Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1073–1080
Marseille, 11–16 May 2020

Joint Learning of Syntactic Features helps Discourse Segmentation

Takshak Desai, Parag Dakle, Dan I. Moldovan
Department of Computer Science
The University of Texas at Dallas
{takshak.desai, paragpravin.dakle, moldovan}@utdallas.edu

Abstract

This paper describes an accurate framework for carrying out multi-lingual discourse segmentation with BERT (Devlin et al., 2019). The model is trained to identify segments by casting the problem as a token classification problem and jointly learning syntactic features like part-of-speech tags and dependency relations. This leads to significant improvements in performance. Experiments are performed in different languages, such as English, Dutch, German, Portuguese Brazilian and Basque to highlight the cross-lingual effectiveness of the segmenter. In particular, the model achieves a state-of-the-art F-score of 96.7 for the RST-DT corpus (Carlson et al., 2003) improving on the previous best model by 7.2%. Additionally, a qualitative explanation is provided for how proposed changes contribute to model performance by analyzing errors made on the test data.

Keywords: Discourse Segmentation; BERT; Multi-task learning

1. Introduction

Discourse Segmentation refers to the task of fragmenting a document into minimal disjoint chunks of text called Elementary Discourse Units (EDUs). In the context of Rhetorical Structure Theory (Mann and Thompson, 1988) or RST, EDUs form the nodes of a discourse tree; while relations between EDUs form arcs or edges between nodes. As a motivating example, consider the discourse tree given in Figure 1. EDUs labeled 1, 2 and 3 form nodes of the tree; and arcs are labeled with ATTRIBUTION (used to indicate instances of reported speech) and PURPOSE relations. This example was taken from the RST-DT corpus (Carlson et al., 2003) and the tree was constructed using the tool provided by Gessler et al. (2019).

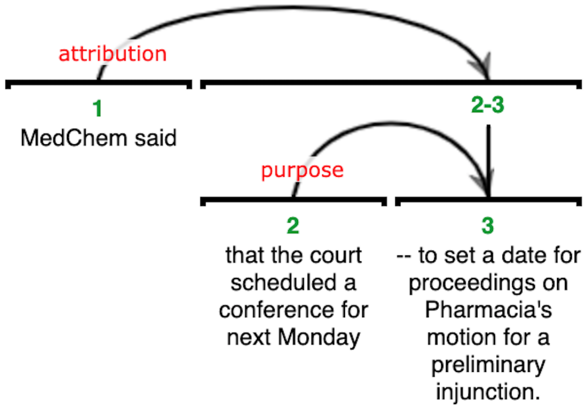
attribution
1
MedChem said
2-3
purpose
2
that the court scheduled a conference for next Monday
3
-- to set a date for proceedings on Pharmacia's motion for a preliminary injunction.

Figure 1: Discourse tree for a portion of text in `wsj_2336`: EDUs form the nodes of the tree, and arcs represents relations.

Discourse segmentation is considered a challenging problem for several reasons. First, the boundary between syntax and discourse is blurry (Carlson and Marcu, 2001). While the clause is considered a basic EDU, segment boundaries are often determined using lexical and syntactic clues. Additionally, the amount of annotated data available is minimal and this makes training of data-hungry models like neural networks difficult.

To assuage the problem of data insufficiency, syntax-free models that leverage pre-trained representations (Wang et al., 2018; Muller et al., 2019) from sentence encoders like ElMo and BERT were proposed: these models achieved very good results for the segmentation task. Likewise, the use of syntactic features such as part-of-speech tags and parse tree features helped achieve better results (Braud et al., 2017; Lin et al., 2019). In fact, the latter achieved state-of-the-art results using pointer networks (Vinyals et al., 2015) with parse tree features.

In this paper, we propose a few changes that leverage BERT's (Devlin et al., 2019) structure in performing segmentation. Our main contributions are:

1. We cast discourse segmentation as a token classification problem, as opposed to sequence tagging. This allows BERT to attend to one token at a time and not the entire sequence, thereby making better decisions.
2. We suggest a simple multi-task learning approach that uses the intermediate layers of BERT to carry out part-of-speech tag prediction, and dependency relation classification. This improves model performance, particularly for languages other than English.
3. Experiments are performed for different languages to demonstrate the cross-lingual effectiveness of our framework. We use multilingual BERT (abbreviated as `bert-multilingual-base`[1]) as our sentence encoder.
4. We also provide a qualitative explanation for what worked in our favour via error analysis. This provides deeper insights into the model's behaviour and explains why we achieved better results[2].

