A More Effective Sentence-Wise Text Segmentation Approach Using BERT

Amit Maraj(✉), Miguel Vargas Martin, and Masoud Makrehchi

Ontario Tech University, Oshawa, ON L1G 0C5, Canada
amit.maraj@ontariotechu.net,
{miguel.martin,masoud.makrehchi}@ontariotechu.ca

Abstract. Text Segmentation is a Natural Language Processing based task that is aimed to divide paragraphs and bodies of text into topical, semantic blocks. This plays an important role in creating structured, searchable text-based representations after digitizing paper-based documents for example. Traditionally, text segmentation has been approached with sub-optimal feature engineering efforts and heuristic modelling. We propose a novel supervised training procedure with a pre-labeled text corpus along with an improved neural Deep Learning model for improved predictions. Our results are evaluated with the P_k and WindowDiff metrics and show performance improvements beyond any public text segmentation system that exists currently. The proposed system utilizes Bidirectional Encoder Representations from Transformers (BERT) as an encoding mechanism, which feeds to several downstream layers with a final classification output layer, and even shows promise for improved results with future iterations of BERT.

Keywords: Text segmentation · Natural Language Processing · Natural language understanding

1 Introduction

Text segmentation can often be seen as a preliminary task useful for cleaning and providing meaningful text for other, more downstream Natural Language Processing (NLP) tasks, such as text summarization, relation extraction, natural language understanding, language modelling, etc. Text segmentation is the task of creating clear distinctions between contextual topics within a document. Recognizing the layout of unstructured digital documents is an important step when parsing the documents into a more structured, machine-readable format for downstream applications. Where a document could be some arbitrary length of text such as a chapter in a book or a section in a research paper, a "somewhat supervised" idea of text segments could be paragraphs. The reason this is not completely supervised is because segments within a body of text can be relatively subjective - depending on the reader. Most text segmentation datasets are very clean and include clear distinctions between segments. For example, within a play, an act can be considered a text segment, but so can dialogue between speakers. This challenge makes it inherently difficult to appropriately identify text segments, which is usually heavily influenced by the data being used in the system.

The research field of document segmentation is very deep and as of recently with the leverage of novel deep learning methods such as Convolutional Neural Networks (CNN), has made leaps beyond what was previously capable. Although this field is of particular interest for its obvious implications, the research area of text segmentation in the realm of deep learning is still novel and relatively undiscovered.

Historically, text segmentation has proven to be a task best tackled with more unsupervised approaches. This is in part, due to a lack of available training data along with viable hardware to run more complex, resource-hungry supervised systems. Technological advancements and increased research efforts in recent years have facilitated breakthroughs in supervised learning, allowing researchers to innovate in this field with less barriers. This is not to mention the ever increasing volume of data - the essential fuel for these supervised learning techniques.

Our research leverages recent advancements in the field of NLP to improve text segmentation results. In particular, we use BERT as a rich sentence encoder and show that by including a more thoughtful text segmentation focused data augmentation technique, state-of-the-art results in this field can be achieved with minimal training. We contribute a step forward in the text segmentation space with an elementary framework, bolstered by useful propositions for future improvements.

2 Related Works

2.1 Text Segmentation

Text segmentation is a fundamental task in NLP, lending way to various implementations and approaches over the years. At a higher level, the process of text segmentation is often more aligned with the study of topic segmentation, which aims to identify more abstract, discontiguous topics within the document [10,13]. Identified topics can include a variable level of sentences, which are, at an atomic level, the building blocks of said topics. The goal in more recent years has been to adopt a more supervised approach for this task [2,4,15,41]. At a finer level, text segmentation includes the analysis of elementary discourse units, often referred to as EDU segmentation [17]. EDUs are small intent-based units within sentences that serve as building blocks to create a structured sentence. Identifying these units is called discourse parsing and are a part of Rhetorical Structure Theory [38].

