW model uses a window-size parameter to form context windows of the source and target argument components, we investigate how the window-size of the context window impacts the prediction performance of the Window-context features. We set up a model with only Window-context features and determine the

	REPORTED	BASELINE	TOPIC	WINDOW	COMBINED	FULL
Accuracy	0.863	0.869	0.857	0.857	0.870	**0.877**
Kappa	–	0.445	0.407	0.449	**0.507***	0.481
Macro F1	0.722	0.722	0.703	0.724	**0.753***	0.739
Macro Precision	0.739	0.758	0.728	0.729	0.754	**0.777**
Macro Recall	0.705	0.699	0.685	0.720	**0.752***	0.715
F1:Support	0.519	0.519	0.488	0.533	**0.583***	0.550
F1:Non-support	0.920	0.925	0.917	0.916*	0.923	**0.929**

Table 2: Support vs. Non-support classification performances on test set. Best values are in bold. Values smaller than baseline are underlined. * indicates significantly different from the baseline ($p < 0.05$).

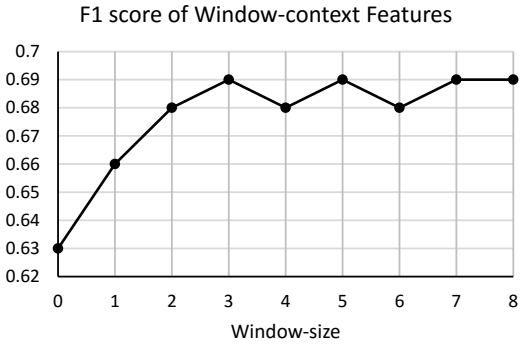

Figure 4: Performance of Window-context feature set by window-size.

window-size in range [0, 8][8] that yields the best F1 score in 10-fold cross validation. We use the training set as determined in (Stab and Gurevych, 2014b) to train/test[9] the models using LibLINEAR algorithm (Fan et al., 2008) without parameter or feature optimization. Cross-validations are conducted using Weka (Hall et al., 2009). We use Stanford parser (Klein and Manning, 2003) to perform text processing. As shown in Figure 4, while increasing the window-size from 2 to 3 improves performance (significantly), using window-sizes greater than 3 does not gain further improvement. We hypothesize that after a certain limit, larger context windows will produce more noise than helpful information for the prediction. Therefore, we set the window-size to 3 in all of our experiments involving Window-context model (all with a separate test set).

[8]Windows-size 0 means covering sentence is the only context sentence. We experimented with not using context sentence at all and obtained worse performance. Our data does not have context window with window-size 9 or larger.

[9]Note that via cross validation, in each fold some of our training set serves as a development set.

Performance on Test Set

We train all models using the training set and report their performances on the test set in Table 2. We also compare our baseline to the reported performance (REPORT) for Support vs. Non-support classification in (Stab and Gurevych, 2014b). The learning algorithm with parameters are kept the same as in the window-size tuning experiment. Given the skewed class distribution of this data, Accuracy and F1 of Non-support (the major class) are less important than Kappa, F1, and F1 of Support (the minor class). To conduct T-tests for performance significance, we split the test data into subsets by essays' ID, and record prediction performance for individual essays.

We first notice that the performances of our baseline model are better than (or equal to) REPORTED, except the Macro Recall. We reason that these performance disparities may be due to the differences in feature extractions between our implementation and Stab and Gurevych's, and also due to the minor set of new features (e.g., new predicted labels, expanded marker set, component position) that we added in our baseline.

Comparing proposed models with BASELINE, we see that WINDOW, COMBINED, and FULL models outperform BASELINE in important metrics: Kappa, F1, Recall, but TOPIC yields worse performances than BASELINE. However, the fact that COMBINED outperforms BASELINE, especially with significantly higher Kappa, F1, Recall, and F1:Support, has shown the value of Topic-context features. While Topic-context features alone are not effective, they help improve WINDOW model which supports our hypothesis that Topic-context and Window-context features are complementary aspects of context, and they together obtain better performance.