[1] https://github.com/google-research/bert

[2] The code used to carry out the experiments reported in this paper is available at https://www.github.com/takshakpdesai/discourse-segmenter.

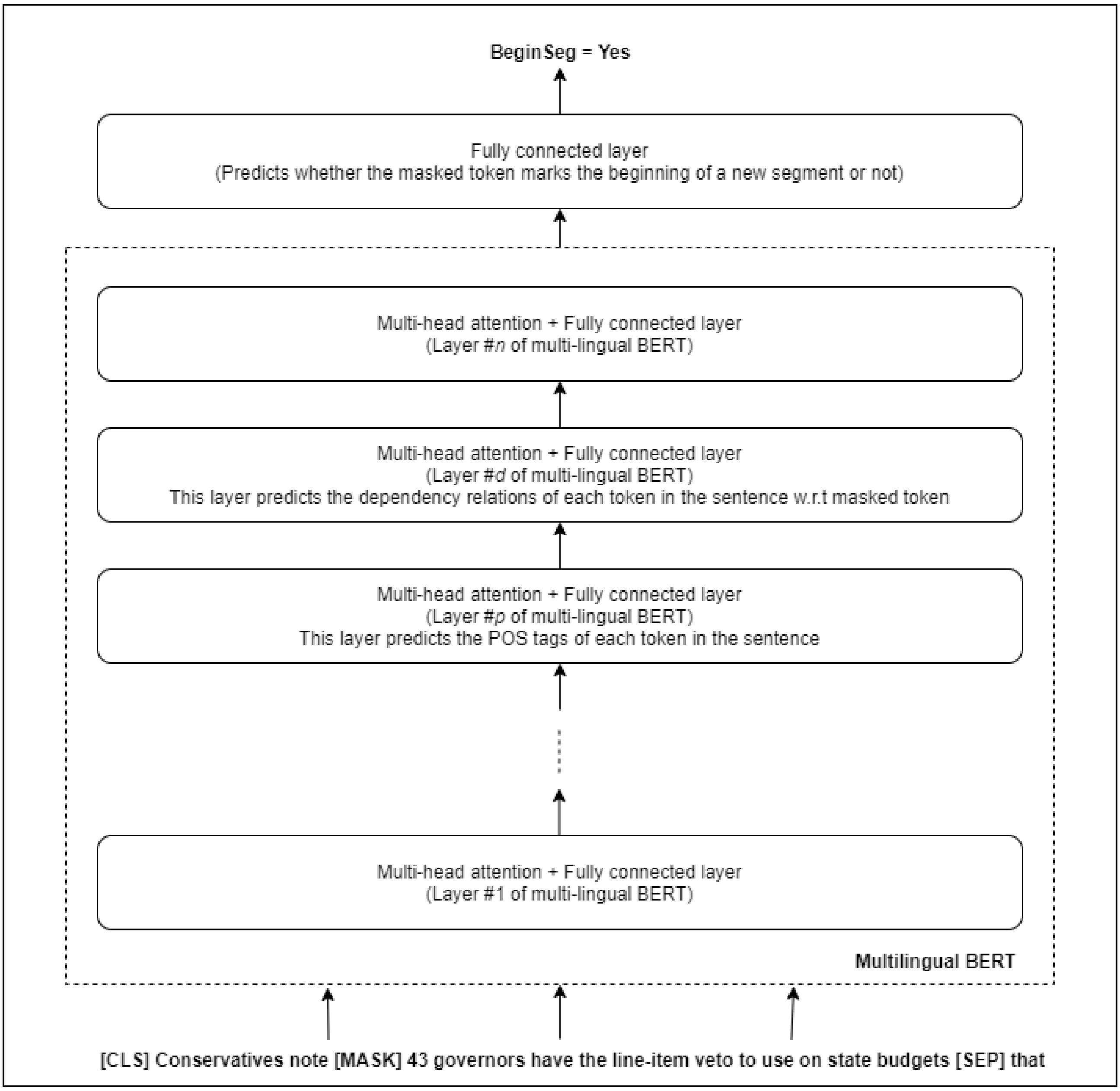
BeginSeg = Yes
Fully connected layer
(Predicts whether the masked token marks the beginning of a new segment or not)
Multi-head attention + Fully connected layer
(Layer #n of multi-lingual BERT)
Multi-head attention + Fully connected layer
(Layer #d of multi-lingual BERT)
This layer predicts the dependency relations of each token in the sentence w.r.t masked token
Multi-head attention + Fully connected layer
(Layer #p of multi-lingual BERT)
This layer predicts the POS tags of each token in the sentence
Multi-head attention + Fully connected layer
(Layer #1 of multi-lingual BERT)
Multilingual BERT
[CLS] Conservatives note [MASK] 43 governors have the line-item veto to use on state budgets [SEP] that