There has been a myriad of unsupervised and supervised approaches towards topic segmentation, both of which have very distinct ways of understanding the document at hand. While unsupervised methods require a more globally informed context (i.e., an understanding of the entire document) before making predictions, traditional supervised techniques can make predictions with less overall contextual knowledge of the document, thus requiring less RAM and upfront processing time. Because of these limitations with unsupervised approaches, use within real-time production applications is unpractical. It is worth noting that some of these unsupervised techniques have been very effective, most notably topic modelling [12,20,25,29]. Latent Dirichlet Allocation (LDA) [6] has seen considerable success in the topic modelling domain, where the algorithm can globally analyze a document and understand topics on a sentence-wise basis successfully. Recent research has even seen LDA be more effective when combined with supervised methods [33] in an effort to provide richer features.

On the other hand, supervised techniques have shown to be more increasingly accurate in predicting segment boundaries, due to the improved capabilities of newer machine learning and deep learning propositions, thus challenges such as global document contextuality and understanding have become less important to the performance of said techniques. Recent research done by Kosharek et al. [15] shows that text segmentation can be tackled as a supervised task using Long Short-Term Memory (LSTM) [14] networks. The LSTM is great at capturing contextuality between its inputs and can understand dependencies from previous inputs. Extending upon this, a more recent approach introduces an LSTM variation, where Barrow et al. [4] layers bi-directional LSTMs to encode sentences, create a segment bound predictor, and a segment labeller.

One such model proposed by Badjatiya et al. [2] shows improvement over pre-existing benchmarks by adopting a sliding window approach to contextuality. The researchers show the effectiveness of using just three sentences (one sentence before, one target sentence, and one sentence after) as input for a binary prediction. Rather than approaching text segmentation from a global or individual context, this approach blends the two effectively. For this reason, we use this as a foundation for our work.

2.2 Data Augmentation

Rich text data is hard to come by and the lack of annotated data especially has proved to be challenging for building larger, supervised models. Data augmentation techniques for numerical data is commonplace in data science and can help to bolster the performance of models. Methods such as Synthetic Minority Over-Sampling (SMOTE) [9] can help with imbalanced data, whereas methods such as autoencoding [30] can help build new features by learning deep connections. In the space of NLP, the field of data augmentation has become vast [8].

While traditional data requires numeric-level alteration for effective augmentation and image data requires pixel-level alteration, approaches for NLP differ as augmentation required word-level alteration. Augmentation has proven to be beneficial in the world of NLP through techniques such as lexical substitution [21], vector substitution [36], back translation [40], sentence and word-wise shuffling [37,44], and embedding combinations [43].

We also prove that with thoughtful augmentation, we can boost the performance of our proposed model by preventing it from having to deal with as much class imbalance, and focus on understanding the difference between the positive and negative classes (target sentence vs. non-target sentences).

2.3 Word Embeddings

In 2013, Mikolov et al. [19] introduced Word2Vec, a word embedding method, which changed the NLP landscape for years to come. The Word2Vec architecture leveraged deep learning coupled with a large corpus to provide rich understanding of words. This discovery sparked further research and improvements in this field [7,18,22,23]. The onset of these word embedding techniques were demonstrated to retain semantic understanding between words, which can be seen in Mikolov et al.'s [19] paper. Word2Vec introduced word-level machine understanding by proving these systems with the capability of acknowledging differences between words such as "car" and "truck", but at the same time, know that those two words are somehow related.

Computer Vision has shown deep promise with the idea of transfer learning over the past decade. Devlin et al. [11] introduced BERT, which has thus revolutionized the field of NLP, similar to what ImageNet did for Computer Vision. BERT is an architecture built on top the previously proposed Transformer by Vaswani et al. [34], which provides a general NLP model, trained on roughly 40 gigabytes of diverse text curated from throughout the web. BERT has been shown to yield state-of-the-art results on a number of benchmarks including GLUE [35], MultiNLI [38], and SQuAD [27].

2.4 BERT

In contrast to ELMo, which learns sentence representations by adopting a bidirectional approach to deep learning, Bidirectional Encoder Representations from Transformers (BERT) approaches this task by reading the entire sequence of words at once. BERT accomplishes this by extending the functionality of vanilla transformers, which is an architecture built upon two fundamental mechanisms - an encoder that reads text input and a decoder that produces the prediction for the task. It has been shown in research that BERT is most useful as a preliminary embedding step that can be fine-tuned with downstream layers for specific NLP tasks.

