Unit Segmentation of Argumentative Texts

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth and **Benno Stein**
Bauhaus-Universität Weimar
99423 Weimar, Germany
<first name>.<last name>@uni-weimar.de

Abstract

The segmentation of an argumentative text into argument units and their non-argumentative counterparts is the first step in identifying the argumentative structure of the text. Despite its importance for argument mining, unit segmentation has been approached only sporadically so far. This paper studies the major parameters of unit segmentation systematically. We explore the effectiveness of various features, when capturing words separately, along with their neighbors, or even along with the entire text. Each such context is reflected by one machine learning model that we evaluate within and across three domains of texts. Among the models, our new deep learning approach capturing the entire text turns out best within all domains, with an F-score of up to 88.54. While structural features generalize best across domains, the domain transfer remains hard, which points to major challenges of unit segmentation.

1 Introduction

Argument mining deals with the automatic identification and classification of arguments in a text. It has become an emerging topic of research mainly owing to its many applications, such as writing support tools (Stab and Gurevych, 2014a), intelligent personal assistants (Rinott et al., 2015), and argument search engines (Wachsmuth et al., 2017).

Unit segmentation is often seen as the first task of an argument mining pipeline. It consists in the splitting of a text into its argumentative segments (called *argument units* from here on) and their non-argumentative counterparts. Afterwards, the roles that the argument units play in the argumentative structure of the text as well as the relations between the units are classified. Conceptually, an argument unit may span a clause, a complete sentence, multiple sentences, or something in between. The size of the units depends on the domain of an argumentative text (in terms of topic, genre, or similar), but can also vary within a text. This makes unit segmentation a very challenging task.

As detailed in Section 2, much existing research on argument mining has skipped the segmentation, assuming it to be given. For applications, however, an automatic segmentation is obligatory. Recently, three approaches have been presented that deal with the unit segmentation of persuasive essays: Persing and Ng (2016) rely on handcrafted rules based on the parse tree of a sentence to identify segments; Stab (2017) uses sequence modeling based on sophisticated features to classify the argumentativeness of each single word based on its surrounding words; and Eger et al. (2017) employ a deep learning architecture that uses different features to do the same classification based on the entire essay. So far, however, it is neither clear what the best segmentation approach is, nor how different features and models generalize across domains and genres of argumentative texts.

In this paper, we carry out a systematic study to explore the major parameters of unit segmentation, reflected in the following three research questions:

1. What features are most effective in unit segmentation?
2. What is the best machine learning model to capture the context of a unit that is relevant to segmentation?
3. To what extent do the features and models generalize across domains?

We approach the three questions on and across three existing argumentation corpora, each repre-

The first two authors equally contributed to this paper.

Proceedings of the 4th Workshop on Argument Mining, pages 118–128
Copenhagen, Denmark, September 8, 2017.

senting a different domain (Section 3): the essays corpus of Stab (2017), the editorials corpus of Al-Khatib et al. (2016), and the web discourse corpus of Habernal and Gurevych (2015). All combinations of training and test domain are considered for these corpora, resulting in nine experiments.

Given the corpora, we follow the existing approaches outlined above in tackling unit segmentation as a token-level classification task (Section 4). To capture the context around each token, we analyze different semantic, syntactic, structural, and pragmatic feature types, and we compare three fundamental machine learning techniques based on these features: standard feature-based classification realized as a support vector machine (SVM), sequence modeling realized as linear-chain conditional random field (CRF), and a new deep learning approach realized as a bidirectional long short-term memory (Bi-LSTM). These models correspond to increasingly complex levels of modeling context: The SVM considers only the current token, resulting in an isolated classification for each word. The CRF is additionally able to consider the preceding classifications. The Bi-LSTM, finally, can exploit all words and classifications before and after the current word.

We evaluate all features and models in Section 5. Our results provide clear evidence that the capability of deep learning to model the entire context is beneficial for unit segmentation within domains. The Bi-LSTM achieves the highest effectiveness on each corpus, even outperforming the approach of Stab (2017) on the essays corpus. Across domains, however, all three perform similar and notably drop in effectiveness. Matching intuition, semantic features turn out best to characterize argument units in the in-domain experiments, whereas structural features are more effective across domains. Our findings indicate that the concepts of argument units in the given corpora do not fully match.

Altogether, the contribution of our paper is an extensive analysis of the benefits and limitations of standard approaches to argument unit segmentation. Nevertheless, argument unit segmentation is by far not a solved task yet, which is why we end with a discussion of its major challenges in Section 6, before we finally conclude (Section 7).

2 Related Work

Unit segmentation is a classical segmentation task, that is related to discourse segmentation (Azar, 1999; Green, 2010; Peldszus and Stede, 2013) as for rhetorical structure theory (Mann and Thompson, 1988). Both discourse and argument units are used as building blocks, which are then hierarchically connected to represent the structure of the text. However, argument units are closer to classical logic, with each unit representing a proposition within the author's argumentation.

