Text Segmentation by Cross Segment Attention

Michal Lukasik, Boris Dadachev, Gonçalo Simões, Kishore Papineni
Google Research
{mlukasik,bdadachev,gsimoes,papineni}@google.com

Abstract

Document and discourse segmentation are two fundamental NLP tasks pertaining to breaking up text into constituents, which are commonly used to help downstream tasks such as information retrieval or text summarization. In this work, we propose three transformer-based architectures and provide comprehensive comparisons with previously proposed approaches on three standard datasets. We establish a new state-of-the-art, reducing in particular the error rates by a large margin in all cases. We further analyze model sizes and find that we can build models with many fewer parameters while keeping good performance, thus facilitating real-world applications.

1 Introduction

Text segmentation is a traditional NLP task that breaks up text into constituents, according to predefined requirements. It can be applied to documents, in which case the objective is to create logically coherent sub-document units. These units, or segments, can be any structure of interest, such as paragraphs or sections. This task is often referred to as *document segmentation* or sometimes simply *text segmentation*. In Figure 1 we show one example of document segmentation from Wikipedia, on which the task is typically evaluated (Koshorek et al., 2018; Badjatiya et al., 2018).

Documents are often multi-modal, in that they cover multiple aspects and topics; breaking a document into uni-modal segments can help improve and/or speed up down stream applications. For example, document segmentation has been shown to improve information retrieval by indexing sub-document units instead of full documents (Llopis et al., 2002; Shtekh et al., 2018). Other applications such as summarization and information extraction can also benefit from text segmentation (Koshorek et al., 2018).

Early life and marriage:
Franklin Delano Roosevelt was born on January 30, 1882, in the Hudson Valley town of Hyde Park, New York, to businessman James Roosevelt I and his second wife, Sara Ann Delano. (...) Aides began to refer to her at the time as "the president's girlfriend", and gossip linking the two romantically appeared in the newspapers.

(...)

Legacy:
Roosevelt is widely considered to be one of the most important figures in the history of the United States, as well as one of the most influential figures of the 20th century. (...) Roosevelt has also appeared on several U.S. Postage stamps.

Figure 1: Illustration of text segmentation on the example of the Wikipedia page of President Roosevelt. The aim of document segmentation is breaking the raw text into a sequence of logically coherent sections (e.g., "Early life and marriage" and "Legacy" in our example).

A related task called *discourse segmentation* breaks up pieces of text into sub-sentence elements called Elementary Discourse Units (*EDUs*). EDUs are the minimal units in discourse analysis according to the Rhetorical Structure Theory (Mann and Thompson, 1988). In Figure 2 we show examples of EDU segmentations of sentences. For example, the sentence "Annuities are rarely a good idea at the age 35 because of withdrawal restrictions" decomposes into the following two EDUs: "Annuities are rarely a good idea at the age 35" and "because of withdrawal restrictions", the first one being a statement and the second one being a justification in the discourse analysis. In addition to being a key step in discourse analysis (Joty et al., 2019), discourse segmentation has been shown to improve a number of downstream tasks, such as text summarization, by helping to identify fine-grained sub-sentence units that may have different levels of importance when creating a summary (Li et al., 2016).

Multiple neural approaches have been recently proposed for document and discourse segmentation. (Koshorek et al., 2018) proposed the use of

Sentence 1:
Annuities are rarely a good idea at the age 35 || because of withdrawal restrictions

Sentence 2:
Wanted: || An investment || that's as simple and secure as a certificate of deposit || but offers a return || worth getting excited about.

Figure 2: Example discourse segmentations from the RST-DT dataset (Carlson et al., 2001). In the segmentations, the EDUs are separated by the || character.

hierarchical Bi-LSTMs for document segmentation. Simultaneously, (Li et al., 2018) introduced an attention-based model for both document segmentation and discourse segmentation, and (Wang et al., 2018) obtained state of the art results on discourse segmentation using pretrained contextual embeddings (Peters et al., 2018). Also, a new large-scale dataset for document segmentation based on Wikipedia was introduced by (Koshorek et al., 2018), providing a much more realistic setup for evaluation than the previously used small scale and often synthetic datasets such as the Choi dataset (Choi, 2000).