The goal for utilizing BERT in our research was to capture semantic representation of sentences, which would provide high level understanding for further downstream layers to compute our domain-specific task. For sake of brevity, we choose to forego an in-depth technical explanation of BERT. BERT's high dimensional output provides our system with enough contextual features to fuel the downstream architecture.

2.5 DistilBERT

Due to hardware limitations, we elected to use DistilBERT [31], a smaller general purpose language representation model, based off BERT. DistilBERT is 40%

smaller than BERT-base, retains 97% of its language understanding capabilities through various tasks, and is up to 60% faster due to the significant reduction in parameters. DistilBERT proved to be a viable alternative to BERT - it has been shown to provide effective results, comparable to BERT. For brevity, all references to BERT in this paper is using the DistilBERT implementation.

3 Model Architecture

The model we propose consists of roughly 70 millions total parameters with around 3.7 million of those being trainable. This number fluctuates widely based upon chosen hyper-parameters. We go over our chosen hyper-parameters below.

We elect to use BERT as an initial sentence encoder out of the box without fine-tuning. This design choice was made as we found during preliminary experimentation that fine-tuning BERT does not seem to provide additional performance. This is perhaps due to our specific domain-related data being very variable in terms of sentence structure and understanding. For example, one segment may look normal, whereas another may begin with a target sentence such as, "Table:".

Our proposed model's architecture consists of an initial BERT-based (i.e., any BERT or BERT variant) context encoder, followed by a dense layer, a 128 cell Bi-LSTM, an attention layer, and a final dense layer. The architecture can be seen more in Fig. 1.

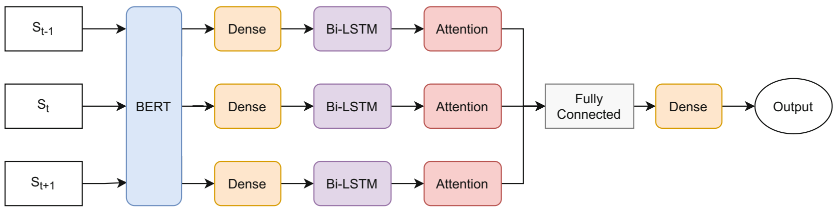


Fig. 1. Model architecture. One BERT structure is used to encode each sentence.

The first hidden dense layer acts as a feature extraction layer, mapping the highly dimensional sentence embedding to a vector with a smaller dimension. The initial dense layer is adopted from the autoencoder architecture, whereby an intermediate hidden layer handles feature extraction by reducing dimensionality. This dense layer has 128 hidden neurons, thus resulting in a reduced size of 128 to represent each sentence instead of 768. This is then sent into a Bi-directional LSTM layer, which captures longer relational structure within the sentence. The bidirectional implementation allows for the retention of important information through the sentence forwards and backwards. The hidden state hyperparameter utilized for the LSTM cells is 128. The attention layers toward the end of the network are utilized to capture the important pieces of the sentences before feeding it into the final dense layer. The attention mechanism used was proposed by Luong et al. [16] as an improvement to the originally introduced attention mechanism by Bahdanau et al. [3]. It is important to emphasize that each of the three streams are independent of each other and do not share weights. The goal is to independently achieve strong sentence encodings that are compared after the fully connected layer.

4 Dataset

We go over a few terms in this section to denote specific text segments and their contributing sentences for ease of reference within our research. We denote a text segment with S, the first sentence within a text segment (i.e., the target sentence) as s_t, and each sentence within a text segment as s_i (i.e., s_1, s_2, ..., s_n).

The dataset created by Badjatiya et al. [2] consists of samples from three very different domains, which were large and diverse enough to train their system - we elect to use these three datasets in our work. The three datasets include the following:

1. **Clinical**: A set of 227 chapters from a medical textbook. Segments are denotes as sections indicated by the author.
2. **Fiction**: A set of 85 fiction books. Each segment is a chapter break within each book.
3. **Wikipedia**: A randomly selected set of 300 documents, widely from the narrative category. The original XML dump denotes categories, which is what's used for each segment.