Figure 2: Model Architecture: The model attempts to classify if the masked word ‘that’ represents the beginning of a new segment or not. The intermediate layers carry out feature prediction i.e. the POS tags and dependency relations of all tokens with respect to masked token in the sentence.

2. Proposed Solution

Figure 2 provides a logical view of our model and its components. Subsequent sub-sections describe the model and related modifications/tasks.

2.1. Multilingual BERT encoder

At the heart of our model lies BERT, a powerful language model (Devlin et al., 2019) that provides universal sentence representations. Using BERT offers two advantages. First, it can effectively capture syntactic, semantic and positional dependencies between tokens and/or sub-tokens in a sentence; leaving little to no room for feature engineering. Second, it has been trained on a very large corpus: this allows us to fine-tune pre-trained BERT models on downstream tasks, especially when training data is minimal. This allows us to get away with the data insufficiency problem since the size of training corpora is small.

To work with languages other than English, Devlin et al. (2019) released multilingual BERT: a single language model pre-trained from multi-lingual corpora in 104 languages; using a shared multi-lingual vocabulary. Despite being a shared language model, which could raise concerns about its cross-lingual effectiveness, multilingual BERT has given surprisingly good results for different language tasks (Pires et al., 2019; Wu and Dredze, 2019).

2.2. Problem Formulation

Deep learning frameworks (Braud et al., 2017; Wang et al., 2018) cast discourse segmentation as a BI tagging problem, where B indicates the beginning of a new span and I indi-

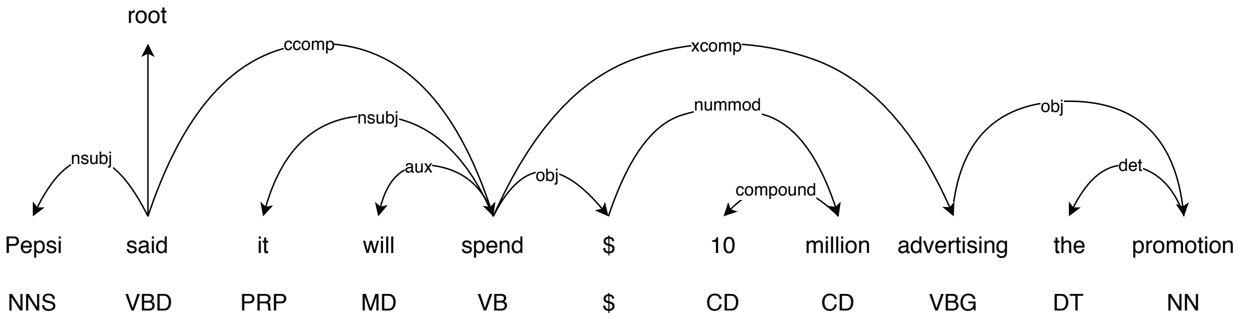

Figure 3: An example showing the features captured for each token in the sentence: we consider the gold part-of-speech tags and dependency relations with respect to the masked token.

cates the continuation of a previous span. As opposed to casting it as a sequence tagging problem, we cast it as a token classification problem. Specifically, given a sequence of tokens $T = t_0, t_1 \dots t_n$, for each token t_i, we present the input to BERT as:

$$[\mathrm{CLS}] \quad t_1 \quad t_2 \dots [\mathrm{MASK}] \dots t_n \quad [\mathrm{SEP}] \quad t_i$$

where the token t_i is replaced by the [MASK] token. Here, [CLS], [SEP] and [MASK] are special tokens used by BERT. Note that the token t_i is masked so as to prevent model overfitting.