Much existing work on argument mining skips the segmentation, assuming segments to be given. Such research mainly discusses the detection of sentences that contain argument units (Teufel, 1999; Palau and Moens, 2009; Mochales and Moens, 2011; Rooney et al., 2012), the classification of the given segments into argumentative and non-argumentative classes (Stab and Gurevych, 2014b), or the classification of relations between given units (Stab and Gurevych, 2014b; Peldszus, 2014; Peldszus and Stede, 2015).

A few publications address problems closely related to unit segmentation. Madnani et al. (2012) identify non-argumentative segments, but they do not segment the argumentative parts. Levy et al. (2014), on the other hand, try to detect segments that are argumentatively related to specific topics. However, they do not segment the whole text.

A unit segmentation algorithm has been already applied by Al-Khatib et al. (2016) in the creation of the editorials corpus analyzed in this paper. The authors developed a rule-based algorithm to automatically pre-segment the corpus texts before the manual annotation. The algorithm was tuned to rather split segments in cases of doubt. During the annotation, annotators were then asked to correct the segmentation by merging incorrectly split segments. The authors argue that—even with a simple algorithm—this approach simplifies the annotation process and makes evaluating inter-annotator agreement more intuitive.

In the few publications that fully address unit segmentation, a detailed analysis of features and models is missing. Previous work employs rule-based identification (Persing and Ng, 2016), feature-based classification (Lawrence et al., 2014), conditional random fields (Sardianos et al., 2015; Stab, 2017), or deep neural networks (Eger et al., 2017). Especially the most recent approaches by Stab and Eger et al. rely on sophisticated structural, syntactical, and lexical features. Eger et al. even report that they beat the human agreement in unit segmentation on the one corpus they consider, but the paper

does not clarify which linguistic cues are most helpful to reach this performance. To remedy this, we also employ a deep neural network based on Bi-LSTMs, but we perform a detailed comparison of models and feature sets.

Previous work trains and tests unit segmentation algorithms on one single corpus. A frequent choice is one of the two versions of the Argument Annotated Essay Corpus (Stab and Gurevych, 2014a; Stab, 2017), which is studied by Persing and Ng (2016), Eger et al. (2017), Stab (2017) himself, and also by us. However, for a unit segmentation algorithm to be integrated into applications, it has to work robustly also for new texts from other domains. This paper therefor extends the discussion of unit segmentation in this direction.

3 Data

This study uses three different corpora to evaluate the models that we developed to segment argument units. The corpora resprented different domains, particularly in terms of genre. We detail each corpus below, give an overview in Table 1, and provide example excerpts in Figure 1.

Essays The Argument Annotated Essays Corpus (Stab and Gurevych, 2014a; Stab, 2017) includes 402 persuasive essays from `essayforum.com` written by students. All essays have been segmented by three expert annotators into three types of argument units (major claims, claims, and premises) and non-argumentative parts. Each argument unit covers an entire sentence or less. The essays are on average 359.5 tokens long with 70% of tokens being part of an argument unit.[1] We employ the test-training split provided by the authors.

Editorials The Webis-Editorials-16 corpus (Al-Khatib et al., 2016) consists of 300 news editorials from the three online news portals Al Jazeera, Fox News, and The Guardian. Prior to the annotation process, the corpus was automatically presegmented based on clauses. After that, three annotators performed the final segmentation by merging segments and distinguishing argument units of six types (common ground, assumption, anecdote, testimony, statistics, and other) from non-argumentative parts. The annotation guidelines define a unit as a segment that spans a proposition (or two or more interwoven propositions) stated by the

[1]The percentage of tokens that are part of an argument unit is calculated from Table 1 as (Arg-B + Arg-I)/Total.