However, these approaches are evaluated on different datasets and as such have not been compared against one another. Furthermore they mostly rely on RNNs instead of the more recent transformers (Vaswani et al., 2017) and in most cases do not make use of contextual embeddings which have been shown to help many classical NLP tasks (Devlin et al., 2018).

In this work we aim at addressing these limitations and bring the following contributions:

1. We compare recent approaches that were proposed independently for text and/or discourse segmentation (Li et al., 2018; Koshorek et al., 2018; Wang et al., 2018) on three public datasets.
2. We introduce three new model architectures based on transformers and BERT-style contextual embeddings to the document and discourse segmentation tasks. We analyze the strengths and weaknesses of each architecture and establish a new state-of-the-art.
3. We show that a simple paradigm argued for by some of the earliest text segmentation algorithms can achieve competitive performance in the current neural era.
4. We conduct ablation studies analyzing the importance of context size and model size.

2 Literature review

Document segmentation Many early research efforts were focused on unsupervised text segmentation, doing so by quantifying lexical cohesion within small text segments (Hearst, 1997; Choi, 2000; Utiyama and Isahara, 2001). Being hard to precisely define and quantify, lexical cohesion has often been approximated by counting word repetitions. However, unsupervised algorithms suffer from two main drawbacks: they are hard to specialize for a given domain and do not naturally deal with multi-scale issues. Indeed, the desired segmentation granularity (paragraph, section, chapter, etc.) is necessarily task dependent and supervised learning provides a way of addressing this property. Therefore, supervised algorithms have been a focus of many recent works.

In particular, multiple neural approaches have been proposed for the task. In one, a sequence labeling algorithm is proposed where each sentence is encoded using a Bi-LSTM over tokens, and then a Bi-LSTM over sentence encodings is used to label each sentence as ending a segment or not (Koshorek et al., 2018). Authors consider a large dataset based on Wikipedia, and report improvements over unsupervised text segmentation methods. In another work, a sequence-to-sequence model is proposed (Li et al., 2018), where the input is encoded using a BiGRU and segment endings are generated using a pointer network (Vinyals et al., 2015). The authors report significant improvements over sequence labeling approaches, however on a dataset composed of 700 artificial documents created by concatenating segments from random articles from the Brown corpus (Choi, 2000). Lastly, (Badjatiya et al., 2018) consider an attention-based CNN-Bi-LSTM model and evaluate it on three small-scale datasets.

Discourse Segmentation Contrary to document segmentation, discourse segmentation has historically been framed as a supervised learning task. However, a challenge of applying supervised approaches for this type of segmentation is the fact that the available dataset for the task is limited (Carlson et al., 2001). For this reason, approaches for discourse segmentation usually rely on external annotations and resources to help the models generalize. Early approaches to discourse segmentation were based on features from linguistic annotations such as POS tags and parsing trees (Soricut and Marcu, 2003; Xuan Bach et al., 2012; Joty

et al., 2015). The performance of these systems was highly dependent on the quality of the annotations.

Recent approaches started to rely on end-to-end neural network models that do not need linguistic annotations to obtain high-quality results, relying instead on pretrained models to obtain word or sentence representations. An example of such work is (Li et al., 2018), which proposes a sequence-to-sequence model getting a sequence of GloVe (Pennington et al., 2014) word embeddings as input and generating the EDU breaks. Another approach utilizes ELMO pretrained embeddings in the CRF-Bi-LSTM architecture and achieves state-of-the-art results on the task (Wang et al., 2018).

3 Architectures

We propose three model architectures for segmentation. One uses only local context around each candidate break, while the other two leverage the full context from the input.

All our models rely on the same preprocessing technique. We feed the raw input into the word-piece tokenizer open-sourced as part of the BERT release (Devlin et al., 2018), more precisely its English, uncased variant, which has a vocabulary size of 30,522 word-pieces.