A similar idea was applied to sentence boundary detection (Schweter and Ahmed, 2019) where a window of k characters is defined for markers such as periods, question marks, etc., and a deep network was trained to identify if a marker demarcates a sentence boundary or not. In this case, markers are well-defined (i.e. a sentence must end with a period, question mark, exclamation point or quotation marks). Unfortunately, for discourse segmentation, such markers are not well-defined which required us to check all tokens in the sentence for EDU boundaries.

We conjecture that casting the problem this way allows BERT to make better decisions as it attends to each token and not the full sequence. This is particularly useful for discourse segmentation as Wang et al. (2018) observed that segmentation is a local problem and demarcating a segment requires only on a small window of neighbouring tokens. Trying to tag the full sequence may introduce unnecessary noise and lead to errors.

Additionally, we define a positional vector p (Zhang et al., 2017) for $T = t_1, t_2 ... t_j$ relative to the masked token t_i as:

$$p_j = \begin{cases} i - j & \text{if } j < i \\ 0 & \text{if } j = i \\ j - i & \text{if } j > i \end{cases} \tag{1}$$

Following Shi and Lin (2019), we embed the positional vector and concatenate it to the encoder representations.

2.3. Joint Learning of Syntactic Features

Syntactic features are very helpful in learning language tasks. Strubell et al. (2018) particularly observed that jointly learning syntactic features improved the performance of semantic role labeling systems. We extend the idea to discourse segmentation, by jointly predicting the following features for the sentence:

1. Part-of-speech tags of all tokens in the sentence
2. Dependency parent, child(ren) and sibling(s) of the masked token
3. Dependency relation of parent token with respect to the masked token

An example is provided in Figure 3. We extract part-of-speech tags of all tokens; and the dependency parent (and corresponding relation), child(ren) and siblings of the masked token. For example, if the token 'spend' is masked, we train the classifier to learn CCOMP relation between 'said' and 'spend'; CHILD relations with respect to the tokens 'it', 'will', '$', and 'advertising'; SIBLING relation with respect to the token 'Pepsi' and NOREL with respect to every other token.

As shown in Figure 2, the first p layers of BERT learn the part-of-speech tags of the words under consideration. This layer passes information to the upper layers: d layers learn dependency relations of other words with respect to the token. The final layer ($n = 12$ in case of BERT) provides the final hidden representation of the sentence, that is fed to the decoder for classification.

Joint training offers several advantages. First, multi-task learning helps improve model performance as features learned during training for syntactic features aids segmentation. Second, features are only required during training, and not during testing or tagging. This gives us an added third advantages: we can make use of gold POS tags and parse-trees. This is particularly advantageous because, as Braud et al. (2017) observed, system-generated parse-trees adversely affect segmentation as opposed to gold trees that improved performance significantly.

2.4. Decoder

Our decoder design is fairly simple. We place a fully connected layer on top of BERT that accepts the final hidden representation of the sentence and predicts whether the token represents the beginning of a new segment or not. We use the softmax activation function to convert the linear

Dataset	Training			Validation			Test		
	# Docs	# Sents	# EDUs	# Docs	# Sents	# EDUs	# Docs	# Sents	# EDUs
eng.rst.rstdt	309	6,672	17,646	38	717	1,797	38	929	2,346
deu.rst.pcc	142	1,773	1,788	17	207	275	17	213	294
nld.rst.nldt	56	1,202	1,350	12	257	347	12	248	344
por.rst.cstn	110	1,595	1,772	14	232	552	12	123	265
eus.rst.rstdt	84	990	1,517	28	350	604	28	320	593

Table 1: Description of how the data is distributed in each dataset. Notice that the amount of training data available for languages like Dutch and Basque is too small.