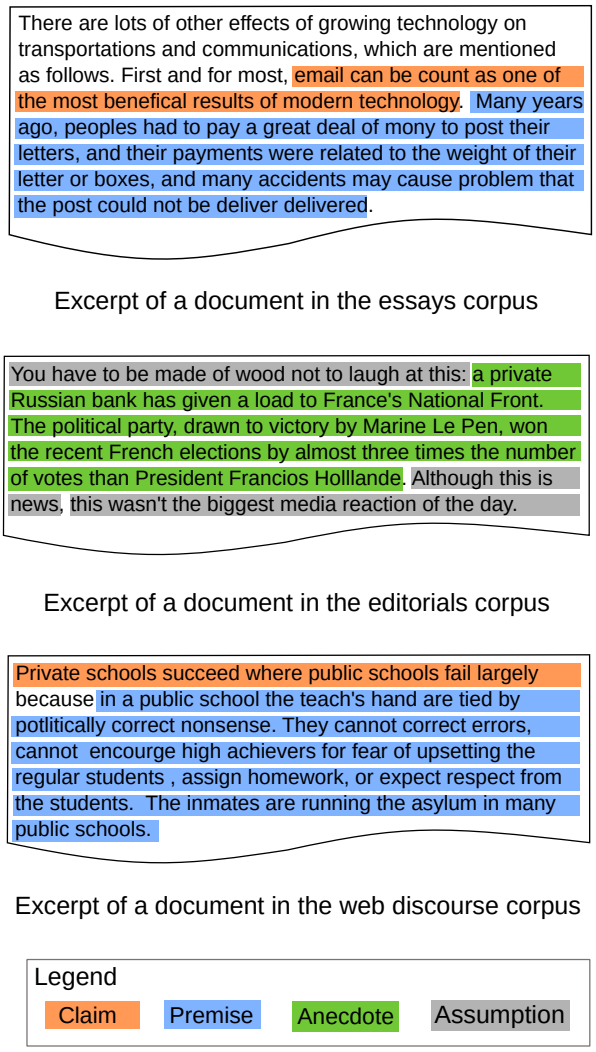

Figure 1: Excerpts of three documents for the essays, editorials and web discourse corpus. Each excerpt is highlighted with argument units as annotated in the original corpus

author to discuss, directly or indirectly, his or her thesis. This corpus contains the longest documents with an average of 957.9 tokens. The editorials are mainly argumentative, with 92% of the tokens in the corpus being part of an argument unit. We employ the provided training-test split.

Web Discourse The Argument Annotated User-Generated Web Discourse corpus (Habernal and Gurevych, 2016) contains 340 user comments, forum posts, blogs, and newspaper articles. Each of these is annotated according to a modified version of Toulmin's model (Toulmin, 1958). In the corpus, argument units belong to one of five types (premise, claim, rebuttal, refutation and backing) and can be arbitrary text spans. Because of the

Corpus	Part	# Documents	Number of tokens				
			Arg-B	Arg-I	Arg-O	Total	Average
Essays	Training	322	4,823	75,621	35,323	115,767	359.5
	Test	80	1,266	18,790	8,699	28,755	359.4
	Total	402	6,089	94,411	44,022	144,522	359.5
Editorials	Training	240	11,323	202,279	17,227	230,829	961.8
	Test	60	2,811	49,102	4,622	56,535	942.3
	Total	300	14,234	251,381	21,849	287,364	957.9
Web Discourse	Training	272	905	32,093	36,731	69,729	256.4
	Test	68	224	7,949	8,083	16,256	239.1
	Total	340	1,129	40,042	44,814	85,985	252.9

Table 1: Number of documents, tokens per class, and average tokens per document per corpus and part.

latter, the units are on average much longer than in the other two corpora: 36.5 tokens compared to 16.5 tokens (essays) and 18.7 tokens (editorials).[2] The complete documents are relatively short though (252.9 tokens on average), and they contain many non-argumentative parts: only 48% of the tokens are part of an argument unit. Since the authors do not provide any split, we randomly split the corpus into a training set (80%) and test set (20%), similar to the other corpora.

The three corpora vary in terms of how arguments are actually annotated in the contained documents. Following Stab (2017), we converted all documents into the BIO format, where each token is labeled according to the position in the segment that it belongs to as *Arg-B* (the first token of an argument unit), *Arg-I* (any other token of an argument unit), or *Arg-O* (not in an argument unit).

4 Method

This paper explores the effectiveness of semantic, syntactic, structural, and pragmatic features when capturing tokens separately, along with their neighbors, or along with the entire text. In line with recent work (see Section 2), we address unit segmentation as a token labeling problem. In the following, we detail each set of features as well as the three machine learning models that we employ. Each model reflects one of the outlined contexts used to classify the tokens. To demonstrate the strengths and weaknesses of the models, we encode the features as analog as possible in each model. However, some variations are necessary due to differences in the way the models utilize the features.

[2] Average length of argument units is calculated from Table 1 as (Arg-B + Arg-I)/Arg-B

4.1 Features

For every token, we extract the following semantic, syntactic, structural and pragmatic features.

Semantic Features Semantic features capture the meaning of tokens. This work employs the simple but often effective way of representing meaning by using the occurrence of each token as a feature (bag-of-words). We also tested word embeddings (Pennington et al., 2014) as semantic features, but found that they performed worse for all models introduced below except for the Bi-LSTM.

Syntactic Features The syntactic features that we employ capture the role of a token in a sentence or argument unit. We resort to standard part-of-speech (POS) tags as produced by the Stanford tagger (Toutanova et al., 2003) for this feature set.