3.1 Cross-segment BERT

For our first model, we represent each candidate break by its left and right local contexts, i.e., the sequences of word-piece tokens that come before and after, respectively, the candidate break. The main motivation for this model is its simplicity; however using only local contexts might be sub-optimal, as longer distance linguistic artifacts are likely to help locating breaks. It is interesting to note that using local context was a common approach with earlier text segmentation models, such as (Hearst, 1997), which were studying semantic shift by comparing the word distributions before and after each candidate break.

In Figure 3(a) we illustrate the model. The input is composed of a *[CLS]* token, followed by the two contexts concatenated together, and separated by a *[SEP]* token. When necessary, short contexts are padded to the left or to the right with *[PAD]* tokens. The input is then fed into a transformer encoder (Vaswani et al., 2017), which is initialized with the publicly available $\text{BERT}_{\text{LARGE}}$ model. The $\text{BERT}_{\text{LARGE}}$ model has 24 layers, uses 1024-dimensional embeddings and 16 attention heads. The model is then fine-tuned on each task. The released BERT checkpoint supports sequences of up to 512 tokens, so we keep at most 255 word-pieces for each side. We study the effect of length of the contexts, and denote the context configuration by n-m where n and m are the number of word piece tokens before and after the *[SEP]* token.

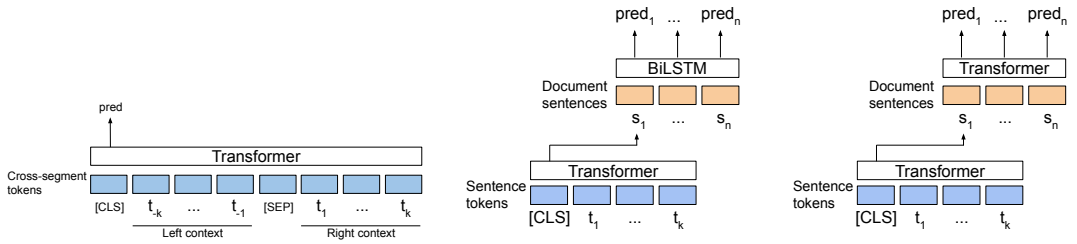
3.2 BERT+Bi-LSTM

Our second proposed model is illustrated in Figure 3(b). It starts by encoding each sentence with $\text{BERT}_{\text{LARGE}}$ independently. Then, the tensors produced for each sentence are fed into a Bi-LSTM that is responsible for capturing a representation of the sequence of sentences with an indefinite size.

When encoding each sentence with BERT, all the sequences start with a *[CLS]* token. If the segmentation decision is made at the sentence level (e.g., document segmentation), we use the *[CLS]* token as input of the LSTM. In cases in which the segmentation decision is made at the word level (e.g., discourse segmentation), we obtain BERT's full sequence output and use the left-most word-piece of each word as input of the LSTM. Note that, due to the context being short for the discourse segmentation task, it is fully encoded in a single pass using BERT. Alternatively, one could encode each word independently; considering that many words consist of a single word-piece, encoding them with a deep transformer encoder would be somewhat wasteful of computing resources.

With this model, we reduce the BERT's inputs to a maximum sentence size of 64 tokens. Keeping this size small helps reduce training and inference times, since the computational cost of transformers (and self-attention in particular) increases quadratically with the input length. Then, the LSTM is responsible for handling the diverse and potentially large sequence of sentences. In practice, we set a maximum document length of 128 sentences, which is enough to accommodate the vast majority of real documents. Longer documents are simply split into consecutive, non-overlapping chunks of 128 sentences and treated as independent documents.

In a sense, the hierarchical nature of this model is closer to the recent neural approaches such as (Koshorek et al., 2018).