Model	eng.rst.rstdt			deu.rst.pcc			nld.rst.rstdt			por.rst.cstn			eus.rst.rstdt		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Baseline	86.86	90.41	88.60	84.91	91.84	88.24	84.87	88.08	86.44	84.67	83.40	84.03	75.85	77.46	75.66
Token	95.45	94.67	95.06	**94.76**	86.05	90.20	**97.69**	86.05	91.49	**92.88**	88.68	90.73	**87.25**	80.78	83.89
Post	96.17	96.21	96.19	93.34	95.58	94.45	93.86	93.31	93.59	87.72	**94.34**	90.91	84.63	84.49	84.56
Depend	94.41	**97.19**	95.78	92.74	95.58	94.14	95.73	91.28	93.45	90.98	91.32	91.15	88.87	82.13	85.36
Both	95.24	95.48	95.36	93.81	92.86	93.33	93.02	93.02	93.02	91.60	90.57	91.08	86.02	81.96	83.94
Ensemble	**96.32**	97.02	**96.67**	92.81	**96.60**	**94.67**	94.72	**93.90**	**94.31**	89.53	93.59	**91.51**	85.59	**85.16**	**85.38**

Table 2: Empirical results and ablation study: As is evident from the obtained results, casting the problem as a token classification (Token) problem led to significant improvement. Likewise, training for POS tags (Post), dependency relations (Depend) or both (Both) improved model performance across all languages.

layer's output to a probability distribution (over two classes i.e. B and I).

3. Experimental Results

3.1. Data

To test the performance of our model, we experimented with datasets in 5 different languages:

1. The RST-DT corpus (Carlson et al., 2003) in English (eng.rst.rstdt)
2. The Potsdam Commentary corpus (Stede and Neumann, 2014) in German (deu.rst.pcc)
3. The Dutch Discourse Treebank (Redeker et al., 2012) (nld.rst.rstdt)
4. The Cross-document Structure Theory News Corpus (Cardoso et al., 2011) in Portugeuse Brazilian (por.rst.cstn)
5. Basque Discourse Treebank (Iruskieta et al., 2013) (eus.rst.rstdt)

Some statistics on the data are included in Table 1.

3.2. Setup

We implemented our tool in PyTorch [3] and used the API provided by researchers at HuggingFace[4] to fine-tune pretrained BERT. All experiments were performed using 8 NVIDIA-GTX 1080 Ti GPUs in parallel.

For training, we constructed batches of size 16. We used cross-entropy for calculating network loss and the Adam optimizer (Kingma and Ba, 2015) for updating network weights. The learning rate is set to $3e - 5$. To tune the hyper-parameters, we held-out a portion of the training data as validation set (see Table 1): all hyper-parameters were tuned on this validation set. We experimented for different values of $p, d \in \{9, 10, 11\}$ but we did not observe any significant difference in the F-scores (± 0.15). We report the best results for each dataset.

3.3. Empirical Results

We report our empirical results for discourse segmentation in Table 2. As a baseline, we cast segmentation as a BI tagging problem, following the guidelines provided by Devlin et al. (2019). One can see that by casting it instead as token classification (Token), the F-score improved significantly. In fact, simply casting the problem as token classification got us very close to the state-of-the-art for the English RST-DT corpus (Muller et al., 2019). Likewise, training for part-of-speech tags (Post) and dependency relations (Depend) also improved the F-score for each language. The best improvements were observed for the German and Dutch datasets; with F-scores improving by 4.47 and 2.87 points respectively.

It can also be observed that training either for POS tags or for dependency relations improves the F-score more significantly as compared to training for both (Both). A likely explanation for this is that training for both leads to poor generalization thereby leading to comparatively poor improvements. To assuage the problem, we increased the number of training iterations, but this led to severe overfitting. In general, ensembling (we did not consider Baseline and Both) helped and gave us better scores when compared to these models in isolation.

4. Analysis of the English dataset

Empirical results summarized in Table 2 show that performing token classification and not sequence tagging improves model performance. We saw an absolute improvement of 6.46 in the F-score showing that change in formulation alone helped achieve impressive results. To understand

[3] https://pytorch.org/

[4] https://github.com/huggingface/transformers

the impact of casting the problem in an alternate fashion and of the syntactic features considered, we analyze errors made by the model on the eng.rst.rstdt dataset.

4.1. Comparison with Existing Models

We first compare the results obtained for the eng.rst.rstdt corpus with previous work done. To ensure fair comparison, we report results for discourse segmentation at the sentence-level and not at the document-level (Braud et al., 2017). Table 3 reports the performance of our model and other competing systems.