Comparing our proposed TOPIC, WINDOW,

	BASELINE	TOPIC	WINDOW	COMBINED	FULL
Accuracy	0.885	0.886	0.872	0.885	**0.887**
Kappa	0.245	0.305*	0.306*	**0.342***	0.274*
Macro F1	0.618	0.651*	0.652*	**0.670***	0.634*
Macro Precision	0.680	0.692	0.663	**0.697**	0.693
Macro Recall	0.595	0.628*	0.644*	**0.652***	0.609*
F1:Support	0.937	0.937	0.928*	0.936	**0.938**
F1:Attack	0.300	0.365*	0.376*	**0.404***	0.330*

Table 3: 5×10-fold cross validation performance of Support vs. Attack classification. * indicates significantly different from the baseline ($p < 0.01$).

COMBINED models with each other shows that COMBINED obtains the best performance while TOPIC performs the worst, which reveals that Topic-context feature set is less effective than Window-context set. While FULL model achieves the best Accuracy, Precision, and F1:Non-support, it has lower performance than COMBINED model in important metrics: Kappa, F1, F1:Support. We reason that the noise caused by word pairs and production rules even dominate the effectiveness of Topic-context and Window-context features, which degrades the overall performance.

Overall, by combining TOPIC and WINDOW models, we obtain the best performance. Most notably, we obtain the highest improvement in F1:Support, and have the best balance between Precision and Recall values among all models. These reveal that our contextual features not only dominate generic features like word pairs and production rules, but also are effective to predict minor positive class (i.e., Support).

6.2 Task 2: Support vs. Attack

To evaluate the robustness of our proposed models, we conduct an argumentative relation classification experiment that classifies a relation as Support or Attack. Because this task was not studied in (Stab and Gurevych, 2014b) and the training/test split for Support vs. Not task is not applicable here, we conduct 5×10-fold cross validation. We do not optimize the window-size parameter of the WINDOW model, and use the value 3 as set up before. Average prediction performance of all models are reported in Table 3.

Comparing our proposed models with the baseline shows that all of our proposed models significantly outperform the baseline in important metrics: Kappa, F1, F1:Attack. More notably than in the Support vs. Non-support classification, all of our proposed models predict the minor class (Attack) significantly more effectively than the baseline. The baseline achieves significantly higher F1:Support than WINDOW model. However, F1:Support of the baseline is in a tie with TOPIC, COMBINED, and FULL.

Comparing our proposed models, we see that TOPIC and WINDOW models reveal different behaviors. TOPIC model has significantly higher Precision and F1:Support, and significantly lower Recall and F1:Attack than WINDOW. Moreover, WINDOW model has slightly higher Kappa, F1, but significantly lower Accuracy. These comparisons indicate that Topic-context and Window-context features are equally effective but impact differently to the prediction. The different nature between these two feature sets is clearer than in the prior experiment, as now the classification involves classes that are more semantically different, i.e., Support vs. Attack. We recall that TOPIC model performs worse than WINDOW model in Support vs. Non-support task.

Our FULL model performs significantly worse than all of TOPIC, WINDOW, and COMBINED in Kappa, F1, Recall, and F1:Attack. Along with results from Support vs. Non-support task, this further suggests that word pairs and production rules are less effective and cannot be combined well with our contextual features.

Despite the fact that the Support vs. Attack task (Task 2) has smaller and more imbalanced data than the Support vs. Non-support (Task 1), our proposed contextual features seem to add even more value in Task 2 compared to Task 1. Using Kappa to roughly compare prediction performance across the two tasks, we observe a greater performance improvement from Baseline to Combined model in Task 2 than in Task 1. This is an evidence that our proposed context-aware features

work well even in a more imbalanced with smaller data classification task. The lower performance values of all models in Support vs. Attack than in Support vs. Non-support indirectly suggest that Support vs. Attack classification is a more difficult task. We hypothesize that the difference between support and attack exposes a deeper semantic relation than that between support and no-relation. We plan to extract textual text similarity and textual entailment features (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014) to investigate this hypothesis in our future work.

7 Conclusions and Future Work

In this paper, we have presented *context-aware argumentative relation mining* that makes use of contextual features by exploiting information from topic context and context sentences. We have explored different ways to incorporate our proposed features with baseline features used in a prior study, and obtained insightful results about feature effectiveness. Experimental results show that Topic-context and Window-context features are both effective but impact predictive performance measures differently. In addition, predicting an argumentative relation will benefit most from combining these two set of features as they capture complementary aspects of context to better characterize the argumentation in justification.