Structural Features Structural features capture the congruence of argument units with sentences, clauses, or phrases. We employ the Stanford parser (Klein and Manning, 2003) to identify sentences, clauses, and phrases in the text and represent them with token labels. In particular, we use one feature for each token and structural level (sentence, clause, phrase), capturing whether the token is at the beginning, within, or at the end of such a structural span, respectively.

Pragmatic Features Pragmatic features capture the effects the author of a text intended to have on the reader. We use lists of discourse markers compiled from the Penn Discourse Treebank (Prasad et al., 2008) and from (Stab, 2017) to identify such markers in the text. The latter have been specifically created for detecting argument units. For each token and discourse marker, we use five binary fea-

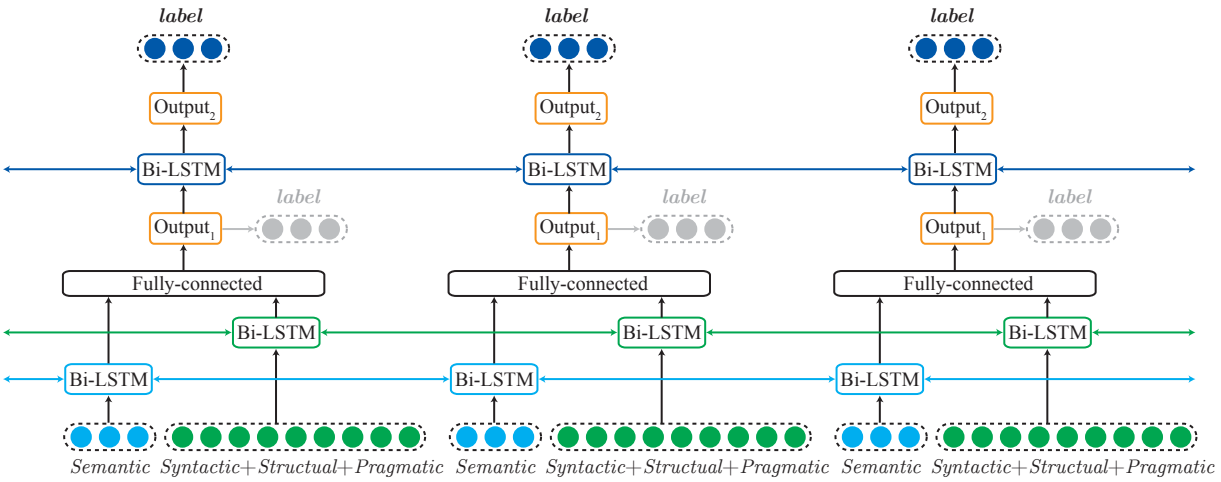

Figure 2: The neural network structure used in our paper with the input feature vectors for three tokens at the bottom. The labels by $Ouput_1$ are estimated without considering label dependency and are not used; instead we report the results for $Output_2$, which considers this dependency.

tures that are 1 iff. the token is before the marker, the beginning of the marker, inside a multi-token marker, the last token of a multi-token marker, or after the marker in the sentence, respectively.

4.2 Models

We make use of three common machine learning models in order to capture an increasing amount of context for the token labeling: a support vector machine (SVM), a conditional random field (CRF), and a bidirectional long short-term memory (Bi-LSTM). To provide a comparison to results from related work, we reimplemented the method of Stab (2017) and use it as a baseline.

Reimplementation The approach of Stab (2017) is based on a CRF sequence model (Lafferty et al., 2001). It has been specifically developed for the segmentation given in the essays corpus. Since the license of the original implementation prohibited the author from giving us access to the code, we fully reimplemented the approach.

Analog to Stab (2017), we employ the CRFSuite (Okazaki, 2007) with the averaged perceptron method (Collins, 2002). For the reimplementation, we use the exact feature sets described by Stab (2017): Structural, Syntactic, LexSyn and Prob. Our reimplementation achieves an F-score of 82.7, which is slightly worse than the value reported by Stab (2017) for unit segmentation (86.7). We attribute this difference to implementation details in the employed features.

SVM We employ a linear SVM model in terms of a standard feature-based classifier that labels each consecutive token independently, disregarding the token's context. In other words, features of neighboring tokens are not considered by the SVM. Accordingly, this model does not capture the transition between labels, as well.

CRF We implement a CRF sequence model to capture the context around the token for labeling the token. For labeling, the linear-chain CRF that we use considers the labels and features of the surrounding tokens within a certain window, which we chose to be of size 5 for our experiments. We use the same framework and method as for the reimplementation.

Since CRFs explicitly capture the local context of a token, we simplify the pragmatic features for this model and use only binary features for whether the token is at the beginning, inside, at the end, or outside of a discourse marker.