Model	P	R	F
Soricut and Marcu (2003)	84.1	85.4	84.7
Subba and Di Eugenio (2007)	85.5	86.6	86.0
Hernault et al. (2010)	91.0	87.2	89.0
Bach et al. (2012)	91.5	90.4	91.0
Feng and Hirst (2014)	92.8	92.3	92.6
Wang et al. (2018)	92.9	95.7	94.3
Lin et al. (2019)	94.1	96.6	95.4
Muller et al. (2019)	95.3	96.8	96.0
Our Model	**96.3**	**97.0**	**96.7**
Human	98.5	98.2	98.3

Table 3: Performance of our model and other systems on the RST-DT dataset. Results are reported assuming parse trees are extracted using the BLLIP parser (as used by authors in the paper)

SPADE is a probabilistic system developed by Soricut and Marcu (2003) that makes use of lexical and syntactic information to predict segment boundaries. Subba and Di Eugenio (2007) designed a simple neural network framework that makes use of similar features to carry out text segmentation. Hernault et al. (2010) and Bach et al. (2012) were the first to cast discourse segmentation as a sequence classification problem and use biLSTM-CRFs with parse tree features. Feng and Hirst (2014) performed segmentation in two passes: the second pass uses global features extracted from the results of the first pass to segment sentences. Wang et al. (2018) used ElMo embeddings with restricted self-attention: a mechanism that computes attention score for each token with respect to a small context and not the full sequence. Lin et al. (2019) used pointer networks with parse tree information to perform join discourse segmentation and relation classification. Muller et al. (2019) used BERT contextual embeddings with convolutional character embeddings as input to a biLSTM architecture to obtain accurate segments.

As is evident from Table 3, we were able to achieve state-of-the-art results on the RST-DT corpus, beating the previous state-of-the-art model by an absolute 0.7 points and by a relative 7.2 points. It can also be observed that many of these systems were high-recall systems i.e. they end up predicting more EDUs than necessary. Our system achieved a higher precision than all, again beating the state-of-the-art by an absolute 1.0 points (relative 10 points).

4.2. Sentence length v/s Number of errors

We compare the proportion of errors made by the models with respect to the length of the sentence i.e. the number of tokens in the sentence. For the purpose of evaluation, we consider a sentence to be incorrectly segmented regardless of whether the type of error (Bach et al., 2012) is over (i.e. a sentence is segmented when it should not be) or miss (a sentence is not segmented when it should be). In Figure 4, we provide a graph showing the proportion of errors made by the models with respect to the sentence length. The proportion of errors is calculated as the ratio of sentences that were incorrectly tagged to the total number of sentences, grouped by the sentence length.

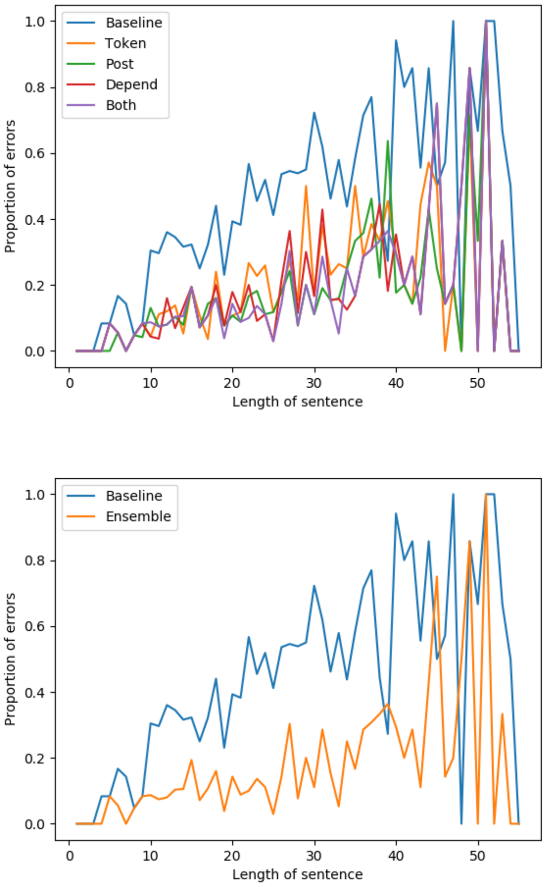

Figure 4: Graphs showing the proportion of errors made v/s the length of sentence.