Due to the supervised nature of our task, careful data preparation was necessary. We decided to work with the same dataset the previous authors curated for their task, as it provided a useful spread across a variety of domains including clinical, fiction, and Wikipedia. Text segmentation as a supervised learning task can be challenging due to the inherently large segment sizes in available corpora. For example, the average segment size for the Wikipedia dataset utilized is 26 sentences long, which is a 1:26 ratio for positive to negative classes. We use a 75%–25% train to test split ratio for training and testing.

4.1 Data Preparation

All of the sentences in our dataset were reduced quite significantly, truncated to 32 words at the maximum. The intuition behind this choice was primarily focused toward prevention of overfitting along with increased performance. We tried sequence lengths of 256, 128, 64, 32, and 16. Results remained constant throughout our testing with a slight increase in precision and recall the smaller the sequence length. These findings are explained in more depth in Sect. 6

We shuffled the data segments at random. To accomplish this, we isolated each segment as its own object, collected all the those objects, and shuffled them in place. We then split the dataset into training and testing subsets. The shuffling of the dataset was necessary to represent topics evenly throughout the training and testing process. In the case where shuffling did not take place, Wikipedia entries of more recent historical events may show up in the training data, while being absent in the training data for example.

4.2 Data Augmentation

The proposed supervised learning system outlined in our research performed extremely poorly upon training with an untouched dataset. Due to the size of data needed to create an effective supervised system, we introduce a data augmentation technique specific to the text segmentation field. Our augmentation technique truncates each text segment at a certain max segment length (MSL), effectively ignoring any sentences past a certain threshold within each segment. For example, if a MSL of five was selected, every text segment within our dataset will consist of 5 or less sentences.

This step was only employed during the training phase of our system. The rationale for the introduction of this approach was to reduce the amount of imbalance between the positive and negative class (e.g., reduction from a 1:29 to a 1:5 for positive to negative classes), thus removing the amount of emphasis naturally put on the negative class. Intuitively, this augmentation approach drastically reduces the size of usable data as the removal of sentences beyond the MSL is ignored. To combat this, we fabricated new segments by associating the sentences after truncation with the previous segment (i.e., fabricated segments will have the same first sentence as the segment immediately prior). See Fig. 2 for a visual representation of how this step works. By utilizing this technique, we were able to create a more balanced dataset, while still retaining representation of sentences and topics throughout.

The inherit challenge with this technique is that it introduces a level of conflict within the training step, whereby every fabricated segment's target sentence will be immediately preceded by a sentence from the same overall text segment. We addressed this by shuffling the dataset after fabricating segments.

The chosen MSLs were arbitrary, but showed that our system could produce promising results at long and short segment sizes. We found that, without our augmented dataset approach, our system began to overfit and became more skewed toward the false prediction (i.e., 0 indicating a prediction of segment continuation, while 1 denoting a sentence being the start of a new segment). This was due to the oversampling of target sentences. For example, an original text segment consisting of 30 sentences will end up having its first sentence seen six times within training through one epoch if we choose an MSL of five. The lack of diversity in these over-sampled target sentences creates a false sense of importance, which our system tends to seek out during inference.

Although our proposed data augmentation technique provided promising results, overfitting is an eventuality. To remedy this, we also decided to explore the efficacy of utilizing our technique without the use of replicating the target sentence. In essence, instead of copying the initial target sentence at the

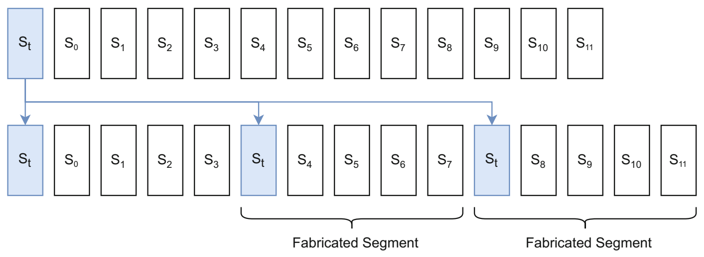


Fig. 2. The real target sentence is copied throughout the course of the segment to be used repeatedly as a fabricated target sentence for the purposes of balancing an imbalanced dataset. We shuffle the resulting segments within the global corpus to ensure sentences immediately prior to fabricated target sentences are sampled from other segments.

beginning of every subsequent segment, we decide to mark the first real sentence as the target sentence for those segments. In short, this method would take a segment of 30 sentences and make it six segments of five sentences, with each segment's first sentence being the original target sentence. This will remove our oversampling problem entirely, but could introduce a misrepresentation of target sentences throughout our dataset.