(a) Cross-Segment BERT (b) BERT+Bi-LSTM (c) Hierarchical BERT

Figure 3: Our proposed segmentation models, illustrating the document segmentation task. In the cross-segment BERT model (left), we feed a model with a local context surrounding a potential segment break: k tokens to the left and k tokens to the right. In the BERT+Bi-LSTM model (center) we first encode each sentence using a BERT model, and then feed the sentence representations into a Bi-LSTM. In the hierarchical BERT model (right), we first encode each sentence using BERT and then feed the output sentence representations in another transformer-based model.

3.3 Hierarchical BERT

Our third model is a hierarchical BERT model that also encodes full documents, replacing the document-level LSTM encoder from the BERT+Bi-LSTM model with a transformer encoder. This architecture is similar to the HIBERT model used for document summarization by (Zhang et al., 2019), encoding each sentence independently. The $[CLS]$ token representations of each sentence is passed onto the document encoder, which is then able to relate the different sentences through cross-attention, as illustrated in Figure 3(c).

Due to the quadratic computational cost of transformers, we use the same limits as BERT+Bi-LSTM for input sequence sizes: 64 word-pieces per sentence and 128 sentences per document.

To keep the number of model parameters comparable with our other proposed models, we use 12 layers for both the sentence and the document encoders, for a total of 24 layers. In order to use the $\text{BERT}_{\text{Base}}$ checkpoint for these experiments, we use 12 attention heads and 768-dimensional word-piece embeddings.

We study two alternative initialization procedures:

- initializing both sentence and document encoders using $\text{BERT}_{\text{Base}}$
- pre-training all model weights on Wikipedia, using the procedure described in (Zhang et al., 2019), which can be summarized as a "masked sentence" prediction objective, analogously to the "masked token" pre-training objective from BERT.

We call this model hierarchical BERT for consistency with the literature.

4 Evaluation methodology

4.1 Datasets

We perform our experiments on datasets commonly used in the literature. Document segmentation experiments are done on Wiki-727K and Choi, while discourse segmentation experiments are done on the RST-DT dataset. We summarize statistics about the datasets in Table 1.

Wiki-727K The Wiki-727K dataset (Koshorek et al., 2018) contains 727 thousand articles from a snapshot of English Wikipedia, which are randomly partitioned into train, development and test sets. We re-use the original splits provided by the authors. While several segmentation granularities are possible, the dataset is used to predict *section* boundaries. The average number of segments per document is 3.5, with an average segment length of 13.6 sentences.

We found that the preprocessing methodology used on the Wiki-727K dataset can have a noticeable effect on the final numerical results, in particular when filtering lists, code snippets and other special elements. We used the original preprocessing script (Koshorek et al., 2018) for fair comparisons.

Choi Choi's dataset (Choi, 2000) is an early dataset containing 700 synthetic documents made of concatenated extracts of news articles. Each document is made of 10 segments, where each segment was created by sampling a document from the Brown corpus and then sampling a random segment length up to 11 sentences.

This dataset was originally used to evaluate unsupervised segmentation algorithms, so it is somewhat ill-designed to evaluate supervised algorithms.

We use this dataset as a best-effort attempt to allow comparison with some of the previous literature. However, we had to create our own splits as no standard splits exist: we randomly sampled 200 documents as a test set and 50 documents as a validation set, leaving 450 documents for training, following evaluation from (Li et al., 2018). Since the Brown corpus only contains 500 documents, the same documents are sampled over and over, necessarily resulting in data leakage between the different splits. Its use should therefore be discouraged in future research.

RST-DT We perform experiments on discourse segmentation by using the RST Discourse Treebank (RST-DT) (Carlson et al., 2001). This dataset is composed of 385 Wall Street Journal articles that are part of the Penn Treebank (Marcus et al., 1994). These articles are already divided in a training set composed of 347 articles and a test set composed of 38 articles. In order to tune the hyperparameters of the model, we randomly sampled 10% of the training set sentences and used them as a validation set.

Since this dataset is used for discourse segmentation, all the segmentation decisions are made at the intra-sentence level (i.e., the context that is used in the decisions is just the sentence). In order to make the evaluation consistent with other systems in the literature we decided to use the sentence splits that are available in the dataset, even though they are not annotated by humans. For this reason, there are cases in which some EDUs (which were manually annotated) overlap between two sentences. In such cases, we merge the two sentences.