As suspected, the baseline model performs poorly when the sentences are longer. However, formulating the problem in an alternate fashion and injecting syntax make the model perform much better. This effect is more discernible for sentence lengths between 30 and 45, indicating that the baseline model could not segment a large fraction of longer sentences correctly.

4.3. Frequent Error Patterns

In Table 4, we list the 8 most frequent tokens that were incorrectly segmented; and the total number of errors made by each model. As one can infer from Table 4, all models give fewer errors than the baseline model. The reduction in total number of errors is more than 62% highlight the efficacy of our models. Additionally, training for syntax further reduced these errors by more than 5%.

Token	Absolute number of Errors				
	Baseline	Token	Post	Depend	Both
and	40	21	15	20	16
that	32	11	8	6	7
to	31	22	18	21	16
the	17	9	9	8	7
as	14	4	2	3	1
in	10	3	4	3	2
for	10	7	7	6	6
if	10	3	4	3	3
Total	608	231	187	201	200

Table 4: Table showing the number of errors made by all models when tagging the 8 most frequent tokens.

On mapping these errors to rules (Carlson and Marcu, 2001), we identified that the following were most frequently violated:

1. **Confusion between infinitival complements and infinitival clauses**: Infinitival components of verbs are never fragmented into separate EDUs, whereas infinitival clauses are segmented only if that clause functions as the satellite of a PURPOSE relation. The model often confuses infinitival complements for infinitival clauses, which leads to tagging errors.
2. **Coordination in Sentences and Clauses**: Coordinated sentences and clauses are broken into separate EDUs, while coordinated verb phrases are not. Additionally, when coordination occurs in subordinate clauses, segmentation depends on whether or not the subordinate construction would normally be segmented as an EDU if it were a single clause, rather than a number of coordinated clauses. Our model made some errors in identifying such patterns.
3. **Confusion among correlative subordinators**: Correlative subordinators consist of a combination of two markers, one in the subordinate clause and the other in the superordinate clause. Examples include 'as ... long as', 'either ... or', etc. These should be broken into separate EDUs, provided the subordinate clause contains a verb. There was some confusion in correctly identifying such constructs and thus performing accurate segmentation.
4. **Punctuation**: Punctuation symbols often indicates segment boundaries. However, there may be cases where EDUs are not segmented. For instance, parenthetical expressions are usually segmented as EDUs, but if the expression is used to indicate missing information, segmentation must not be carried out. Likewise, phrases separated by semi-colons and commas are not EDUs.

While modeling segmentation as token classification helped remove these errors, injecting syntax helped remove these errors further. In particular, we observed that jointly training for part-of-speech tags helped remove punctuation errors and resolve confusions between infinitival complements and clauses. Likewise, jointly training for dependency relations helped remove errors related to coordination and correlative subordinators. Concrete examples of each error type are provided in Table 5.

5. Conclusions

Results obtained and analysis performed show how injecting syntax into the model helped achieve better results. In particular, the joint learning of syntactic features allowed the model uncover complex syntactic patterns that could not be captured by simply fine-tuning BERT. Further, the use of syntactic features helped achieve solid gains for languages such as German and Dutch; highlighting both the importance of syntax; and also certain limitations of multilingual BERT.

Several complex cases of discourse segmentation could be effectively captured by our model. We believe that having knowledge of sentence-level semantics (Moldovan and Blanco, 2012) may help identify such nuanced patterns even better. This was in fact empirically proven by Lin et al. (2019) who jointly carried out discourse segmentation and coherence relation classification, observing an incremental improvement in model performance.

A potential drawback of our system is that the time taken to tag a full sequence is quite large as the model performs sentence segmentation in $O(n)$ time while other models take $O(S)$ time, n being the number of tokens in the document, and $S << n$ being the number of sentences in the document. However, with a sufficiently large batch size; and the availability of multiple GPUs, this bottleneck can be practically resolved by performing sentence segmentation in parallel.

6. Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. We also thank the organizers of the shared task of DISRPT-2019 (Zeldes et al., 2019) for providing all data in a uniform format.

References

Bach, N. X., Minh, N. L., and Shimazu, A. (2012). A reranking model for discourse segmentation using subtree features. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 160–168. Association for Computational Linguistics.

Braud, C., Lacroix, O., and Søgaard, A. (2017). Does syntax help discourse segmentation? not so much. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2432–2442.