The idea of subsequently re-sampling the first sentence in every segment poses its own challenges. In a situation where our augmentation technique is used, given an average segment size of 25 sentences and a MSL of five, the beginning sentence can be re-sampled up to 5 times, introducing an obvious overfitting problem. We believe that introducing other NLP augmentation methods on the text segment's first sentence every time it is subsequently sampled (e.g., antonym replacements, synonym replacements, word embedding distances, etc.) can provide enough of a synthetic difference to reduce the overfitting challenge. Qiu et al. [26] shows the effectiveness of some of these augmentation techniques. We hope to explore this regularization technique in future works.

It is worth noting that although our data augmentation technique altered the dataset dramatically, an imbalance was still prevalent. For example, given an MSL of five, the dataset will still have a 5:1 class ratio in favor of the negative class. We chose to overcome this by using a weighted binary cross entropy loss function. Badjatiya et al. [2] showed the effectiveness of utilizing a weighted loss function in their research. This design choice introduced a faster path to overfitting due to the weighting emphasis on the positive classes, which is why we notice a performance ceiling at around 15 epochs.

5 Training

The training procedure seemed to start overfitting after roughly 15 epochs. At 30 epochs, training performance began to flat-line. However, testing performance began to see marginal improvements beyond the 5–10 epoch mark. We believe this is due to the tendency of our system to overfit on the training data over time.

BERT's parameters do not have to be trained. We used the out-of-the-box BERT configuration as our default system due to the enormous dataset and vocabulary it comes pre-trained with. The BERT authors mentioned that simple fine-tuning of 4–5 epochs on your custom dataset will give BERT the ability to understand and make meaningful embeddings. Our preliminary testing with fine-tuning BERT before our downstream model produced less accurate results by a wide margin. We suspect this is due to overfitting on larger sentences. Ultimately, we stopped this endeavor short, but would like to revisit this in future work.

Due to hardware limitations, a larger dataset could not be used. We stuck to the dataset outlined in [2] due to the variety of domain, and intra-domain topics within the respective corpora.

We also explored the use of a weighted binary cross entropy loss function, but it did not enhance the performance. We believe this is due to the lack of real target sentences. We plan on testing the possibility of using a weighted loss function along with more aggressive regularization in our future works.

5.1 Hardware Specifications

Through our tests, a variety of hyper-parameters were chosen and tested. All of the respective models were trained on a GTX 1070 graphics card with 8GB of GDDR5 memory. Each epoch took roughly 400–600 s and every model was trained for 5, 15, 30, and 40 epochs. Epoch training time decreased significantly when the MSL for sentence input was reduced. For reference, with an MSL of 32, each epoch took 100–120 s to complete.

In the future, we believe domain-specific data can be used to fine-tune our model over 5–10 epochs to make viable predictions on inference datasets. With a max sequence length of 32, fine-tuning our system with a dataset of roughly 50,000 samples would take no longer than 15 min on consumer level hardware.

6 Results

It is worth re-emphasizing that the dataset we chose to work with is extremely imbalanced, whereby roughly 95% on average (96% for Clinical, 97% for Wikipedia, and 92% for Fiction) had a ground truth class of 0. This created a large challenge in training our system to recognize and accurately predict these very infrequent text segments (i.e., predicting 1 to denote the beginning of a new segment). This is, of course, indicative of a real-world scenario, whereby a randomly sampled sentence would be much more likely to be part of a segment than the beginning of a segment.

As discussed in Sect. 4.1, the performance with varying max sequence lengths tend to lean in favor of shorter rather than longer. We elect to use 32 as the value for our testing due to the observed performance to training speed tradeoff. In addition, performance tends to degrade as the sequence length grows and training takes significantly longer.