Bi-LSTM Finally, we also build a Bi-LSTM neural network to capture the entire text as context. The architecture of the model is illustrated in Figure 2 and further explained below.

Compared to the CRF, the Bi-LSTM model does not utilize a window while classifying a token but considers the whole the text at once. Instead of using the tokens directly as semantic features, we use the word embedding of the tokens (Pennington et al., 2014), as this is common for neural networks. In particular, we use the standard pre-trained em-

		Test on Essays			Test on Editorials			Test on Web Discourse		
Features	**Models**	Essays	Editorials	Web Dis.	Essays	Editorials	Web Dis.	Essays	Editorials	Web Dis.
Semantic	SVM	53.42	40.89	28.89	50.00	53.96	16.20	31.71	26.58	33.34
	CRF	76.56	53.06	26.31	66.30	78.90	8.48	37.51	37.25	42.53
	Bi-LSTM	87.91	**57.11**	36.00	60.70	81.56	24.63	**41.29**	36.44	**54.98**
Syntactic	SVM	49.66	36.14	26.45	49.98	51.36	14.32	28.44	25.33	31.93
	CRF	66.79	48.40	15.48	68.30	76.74	5.05	34.73	38.13	24.25
	Bi-LSTM	83.10	55.70	21.65	64.92	80.35	15.28	36.58	37.40	43.02
Structural	SVM	41.19	36.14	26.45	49.53	77.71	5.96	27.97	37.98	27.52
	CRF	60.12	48.41	15.48	68.96	77.55	5.68	34.64	**38.30**	22.51
	Bi-LSTM	69.77	48.63	**41.19**	61.54	79.62	**38.08**	35.46	37.75	39.51
Pragmatic	SVM	38.75	28.65	30.09	31.33	33.02	22.38	30.85	22.24	35.59
	CRF	40.15	31.66	15.48	37.06	40.20	5.02	24.30	30.30	23.70
	Bi-LSTM	76.47	54.72	15.24	57.66	75.31	5.24	34.88	36.68	22.76
All	SVM	61.40	50.88	31.26	58.84	79.89	22.55	39.14	37.42	42.76
	CRF	79.15	52.50	21.74	**69.80**	81.97	8.00	37.09	37.63	37.74
	Bi-LSTM	**88.54**	**57.11**	36.97	60.69	**84.11**	20.85	39.78	36.56	54.51
Reimplementation		82.70	52.00	20.00	67.00	78.00	6.00	31.66	37.30	49.00

Table 2: The in-domain (gray background) and cross-domain macro F-scores on each test (first header row) after training on one of the training sets (second header row). Each row lists the results of one of the three models (SVM, CRF, and Bi-LSTM) using one of the four feature types (semantic, syntactic, structural, and pragmatic) in isolation or their combination (all). For each column, the highest value is marked in bold. The bottom line shows the F-scores of our reimplementation of the approach of Stab (2017).

bedding of Pennington et al. (2014), which has a dimensionality of 300. For the other feature sets, we concatenate all the boolean features described in the previous section into a sparse feature vector (more precisely, a one-hot vector).

The architecture in Figure 2 should be viewed from bottom to top. We first feed the features into bidirectional LSTMs (Schuster and Paliwal, 1997). Next, we feed the semantic features into a separate Bi-LSTM in order to be able to use a different kernel for the dense feature vector of the semantic features than for the one-hot vectors. The output of the two Bi-LSTM layers is then concatenated and fed into a fully-connected layer. To model label dependencies, we add another Bi-LSTM and another output layer. Both output layers are softmax layers, and they are trained to fit the labels of tokens. We process only the result of the second output layer, though. As we will see in Section 5, the second output layer does indeed better capture the sequential relationship of labels.

5 Experiments

Using the three corpora detailed in Section 3, we conduct in-domain and cross-domain experiments to answer the three research questions from Section 1. In each experiment, we use the training set of one corpus for training the model and the test set of the same or another corpus for evaluating the model. In all cases, we test all four considered feature sets both in isolation and in combination. We report the macro F-score as an evaluation measure, since this allows for a comparison to related work and since we consider all three classes (*Arg-B*, *Arg-I*, and *Arg-O*) to be equally important.

Table 2 lists the macro F-scores of all combinations of features and models as well as of our reimplementation of the approach of Stab (2017) for all combinations of training and test set.

5.1 Comparison to Previous Work

To put our results into context, we also imitate the experiment setting of Stab (2017). For this purpose, we randomly split the test set of the essays corpus into five equally-sized subsets and use the student's t-test to compare the F-scores of our methods on each subset with the result of Stab (2017). We find that our best-performing method, the Bi-LSTM using all features, achieves a significantly better F-score (88.54 versus 86.70) with p-value < 0.001.