4.2 Metrics

Following the trend of many studies on text segmentation (Soricut and Marcu, 2003; Li et al., 2018), we evaluate our approaches based on Precision, Recall and F1-score with regard to the internal boundaries of the segments only. In our evaluation, we do not include the last boundary of each sentence/document because it would be trivial to categorize it as a positive boundary, leading to an artificial inflation of the results.

To allow comparison with the existing literature, we also use the P_k metric (Beeferman et al., 1999) to evaluate our results on Choi's dataset:

$$P_k(ref, hyp) = \sum_{i=0}^{n-k} \delta_{ref}(i, i+k) \neq \delta_{hyp}(i, i+k)$$

	Docs	Sections	Sentences
Wiki-727K Train	582,146	2,025,358	26,988,063
Wiki-727K Dev	72,354	179,676	3,375,081
Wiki-727K Test	73,233	182,563	3,457,771
Choi Train	450	4,500	31,075
Choi Dev	50	500	3,291
Choi Test	200	2,000	14,039
	Docs	Sentences	EDUs
RST-DT Train	347	7,028	19,443
RST-DT Test	38	864	2,346

Table 1: Statistics about the datasets.

P_k relies on a sliding window of size k to compare a candidate segmentation (hyp) to a reference (ref) for an input of n sentences or words (depending on the task at hand). At every position of the window, it checks whether its beginning (at sentence index i) and end (at index $i+k$) belong to the same segment in the reference (denoted by $\delta_{ref}(i, i+k)$, equal to 1 when the two window ends are in the same segment and 0 otherwise). It then adds a penalty of 1 if the candidate segmentation disagrees. As such, lower P_k scores indicate better performance. In practice, k is document-dependent and set to half the average segment size in the reference segmentation.

5 Results

In Table 2, we report results from document and discourse segmentation experiments on the three datasets presented in Section 4.1. We include several state-of-the-art baselines which had not been compared against one another as they have been proposed independently over a short time period: hierarchical Bi-LSTM (Koshorek et al., 2018), SEGBOT (Li et al., 2018) and Bi-LSTM+CRF+ELMO (Wang et al., 2018). We estimate standard deviations for our proposed models and were able to calculate those from the hierarchical Bi-LSTM, whose code and trained checkpoint were publicly released.

To train our models, we used the AdamW optimizer (Loshchilov and Hutter, 2017) with a 10% dropout rate as well as a linear warmup procedure. Learning rates are set between 1e-5 and 5e-6, chosen to maximize the F1-score on the validation sets from each dataset. For the more expensive models, and especially on the Wiki-727K dataset, we trained our models using Google Cloud TPUs.

We can see from the table that our models outperform the baselines across all datasets, reducing

	Wiki-727K			RST-DT			Choi	
	F1	Recall	Precision	F1	Recall	Precision	F1	P_k
Bi-LSTM (Koshorek et al., 2018)	57.7±0.1	49.5±0.2	69.3±0.1	-	-	-	-	-
SEGBOT (Li et al., 2018)	-	-	-	92.2	92.8	91.6	-	0.33
Bi-LSTM+CRF (Wang et al., 2018)	-	-	-	94.3	95.7	92.8	-	-
Cross-segment BERT 128-128	66.0±0.1	63.2±0.2	69.1±0.1	95.0±0.5	**98.0±0.4**	92.1±0.8	**99.9±0.1**	**0.07±0.4**
BERT+Bi-LSTM	59.9±0.1	53.9±0.1	67.3±0.1	**95.7±0.4**	96.8±0.5	**94.6±0.5**	99.8±0.1	0.17±0.6
Hier. BERT	**66.5±0.1**	**63.5±0.1**	**69.8±0.1**	95.2±0.4	96.6±0.7	93.8±0.5	99.5±0.1	0.38±0.9
Human (Wang et al., 2018)	-	-	-	98.5	98.2	98.3	-	-