The results obtained in this preliminary study are promising and encourage us to explore more directions to enable contextual features. Our next step will investigate uses of topic segmentation to identify context sentences and compare this linguistically-motivated approach to our current window-size heuristic. We plan to follow prior research on graph optimization to refine the argumentation structure and improve argumentative relation prediction. Also, we will apply our context-aware argumentative relation mining to different argument mining corpora to further evaluate its generality.

Acknowledgments

This research is supported by NSF Grant 1122504. We thank the reviewers for their helpful feedback. We also thank Christian Stab for providing us the data split for the first experiment.

References

Or Biran and Owen Rambow. 2011. Identifying Justifications in Written Dialogs by Classifying Text as Argumentative. *International Journal of Semantic Computing*, 5(4):363–381.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland, June. Association for Computational Linguistics.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 18(1):32–39, January.

Elena Cabrio and Serena Villata. 2012. Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 208–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elena Cabrio, Sara Tonelli, and Serena Villata. 2013. From Discourse Analysis to Argumentation Schemes and Back: Relations and Differences. In *Computational Logic in Multi-Agent Systems*, volume 8143 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jianguang Du, Jing Jiang, Liu Yang, Dandan Song, and Lejian Liao. 2014. Shell Miner: Mining Organizational Phrases in Argumentative Texts in Social Media. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 797–802, Washington, DC, USA. IEEE Computer Society.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9:1871–1874, June.

Eirini Florou, Stasinos Konstantopoulos, Antonis Koukourikos, and Pythagoras Karampiperis. 2013. Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social*

Sciences, and Humanities, pages 49–54, Sofia, Bulgaria, August. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, November.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland, June. Association for Computational Linguistics.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context Dependent Claim Detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, Dublin, Ireland, August.

Marco Lippi and Paolo Torroni. 2015. Context-independent Claim Detection for Argument Mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 185–191, Buenos Aires, Argentina. AAAI Press.

Nitin Madnani, Michael Heilman, Joel Tetreault, and Martin Chodorow. 2012. Identifying High-Level Organizational Elements in Argumentative Discourse. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–28, Montrèal, Canada. Association for Computational Linguistics.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic Detection of Arguments in Legal Texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, pages 225–230, New York, NY, USA. ACM.

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

Huy Nguyen and Diane Litman. 2016. Improving argument mining in student essays by learning and exploiting argument indicators versus essay topics. In *Proceedings 29th International FLAIRS Conference*, Key Largo, FL, May.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Joonsuk Park and Claire Cardie. 2014. Identifying Appropriate Support for Propositions in Online User Comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland, June. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2013. From Argument Diagrams to Argumentation Mining in Texts: A Survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, January.

Andreas Peldszus and Manfred Stede. 2015a. Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 938–948, Lisbon, Portugal, September. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2015b. Towards Detecting Counter-considerations in Text. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 104–109, Denver, CO, June. Association for Computational Linguistics.

Andreas Peldszus. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97, Baltimore, Maryland, June. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L08-1093.

Vahed Qazvinian and Dragomir R. Radev. 2010. Identifying Non-explicit Citing Sentences for Citation-based Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 555–564, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diarmuid Ó Séaghdha and Simone Teufel. 2014. Unsupervised learning of rhetorical structure with untopic models. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING-14)*, Dublin, Ireland.

Alan Sergeant. 2013. Automatic argumentation extraction. In *The 10th European Semantic Web*

Conference, pages 656–660, Montpellier, France. Springer.

Christian Stab and Iryna Gurevych. 2014a. Annotating Argument Components and Relations in Persuasive Essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Christian Stab, Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation Mining in Persuasive Essays and Scientific Articles from the Discourse Structure Perspective. In Elena Cabrio, Serena Villata, and Adam Wyner, editors, *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*, pages 40–49, Bertinoro, Italy, July. CEUR-WS.

Simone Teufel and Marc Moens. 2002. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4), December.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Jianxiang Wang and Man Lan. 2015. A Refined End-to-End Discourse Parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China, July. Association for Computational Linguistics.