		Prediction								
	Label	B-B	B-I	B-O	I-B	I-I	I-O	O-B	O-I	O-O
Gold	B-B	**0**	0	0	0	0	0	0	0	0
	B-I	1	**956**	11	0	152	0	0	0	0
	B-O	0	0	**0**	0	0	0	0	0	0
	I-B	0	0	0	**0**	0	0	0	0	0
	I-I	0	71	0	4	**16363**	78	59	77	872
	I-O	0	0	0	0	83	**1109**	0	0	74
	O-B	0	4	0	10	131	0	**958**	17	144
	O-I	0	0	0	0	0	0	0	**0**	0
	O-O	0	129	7	1	1285	157	139	87	**5550**

Table 3: Confusion matrix opposing the number of gold BIO labels of pairs of consecutive tokens in the essays corpus to those predicted by our best-performing method, the Bi-LSTM using all features. The correct predictions (on the diagonal) are marked in bold.

Furthermore, although the results of our reimplementation of the approach of Stab (2017) are lower than those reported by the author, our own CRF approach performs comparably well in almost all cases using simple linguistic features.

5.2 Improvement by Second Output Layer

A side effect of predicting the BIO label of each token separately is that two consecutive tokens can be labeled as *Arg-O* and *Arg-I*. This is not reasonable, since it corresponds to a unit without beginning. Without the second output layer $Output_2$, our neural network method produced about 400 such pairs. However, when we added the layer, the number dropped by half to 200 pairs. While the effect on the F-score is small, using the second output layer therefore produces more comprehensible results. We thus only report the results with $Output_2$.

5.3 Error Analysis

To learn about the behavior of our best-performing Bi-LSTM model, we carried out an error analysis. Table 3 presents the confusion matrix of the gold BIO label pairs and the predicted pairs on the essays corpus. While it is not possible to discuss all errors here, we observed a few typical cases, as discussed in the following.

In particular, some wrong predictions result from cases where the Bi-LSTM combines several units into one. For instance, the two units in *"... [the criminal is repeated second time]; also, [it is regarded as the "legalized revenge"...]"* are predicted as one unit. This produces errors of the types *(I-O, I-I)*, *(O-B, I-I)*, and *(O-O, I-I)* (gold vs. prediction). Conversely, the Bi-LSTM also sometimes chops one unit into several units. For instance, the unit *"Crimes kill someone which is illegal; nevertheless, the government use law to punish them..."* is chopped into *"[Crimes kill someone which is illegal]"* and *"[the government use law to punish them...]"*. This will create *(I-I, I-O)*, *(I-I, O-O)*, and *(I-I, O-B)* errors, despite noticing that it may also make sense for some annotators.

Finally, some *(I-O, I-I)* errors occurred a number of times, because of the delimiter of units (such as ",", "." or ";") were not included in the gold data but predicted as being part of it by our Bi-LSTM.

6 Discussion

Given our experimental results, we come back to the three research questions we initially raised, and then turn our head to ongoing research.

6.1 Major Parameters of Unit Segmentation

Our study aims to provide insights into three major parameters of unit segmentation: features, models, and domains. Each of them is reflected in one of our guiding research questions from Section 1.

Research Question 1 *What features are most effective in unit segmentation?*

According to the results of the in-domain experiments, the semantic features are the most effective. The models employing these features, achieve the highest F-scores, except for the SVM on *editorials*, where structural features perform better. However, there is no feature type that dominates the cross-domain experiments. At least, the structural features seem rather robust when the training and test sets are from different domains.

		Sentence			Clause			Phrase		
Corpus	**Label**	B	I	E	B	I	E	B	I	E
Essays	Arg-B	0.30	-0.19	-0.05	0.23	-0.13	-0.06	0.04	0.04	-0.08
	Arg-I	-0.30	**0.44**	-0.30	-0.23	0.34	-0.22	0.04	0.03	-0.08
	Arg-O	0.18	-0.37	0.33	0.14	-0.29	0.25	-0.06	-0.04	0.11
Editorials	Arg-B	**0.75**	**-0.51**	-0.05	**0.57**	-0.38	-0.07	0.15	-0.09	-0.09
	Arg-I	**-0.53**	**0.74**	**0.48**	**-0.44**	**0.58**	-0.33	0.02	0.12	0.11
	Arg-O	0.05	**-0.50**	**0.64**	0.09	**-0.41**	**0.47**	-0.10	-0.09	0.21
Web Discourse	Arg-B	**0.48**	-0.33	-0.03	0.32	-0.22	-0.04	0.10	-0.06	-0.05
	Arg-I	-0.12	0.09	0.00	-0.09	0.07	0.00	-0.02	0.01	0.01
	Arg-O	0.18	0.01	-0.01	0.01	0.01	0.08	0.00	0.00	0.00

Table 4: Pearson correlation between argument unit boundaries and structural features. Values range from -1.00 (total negative correlation) to 1.00 (total positive correlation). Absolute values above or equal to 0.40 can be seen as moderately correlated and are marked in bold.