Table 2: Test set results on text segmentation and discourse segmentation for baselines and our models. Where possible, we estimate standard deviations by bootstrapping the test set 100 times.

the relative error margins from the best baseline by 20%, 12% and 79% respectively on the Wiki-737K, RST-DT and Choi datasets. The improvements are statistically significant for Wiki-727K and RST-DT. Even though the improvement is not statistically significant on the Choi dataset due to high standard deviation, the error is impressively low. It is important to point out that the Choi dataset is a small-scale synthetic dataset, and as such limited. Since each document is a concatenation of extracts from random news articles, it is an artificially easy task for which a previous neural baseline achieved an already low error margin. Moreover, on the Choi dataset the cross-segment BERT model obtains very good results compared to the hierarchical models (which do not attend across the candidate break). This aligns with the expectation that locally attending across a segment break is sufficient here, as we expect large semantic shifts due to the artificial nature of the dataset.

Hierarchical models, with a sentence encoder followed by a document encoder, are somewhat ill-designed for RST-DT. Indeed, the *discourse segmentation* task is about segmenting individual sentences, and there is no notion of document context. Still, we trained hierarchical models on the RST-DT dataset for completeness. Using a single Bi-LSTM layer increased the F1-score by 0.4% over making predictions directly using BERT (F1-score of 95.3%). It's worth noting that several known LSTM downsides were particularly apparent on the Wiki-727K dataset: the model was harder to train and significantly slower during both training and inference.

Regarding the hierarchical BERT model, different initialization methods were used for the two document segmentation datasets. On the Choi dataset, a HIBERT initialization (a model fully pre-trained end-to-end for hierarchical BERT, similarly to (Zhang et al., 2019) was necessary to get good results, due the small dataset size. On the contrary, we obtained slightly better results initializing both levels of the hierarchy with $\text{BERT}_{\text{Base}}$ on the Wiki-727K dataset, even though the model took longer to converge. Other initializations, e.g., random for both levels of the hierarchy or $\text{BERT}_{\text{Base}}$ at the lower level and random at the upper level, gave worse results.

Perhaps the most surprising result from Table 2 is the good performance of our cross-segment BERT model across all datasets, since it only relies on local context to make predictions. And while the BERT checkpoints were pre-trained using (among other things) a next-sentence prediction task, it was not clear a priori that our cross-segment BERT model would be able to detect much more subtle semantic shifts. To further evaluate the effectiveness of this model, we tried using longer contexts. In particular, we considered using a cross-segment BERT with 255-255 contexts, achieving 67.1 F1, 73.9 recall and 61.5 precision scores. Therefore, we can see that encoding the full document in a hierarchical manner using transformers does not improve over cross-segment BERT on this dataset. This suggests that BERT self-attention mechanism applied across candidate segment breaks, with a limited context, is in this case just as powerful as separately encoding each sentence and then allowing a flow of information across encoded sentences. In the next section we further analyze the impact of context length on the results from the cross-segment BERT model.

6 Analyses

In this section we perform additional analyses and ablation studies to better understand our segmentation models.

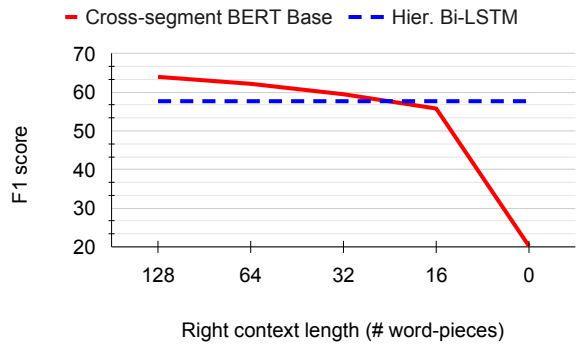


Figure 4: Analysis of the importance of the right context length (solid red line). Dashed blue line denotes the hierarchical Bi-LSTM baseline encoding the full context (Koshorek et al., 2018).