Cardoso, P. C., Maziero, E. G., Jorge, M. L., Seno, E. M., Di Felippo, A., Rino, L. H., Nunes, M. G., and Pardo, T. A. (2011). Cstnews-a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105.

1	**Segments**: [With 700 branches in Spain and 12 banking subsidiaries, five branches and 12 representative-offices abroad, the Banco Exterior group has a lot] [**to** offer to a potential suitor.] **Explanation**: Infinitival complements of verbs are not segmented as separate EDUs. However, both Baseline and Token confuse the infinitival clause in the sentence for a infinitival complement and end up leaving the sentence as a single EDU. Training for dependency relations allowed the model to identify this correctly as an infinitival clause and perform correct segmentation.
2	**Segments**: [The government directly owns 51.4%] [**and** Factorex, a financial services company, holds 8.42%] **Explanation**: The baseline makes an error as it cannot identify that the sentence contains a superordinate and a subordinate clause; and therefore must be segmented. Training for both part-of-speech tags and dependency relations allowed the model to correctly identify these as two different clauses and segment them.
3	**Segments**: [A private market like this just isn't big **enough**] [**to** absorb all the business.] **Explanation**: The baseline model fails to identify the comparative 'enough . . . to' as a correlative and does not segment the sentence. However, training for syntactic features allowed the model to correctly identify this construct and hence perform correct segmentation.
4	**Segments**: [On the Big Board, Crawford & Co., Atlanta,] **[(CFD)]** begins trading today] **Explanation**: Both baseline and token incorrectly assume that the parenthetical expression expresses missing information and must not be segmented. However, by predicting the POS tag of CFD as NNP which is the same as the POS tags of the words preceding it, learning POS tags allowed the model to correctly segment this sentence.

Table 5: We highlight some of the common errors made by the model and how learning syntactic features helped eliminate such errors.

Carlson, L. and Marcu, D. (2001). Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54:56.

Carlson, L., Marcu, D., and Okurowski, M. E. (2003). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Feng, V. W. and Hirst, G. (2014). Two-pass discourse segmentation with pairing and global features. *arXiv preprint arXiv:1407.8215*.

Gessler, L., Liu, Y., and Zeldes, A. (2019). A discourse signal annotation system for rst trees. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 56–61, Minneapolis, MN, June. Association for Computational Linguistics.

Hernault, H., Bollegala, D., and Ishizuka, M. (2010). A sequential model for discourse segmentation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 315–326. Springer.

Iruskieta, M., Aranzabe, M. J., de Ilarraza, A. D., Gonzalez, I., Lersundi, M., and de Lacalle, O. L. (2013). The rst basque treebank: an online search interface to check rhetorical relations. In *4th workshop RST and discourse studies*, pages 40–49.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Yoshua Bengio et al., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lin, X., Joty, S., Jwalapuram, P., and Bari, M. S. (2019). A unified linear-time framework for sentence-level discourse parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Moldovan, D. and Blanco, E. (2012). Polaris: Lymba's semantic parser. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 66–72.

Muller, P., Braud, C., and Morey, M. (2019). Tony: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Redeker, G., Berzlánovich, I., van der Vliet, N., Bouma, G., and Egg, M. (2012). Multi-layer discourse annotation of a dutch text corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Schweter, S. and Ahmed, S. (2019). Deep-EOS: General-Purpose Neural Networks for Sentence Boundary Detection. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*. accepted.

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 149–156, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stede, M. and Neumann, A. (2014). Potsdam commentary corpus 2.0: Annotation for discourse research. In *LREC*, pages 925–929.

Strubell, E., Verga, P., Andor, D., Weiss, D., and McCallum, A. (2018). Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Subba, R. and Di Eugenio, B. (2007). Automatic discourse segmentation using neural networks. In *Proc. of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, pages 189–190.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Wang, Y., Li, S., and Yang, J. (2018). Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967.

Wu, S. and Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844.

Zeldes, A., Das, D., Maziero, E. G., Antonio, J., and Iruskieta, M. (2019). Introduction to discourse relation parsing and treebanking (DISRPT): 7th workshop on rhetorical structure theory and related formalisms. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 1–6, Minneapolis, MN, June. Association for Computational Linguistics.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.