While the results of the semantic features across *essays* and *editorials* — two domains that are comparably similar — remain high, the performance of the models employing them dramatically drop when tested on *web discourse* after training on either of the other. The intuitive explanation for this decrease in the domain transfer is that important content words are domain-specific. Thus, the learned knowledge from one domain cannot be transferred to other domains directly. In contrast, structural features capture more general properties of argumentative text, which is why we can use them more reliably in other domains.

As shown in Table 4, the sentence, clause, and phrase boundaries correlate with the boundaries of argument units. Especially in the editorials corpus, the boundaries of sentences and clauses show high Pearson coefficients. This reveals why we can still achieve reasonable performance when the training and test set differ considerably.

Research Question 2 *What is the best machine learning model to capture the context of a unit that is relevant to segmentation?*

Comparing the different models, the SVM performs worst in most experiments. This is not surprising, because the SVM model we used utilizes local information only. In a few cases, however, the SVM performed better than the other models, e.g., when evaluating pragmatic features on *essays* that were learned on *web discourse*. One reason may be that such features rather have local relevance. As a matter of fact, adding knowledge from previous and preceding tokens will add noise to a model rather than being beneficial.

Overall, the models employing sequential features turn out stronger. Among them, the Bi-LSTM achieves the best results in most cases regardless of the domain or the features. This suggests that context information from the tokens around a token to be classified is generally useful. In addition, using neural networks seems to be a better choice to encode those features.

Another advantage of using a Bi-LSTM is that this model can utilize all features related to tokens from the beginning to the end of the document. This allows the Bi-LSTM to capture long-distance dependencies. For a CRF, such dependencies are hard to encode, requiring to increase the complexity of the model dramatically and thus making the problem intractable.

Research Question 3 *To what extent do the features and models generalize across domains?*

From the results and the previous discussion, we conclude that our structural features (capturing the boundaries of phrases, clauses, and sentences) and the Bi-LSTM model are the most domain-robust. Other features, especially the semantic ones tend to be more domain-dependent. The ability to model long-distance dependencies and a more advanced feature encoding indicate why the Bi-LSTM apparently learns more general, less domain-specific features of the given argumentative texts.

6.2 Major Challenges of Unit Segmentation

The drastic effectiveness loss in the domain transfer suggests that the notion of an argument unit is not entirely the same across argumentative text corpora. This hypothesis is supported by the high variance in

the size of argument units, ranging from clause-like segments (Al-Khatib et al., 2016) to partly multiple sentences (Rinott et al., 2015). At the same time, it seems reasonable to assume that there is a common concept behind argument units that connects their different notions and that distinguishes argument units from other types of segments. Under this assumption, a general question arises that we see as fundamental in research on unit segmentation:

Open Question about Argument Units *What makes argument units different from syntactic and discourse units, and at what point do they deviate?*

The difference between argument units and elementary discourse units is discussed by Stede et al. (2016). The authors claim that the boundaries of the more coarse-grained argument units clearly are also boundaries of discourse units. While this may be the case in their corpus as a result of their annotation scheme, no reason is given why the claim should generally be true. Accordingly, for other corpora such as the essays corpus studied in this paper, the claim simply does not hold.

In principle, it is possible to more generally study the raised question based on a matching of argument units with the syntactic and/or discourse units in different datasets. A generally satisfying answer might not exist, though, because we expect the segmentation into argument units to be task-specific to some extent. Similar observations have been made for discourse units (Taboada and Mann, 2006). In case of argument units, some annotations, for example, model the hierarchical structure of a text primarily (Stab, 2017), whereas others aim to capture self-contained evidence (Rinott et al., 2015). Even for a given task, however, unit segmentation remains challenging, though, as underlined by the limited effectiveness we observed in some experiments. As a result, the notion of an argument unit is a topic of ongoing discussion in the community. This brings up another question:

Open Question in Unit Segmentation *What knowledge is needed to effectively perform unit segmentation?*

In particular, it has been discussed controversially in the community as to whether unit segmentation should actually be tackled as the first step of argument mining. When doing so, no knowledge about the main claims of an argumentation, the applied reasoning, and similar is given, making the feasibility of distinguishing argumentative from non-argumentative parts doubtful. Of course, other orderings might lead to analog problems, which would then suggest to jointly approach the different steps. We plan to explore the best ordering and decomposition of mining steps in future work.