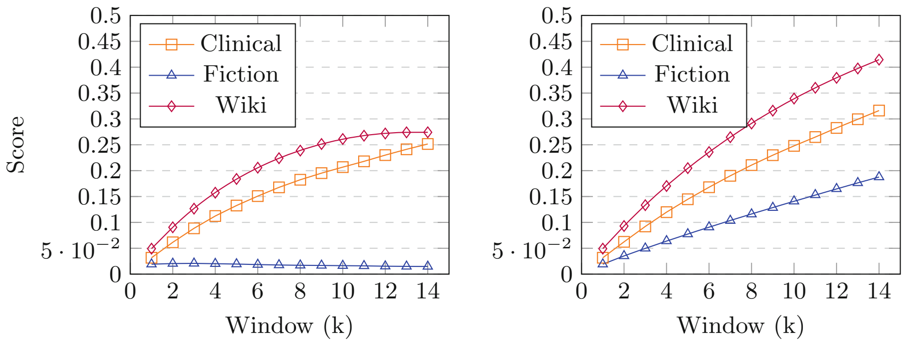
We decided to explore the use of metrics that were more indicative of the actual performance of a text segmentation system instead of the typical accuracy metric. For reference, due to the imbalanced datasets we worked with, a system that predicts 0 for every sentence would receive an accuracy score of roughly 95%. This is of course, not reflective of the actual "learning" and performance of a supervised machine learning system. For this obvious reason, we could not use accuracy as a reliable metric for evaluating our system. We are using WindowDiff [24] and P_k [5] are representative metrics as they have shown to be reliable in the text segmentation space for the past two decades. Pk was originally introduced in 1999 and WindowDiff built upon this algorithm by providing more penalization toward false positive predictions. Due to P_k's leniency towards false positives, WindowDiff scores are usually higher by comparison (higher values indicate poorer performance).

In a task like text segmentation, classifications for the wrong class can be forgiving when it is a false positive. Some results that show our system's eagerness to over-predict can be seen in Table 1. This is untrue however for the mirror case, whereby an imperfect system should ideally be better at predicting the false positive class

We begin to see a somewhat linear growth in results as the window size grows. Suggested in the original [5] P_k paper, the optimal k-value that should be used is half the average segment size - the Wikipedia dataset has an average of 25.97 words. The larger the k-value, the more strict the penalization factor is when observing a false positive within the sliding window. For example, Figs. 3a and 3b illustrate this clearly.

Table 1. Sample results from Wikipedia prediction. The model as trained tends to over-predict. In this case, the model predicted false positives for lines 3–5. We believe this is due to no matching words in the sentences before and after. E.g., for sentence 3, the word "Duryodhana", which might be interpreted as an important feature, is not seen in the sentence before or after. $\hat{\boldsymbol{y}}$ is the predicted value, whereas $\boldsymbol{y}$ is the true value.

Text	$\hat{y}$	y
In Villiputuralvar's 14th-century version, Krishna...	0	0
There is no mention of Duryodhana in this version...	1	0
In other accounts, Aravan is sacrificed in order t...	1	0
In the traditions of the village of Neppattur, in ...	1	0
So Krishna prescribes the human sacrifice of Arava...	0	0
This allows Aravan to make the initial sacrifice o...	0	0
Gattis grew up in Forney, Texas, and began playing...	1	1
His parents divorced when he was eight years old, ...	0	0
Busy playing baseball, Gattis never processed his ...	0	0
Gattis played for the Dallas Tigers, one of the pr...	0	0



(a) $\mathrm{P_k}$ **(b)** WindowDiff

Fig. 3. P_k and WindowDiff Results.

Our results show a considerable improvement over Badjatiya et al.'s [2] work, where their attention based neural model achieved roughly 0.32, 0.34, and 0.38 for clinical, wikipedia, and fiction datasets respectively using the P_k metric. Our results show sub 0.3 results for all these datasets throughout a similar training pattern and identical window sizes.

7 Limitations

All of the models we tested were trained on an Nvidia GTX 1070 graphics card with 8 GB of GDDR5 memory. While we were capable of training the system at a reduced scale, increasing hyper-parameters for longer training stints became challenging. For example, increasing the sequence length from 64 to 128 and the dense layer's hidden neurons would significantly impact the required training time. The LSTM cell size in particular (i.e., increasing from 128 to 256 or 512) inflated the required training time. Utilizing a Gated Recurrent Unit (GRU) cell instead of LSTM was not an option as it does not return the hidden state as it is necessary for the following attention layer.