Experiments revolve around the cross-segment BERT model. We choose this model because it has several advantages over its alternatives:

- It outperforms all baselines previously reported as state-of-the-art, and its results are competitive with the more complex hierarchical approaches we considered.
- It is conceptually close to the original BERT model (Devlin et al., 2018), whose code is open-source, and is as such simple to implement.
- It only uses local document context and therefore does not require encoding an entire document to segment a potentially small piece of text of interest.

One application for text segmentation is in assisting a document writer in composing a document, for example to save them time and effort. The task proposed by (Lukasik and Zens, 2018), aligned with what industrial applications such as Google Docs Explore provide, was to recommend related entities to a writer in real time. However, text segmentation could also help authors in structuring their document better by suggesting where a section break might be appropriate. Motivated by this application, we next analyze how much context is needed to reliably predict a section break.

6.1 Role of trailing context size

For the aforementioned application, it would be helpful to use as little trailing (after-the-break) context as possible. This way, we can suggest section breaks sooner. Reducing the context size also speeds up the model (as cost is quadratic in sequence length). To this end, we study the effect of trailing context size, going from 128 word-piece tokens down to 0. For this set of experiments, we held the leading context size fixed at 128 tokens, and tuned $\text{BERT}_{\text{BASE}}$ with a batch size of 1536 examples and a learning rate of 5e-5. The results for these 128-n experiments are shown in Figure 4.

While the results are intuitive, it is not clear whether the performance drops because of smaller trailing context or because of smaller overall context. To answer this, we ran another experiment with 256 tokens on the left and 0 tokens on the right (256-0). With all else being the same, this 256-0 experiment attains F1 score of 20.2. This is much smaller than 64.0 F1 with 128 tokens on each side of the proposed break. Clearly, it is crucial that the model sees both sides of the break. This aligns with the intuition that word distributions before and after a true segment break are typically quite different (Hearst, 1997). However, presenting the model with just the distributions of tokens on either side of the proposed break leads to poor performance: In another experiment, we replaced the running text on either side with a sorted list of 128 most frequent tokens seen in a larger context (256 tokens) on either side, padding as necessary, and tuned $\text{BERT}_{\text{BASE}}$ with all else the same. This 128-128 experiment attains 39.1 F1 score, compared to 64.0 with 128-128 running text on either side. This suggests that high-performing models are doing more than just counting tokens on each side to detect semantic shift.

Architecture	Parameters	F1
L24-H1024-A16	336M	66.0
L12-H768-A12	110M	64.0
L12-H512-A8	54M	63.4
L12-H256-A8	17M	62.3
L6-H256-A8	13M	60.2
L4-H256-A4	11M	58.2
L12-H128-A8	6M	59.2
L6-H128-A8	5M	57.9
L12-H64-A8	2.6M	55.5

Table 3: Effect of model architecture on Wiki-727K results.

6.2 Role of Transformer architecture

The best cross-segment BERT model relies on $\text{BERT}_{\text{Large}}$. While powerful, this model is slow and expensive to run. For large-scale applications such as offline analysis for web search or online document processing such as Google Docs or Microsoft Office, such large models are prohibitively

Architecture	Parameters	F1
L4-H256-A4	11M	63.0
L6-H128-A8	5M	62.5

Table 4: Distillation results on the Wiki-727K dataset.

expensive. Table 3 shows the effect of model size on performance. For these experiments, we initialized the training with models pre-trained as in the BERT paper (Devlin et al., 2018). The first two experiments are initialized with $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{BASE}}$ respectively.

Overall, the larger the model, the better the performance. These experiments also suggest that the configuration also matters, in addition to the size. A 128-dimensional model with more layers can outperform a 256-dimensional model with fewer layers. While the new state-of-the-art is several standard deviations better than the previous one (as reported in Table 2), this gain came at a steep cost in the model size. This is unsatisfactory, as large size hinders the possibility of using the model at scale and with low latency, which is desirable for this application (Wang et al., 2018). In the next section, we explore smaller models with better performance using model distillation.