7 Conclusion

Most existing research on argument mining either ignores the task of argument unit segmentation, assuming the units to be given, or considers an argument unit to simply span exactly a sentence or a clause (Teufel, 1999; Palau and Moens, 2009; Mochales and Moens, 2011; Rooney et al., 2012). Recently, the task of argument unit segmentation was tackled on persuasive student essays by casting the problem as a sequence labeling task, classifying each token as being either at the beginning, inside, or outside an argument unit (Stab, 2017; Eger et al., 2017). Both approaches perform comparably well while employing different sequential models and different feature types: Stab (2017) uses local linguistic features whereas Eger et al. (2017) capture the global semantic and argumentative context.

In this work, we adopt the approach to frame argument unit segmentation as a sequence labeling task. We conduct a systematic comparison of three machine learning models that encode the context and the linguistic features of a token differently. Among these, our new Bi-LSTM neural network model utilizes structural, syntactic, lexical and pragmatic features, and it captures long-distance dependencies for argument unit segmentation. In in-domain experiments and cross-domain experiments on three different corpora, we study what model and feature set perform best.

Our experiments show that structural and semantic features are the most effective for argument unit segmentation across domains, while semantic features are the best for detecting the boundaries of argumentative units within domains. We also find that a sequential model capturing a wider context (i.e., our Bi-LSTM) tends to perform better within and across domains. Nevertheless, the results reported in Section 5 show the insufficiency of the employed linguistic features and machine learning models for a domain-robust argument unit segmentation. We therefor conclude that further research is needed in order to clarify the difference between argument units and other types of units as well as to find out what knowledge is best to segment argumentative texts into these units.

References

Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A News Editorial Corpus for Mining Argumentation Strategies. In *Proceedings of the 26th International Conference on Computational Linguistics*.

M. Azar. 1999. Argumentative Text as Rhetorical Structure: An Application of Rhetorical Structure Theory. *Argumentation* 13:97–114.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, volume 10, pages 1–8.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* In Press.

Nancy L. Green. 2010. Representation of Argumentation in Text with Rhetorical Structure Theory. *Argumentation* 24:181–196.

Ivan Habernal and Iryna Gurevych. 2015. Exploiting Debate Portals for Semi-Supervised Argumentation Mining in User-Generated Web Discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2127–2137.

Ivan Habernal and Iryna Gurevych. 2016. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics* .

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

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional Random Fields: Probabilistic models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. volume 1, pages 282–289.

John Lawrence, Chris Reed, Colin Allen, Simon McAlister, and Andrew Ravenscroft. 2014. Mining Arguments From 19th Century Philosophical Texts Using Topic Based Modelling. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, Baltimore, Maryland.

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*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland.

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*. Association for Computational Linguistics, Stroudsburg, Pennsylvania.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text - Interdisciplinary Journal for the Study of Discourse* 8.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation Mining. *Artificial Intelligence and Law* 19(1):1–22.

Naoaki Okazaki. 2007. CRFsuite: A Fast Implementation of Conditional Random Fields (CRFs) .

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*. ACM, pages 98–107.

Andreas Peldszus. 2014. Towards Segment-based Recognition of Argumentation Structure in Short Texts. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, Baltimore, Maryland.

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

Andreas Peldszus and Manfred Stede. 2015. Joint prediction in MST-style Discourse Parsing for Argumentation Mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Isaac Persing and Vincent Ng. 2016. End-to-End Argumentation Mining in Student Essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 1384–1394. https://doi.org/10.18653/v1/N16-1164.

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie L Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, M. Mitesh Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show Me Your Evidence — An Automatic Method for Context Dependent Evidence Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 440–450. https://doi.org/10.18653/v1/D15-1050.

Niall Rooney, Hui Wang, and Fiona Browne. 2012. Applying Kernel Methods to Argumentation Mining. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*.

Christos Sardianos, Ioannis Manousos Katakis Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument Extraction from News. In *Proceedings of the Second Workshop on Argumentation Mining*. Association for Computational Linguistics, Denver, Colorado.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

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

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*. Association for Computational Linguistics, pages 46–56. https://doi.org/10.3115/v1/D14-1006.

Christian Matthias Edwin Stab. 2017. *Argumentative Writing Support by Means of Natural Language Processing*. Ph.D. thesis, Technische Universität Darmstadt.

Manfred Stede, Stergos Afantenos, Andreas Peldszus, Nicholas Asher, and Jérémy Perret. 2016. Parallel Discourse Annotations on a Corpus of Short Texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Maite Taboada and William C. Mann. 2006. Rhetorical Structure Theory: Looking Back and Moving Ahead. *Discourse Studies* 8(3):423–459.

Simone Teufel. 1999. *Argumentative Zoning: Information Extraction from Scientific Text*. Ph.D. thesis, University of Edinburgh.

Stephen Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, volume 1, pages 173–180.

Henning Wachsmuth, Benno Stein, and Yamen Ajjour. 2017. "PageRank" for Argument Relevance. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. volume 1, pages 1117–1127.