Due to hardware limitations, dataset size was critical in the training and benchmarking of our system. We elected to work with static datasets with domain-specific verbiage and terminologies. For example, the clinical dataset included medical terms which would usually be considered out-of-vocabulary (OOV) in systems with vocabulary restrictions. Because we are using BERT as our encoding strategy, most of these words are taken care of gracefully by Word-Piece, a sub-word algorithm proposed by Wu et al. [39], which builds off the Byte Pair Encoding (BPE) strategy [32]. Unfortunately, the generic dataset used to pre-train BERT does not include many utterances of these rarer words and as of such, lacks a richer domain-specific linguistic understanding. While the BERT authors have shown that fine-tuning BERT on the dataset being worked with yields favorable results, our preliminary analysis testing this resulted in little to no improvement. A larger dataset and longer training times could be beneficial for fine-tuning BERT in this setting.

8 Future Work

Although our work is very preliminary, it opens the door to a wide variety of future possibilities - most of which we hope to explore. Drawing inspiration from Badjatiya et al.'s [2] work, we kept a 3 sentence approach to our system. We hypothesize that with the increased contextual leverage, a 4 or 5 sentence approach could significantly improve the results of our system. This would deepen the contextual understanding of the system by providing more samples per record. We can see based on Table 1 that in certain situations, important words may not show up in the sentences immediately before or after, but in sentences two orders of magnitude away.

We also believe that introducing other NLP augmentation methods on the text segment's first sentence every time it is subsequently sampled (e.g., back translation, antonym replacements, synonym replacements, etc.) can provide enough of a synthetic difference to reduce the overfitting challenge. Qiu et al. [26] shows the effectiveness of some of these augmentation techniques.

In recent years, lots of breakthrough success has come from multi-modal approaches, whereby a system will take multiple types of inputs. For example, Adobe has shown that novel Document Segmentation success can be achieved by feeding both text and image based data into a system instead of one or the other [42]. We believe that including more sentence topical information as additional input features into the system can help it learn similarities and differences. For example, utilizing Topic Modelling techniques such as LDA in combination with BERT, has been shown to be an effective combination by Shoa [33].

Through a Siamese Network learning technique, more semantically-aware BERT-based sentence vectors can be generated in works done by [28]. Substituting the initial BERT encoding with these vectors not only reduces dimensionality significantly, but also has proven to demonstrate a higher level of semantic understanding. In a similar vein, Angelov et al. [1] created a topic-based implementation using BERT, which has proved to be highly accurate when it comes to identifying topics due to BERT's contextual nature.

9 Conclusion

In conclusion, our research aimed to provide a sentence-wise approach to text segmentation by utilizing modern NLP techniques. We show that BERT and BERT-based architectures (namely DistilBERT) result in measurable improvements within specific domains, given a sufficient amount of training data. Our research shows that our approach holds true for three completely different domains (fiction, wikipedia, clinical). Fiction outperformed the other two domains, giving us hope that our approach can be extended to similar structure-based documents in other domains, such as legal and accounting.

Acknowledgements. The second author thanks the support of an NSERC Discovery Grant.

References

1. Angelov, D.: Top2Vec: distributed representations of topics. arXiv:2008.09470 [cs, stat], August 2020)
2. Badjatiya, P., Kurisinkel, L.J., Gupta, M., Varma, V.: Attention-based neural text segmentation. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) ECIR 2018. LNCS, vol. 10772, pp. 180–193. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76941-7_14
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 [cs, stat], May 2016
4. Barrow, J., Jain, R., Morariu, V., Manjunatha, V., Oard, D., Resnik, P.: A joint model for document segmentation and segment labeling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 313–322. Association for Computational Linguistics, July 2020. https://doi.org/10.18653/v1/2020.acl-main.29, https://www.aclweb.org/anthology/2020.acl-main.29
5. Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. Mach. Learn. **34**(1), 177–210 (1999). https://doi.org/10.1023/A:1007506220214
6. Blei, D.M.: Latent Dirichlet Allocation, p. 30
7. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
8. Chaudhary, A.: A visual survey of data augmentation in NLP, May 2020. https://amitness.com/2020/05/data-augmentation-for-nlp/
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
10. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. arXiv:cs/0003083, March 2000
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 [cs], May 2019
12. Eisenstein, J., Barzilay, R.: Bayesian unsupervised topic segmentation. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 334–343. Association for Computational Linguistics, Honolulu, October 2008. https://www.aclweb.org/anthology/D08-1035
13. Hearst, M.A.: TextTiling: a quantitative approach to discourse segmentation. Technical report (1993)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Koshorek, O., Cohen, A., Mor, N., Rotman, M., Berant, J.: Text segmentation as a supervised learning task. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 469–473. Association for Computational Linguistics, New Orleans, June 2018. https://doi.org/10.18653/v1/N18-2075, https://www.aclweb.org/anthology/N18-2075
16. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv:1508.04025 [cs], September 2015
17. Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge (2000).Google-Books-ID: VyjED9VOn5MC

18. McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: contextualized word vectors. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 6294–6305. Curran Associates, Inc. (2017). http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 [cs], September 2013
20. Misra, H., Yvon, F., Jose, J.M., Cappe, O.: Text segmentation via topic modeling: an analytical study. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. CIKM 2009, pp. 1553–1556. Association for Computing Machinery, New York, November 2009. https://doi.org/10.1145/1645953.1646170
21. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, March 2016. https://ojs.aaai.org/index.php/AAAI/article/view/10350, number: 1
22. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, October 2014. https://doi.org/10.3115/v1/D14-1162, https://www.aclweb.org/anthology/D14-1162
23. Peters, M.E., et al.: Deep contextualized word representations. arXiv:1802.05365 [cs], March 2018
24. Pevzner, L., Hearst, M.A.: A critique and improvement of an evaluation metric for text segmentation. Comput. Linguist. **28**(1), 19–36 (2002)
25. Purver, M., Körding, K.P., Griffiths, T.L., Tenenbaum, J.B.: Unsupervised topic modelling for multi-party spoken discourse. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 17–24. Association for Computational Linguistics, Sydney, July 2006. https://doi.org/10.3115/1220175.1220178, https://www.aclweb.org/anthology/P06-1003
26. Qiu, S., et al.: EasyAug: an automatic textual data augmentation platform for classification tasks. In: Companion Proceedings of the Web Conference 2020. WWW 2020, pp. 249–252. Association for Computing Machinery, New York, April 2020. https://doi.org/10.1145/3366424.3383552
27. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text, June 2016. https://arxiv.org/abs/1606.05250v3
28. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs], August 2019
29. Riedl, M., Biemann, C.: TopicTiling: a text segmentation algorithm based on LDA. In: Proceedings of ACL 2012 Student Research Workshop, pp. 37–42. Association for Computational Linguistics, Jeju Island, July 2012. https://www.aclweb.org/anthology/W12-3307
30. Rumelhart, D.E., Mcclelland, J.L.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. Foundations (1986)
31. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs], February 2020
32. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv:1508.07909 [cs], June 2016
33. Shoa, S.: Contextual Topic Identification: Identifying meaningful topics for sparse Steam reviews, March 2020. Publication Title: Medium

34. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017). http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
35. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: a multi-task benchmark and analysis platform for natural language understanding, April 2018. https://arxiv.org/abs/1804.07461v3
36. Wang, W.Y., Yang, D.: That's so annoying!!!: a lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2557–2563. Association for Computational Linguistics, Lisbon, September 2015. https://doi.org/10.18653/v1/D15-1306, https://www.aclweb.org/anthology/D15-1306
37. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks, January 2019. https://arxiv.org/abs/1901.11196v2
38. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference, April 2017. https://arxiv.org/abs/1704.05426v4
39. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv:1609.08144 [cs], October 2016
40. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv:1904.12848 [cs, stat], November 2020
41. Yang, H.: BERT meets Chinese word segmentation, September 2019. https://arxiv.org/abs/1909.09292v1
42. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks, pp. 5315–5324 (2017). https://openaccess.thecvf.com/content_cvpr_2017/html/Yang_Learning_to_Extract_CVPR_2017_paper.html
43. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. arXiv:1710.09412 [cs, stat], April 2018
44. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. arXiv:1509.01626 [cs], April 2016