6.3 Model distillation

As can be seen from the previous section, performance degrades quite quickly as smaller and therefore more practical networks are used. An alternative to the pre-training/fine-tuning approach used above is distillation, which is a popular technique to build small networks (Bucila et al., 2006; Hinton et al., 2015). Instead of training directly a small model on the segmentation data with binary labels, we can instead leverage the knowledge learnt by our best network —called in this context the 'teacher'— as follows. First, we record the predictions, or more precisely the output logits, from the teacher model on the full dataset. Then, a small 'student' model is trained using a combination of a cross-entropy loss with the true labels, and a MSE loss to mimick the teacher logits. The relative weight between the two objectives is treated as a hyperparameter.

Distillation results are presented in Table 4. We can see that the distilled models perform better than models trained directly on the training data without a teacher, increasing F1-scores by over 4 points. We notice that distillation allows much more compact models to significantly outperform the previous state-of-the-art. Unfortunately, we cannot directly compare model sizes with (Koshorek et al., 2018) since they rely on a subset of the embeddings from a public word2vec archive that includes over 3M vocabulary items, including phrases, most of which are likely never used by the model. It is however fair to say their hierarchical Bi-LSTM model relies on dozens of millions of embedding parameters (even though these are not fine-tuned during training) as well as several million LSTM parameters.

7 Conclusion

In this paper, we introduce three new model architectures for text segmentation tasks: a cross-segment BERT model that uses only local context around candidate breaks, as well as two hierarchical models, BERT+Bi-LSTM and hierarchical BERT. We evaluated these three models on document and discourse segmentation using three standard datasets, and compared them with other recent neural approaches. Our experiments showed that all of our models improve the current state-of-the-art. In particular, we found that a cross-segment BERT model is extremely competitive with hierarchical models which have been the focus of recent research efforts (Chalkidis et al., 2019; Zhang et al., 2019). This is surprising as it suggests that local context is sufficient in many cases. Due to its simplicity, we suggest at least trying it as a baseline when tackling other segmentation problems and datasets.

Naturally these results do not imply that hierarchical models should be disregarded. We showed they are strong contenders and we are convinced there are applications where local context is not sufficient. We tried several encoders at the upper-level of the hierarchy. Our experiments suggest that deep transformer encoders are useful for encoding long and complex inputs, e.g., documents for document segmentation applications, while Bi-LSTMs proved useful for discourse segmentation. Moreover, RNNs in general may also be useful for very long documents as they are able to deal with very long input sequences.

Finally, we performed ablation studies to better understand the role of context and model size. Consequently, we showed that distillation is an effective technique to build much more compact models to use in practical settings.

References

Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. *CoRR*, abs/1808.09935.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1):177–210.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4317–4323.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 26–33, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Marti Hearst. 1997. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Gabriel Murray. 2019. Discourse analysis and its applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–17, Florence, Italy. Association for Computational Linguistics.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2015. CODRA: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. *CoRR*, abs/1803.09337.

Jing Li, Aixin Sun, and Shafiq Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4166–4172. International Joint Conferences on Artificial Intelligence Organization.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The role of discourse units in near-extractive summarization. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147, Los Angeles. Association for Computational Linguistics.

Fernando Llopis, Antonio Ferrández Rodríguez, and José Luis Vicedo González. 2002. Text segmentation for efficient information retrieval. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 373–380, Berlin, Heidelberg. Springer-Verlag.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Michal Lukasik and Richard Zens. 2018. Content explorer: Recommending novel entities for a document writer. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3371–3380, Brussels, Belgium. Association for Computational Linguistics.

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(3):243–281.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. 2018. Applying topic segmentation to document-level information retrieval. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*,

CEE-SECR '18, pages 6:1–6:6, New York, NY, USA. ACM.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 499–506.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2692–2700, Cambridge, MA, USA. MIT Press.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing EMNLP-18*, pages 962–967. Association for Computational Linguistics.

Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu. 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, Seoul, South Korea. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069.