Training and Domain Adaptation for Supervised Text Segmentation

Goran Glavaš[1] Ananya Ganesh[2] Swapna Somasundaran[2]

[1]Data and Web Science Group, University of Mannheim
goran@informatik.uni-mannheim.de

[2]Educational Testing Service
{aganesh002,ssomasundaran}@ets.org

Abstract

Unlike traditional unsupervised text segmentation methods, recent supervised segmentation models rely on Wikipedia as the source of large-scale segmentation supervision. These models have, however, predominantly been evaluated on the in-domain (Wikipedia-based) test sets, preventing conclusions about their general segmentation efficacy. In this work, we focus on the domain transfer performance of supervised neural text segmentation in the educational domain. To this end, we first introduce K12SEG, a new dataset for evaluation of supervised segmentation, created from educational reading material for grade-1 to college-level students. We then benchmark a hierarchical text segmentation model (HITS), based on RoBERTa, in both in-domain and domain-transfer segmentation experiments. While HITS produces state-of-the-art in-domain performance (on three Wikipedia-based test sets), we show that, subject to the standard full-blown fine-tuning, it is susceptible to domain overfitting. We identify adapter-based fine-tuning as a remedy that substantially improves transfer performance.

1 Introduction

Organizing long texts into coherent segments facilitates human text comprehension as well as downstream tasks like text summarization (Angheluta et al., 2002; Bokaei et al., 2016), passage retrieval (Huang et al., 2003; Prince and Labadié, 2007; Shtekh et al., 2018), and sentiment analysis (Xia et al., 2010; Li et al., 2020). Text segmentation is very important in the educational domain as it enables large-scale passage extraction. Educators, for example, need to extract coherent passage segments from books to create reading materials for students. Similarly, test developers must create reading assessments at scale by extracting coherent segments from a variety of sources.

Most segmentation models allow for (i.e., sequential) segmentation (Hearst, 1994; Choi, 2000; Riedl and Biemann, 2012; Glavaš et al., 2016; Koshorek et al., 2018; Glavaš and Somasundaran, 2020), though methods for hierarchical segmentation have been proposed as well (Eisenstein, 2009; Du et al., 2013; Bayomi and Lawless, 2018). Owing to the absence of large annotated datasets, (linear) text segmentation has long been limited to unsupervised models, relying on various measures of lexical and semantic sentence overlap (Hearst, 1994; Choi, 2000; Utiyama and Isahara, 2001; Fragkou et al., 2004; Glavaš et al., 2016) and topic modeling (Brants et al., 2002; Misra et al., 2009; Riedl and Biemann, 2012). More recently, Koshorek et al. (2018) automatically created a large segment-annotated dataset WIKI727 by leveraging the headings structure in Wikipedia articles, effectively enabling supervised text segmentation; they then trained a hierarchical recurrent neural segmentation model on WIKI727. In subsequent work, Glavaš and Somasundaran (2020) improved on their segmentation performance by (i) replacing recurrent components of the hierarchical model with transformer networks (Vaswani et al., 2017) and (ii) adding an auxiliary self-supervised coherence objective. Although both Koshorek et al. (2018) and Glavaš and Somasundaran (2020) report massive gains over unsupervised baselines, their models have mostly been subject to *in-domain* evaluation on test sets also derived from Wikipedia.

In this work, in contrast, we concern ourselves with domain transfer of supervised text segmentation models, with a focus on the educational domain. To investigate the effects of domain transfer in supervised text segmentation, we first introduce K12SEG – a segment-annotated dataset which we automatically created from educational texts designed for grade-1 to college-level student population. We then benchmark a hierarchical neural

Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 110–116
April 20, 2021 ©2021 Association for Computational Linguistics

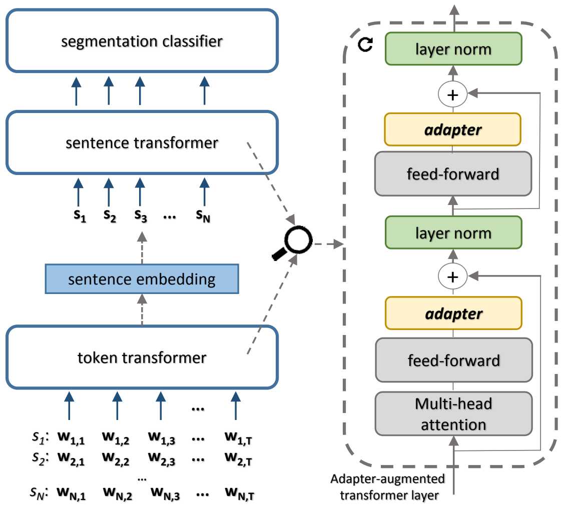
segmentation classifier
sentence transformer
s_1 s_2 s_3 ... s_N
sentence embedding
token transformer
s_1: $w_{1,1}$ $w_{1,2}$ $w_{1,3}$ ... $w_{1,T}$
s_2: $w_{2,1}$ $w_{2,2}$ $w_{2,3}$ ... $w_{2,T}$
...
s_N: $w_{N,1}$ $w_{N,2}$ $w_{N,3}$ ... $w_{N,T}$
layer norm
+
adapter
feed-forward
layer norm
+
adapter
feed-forward
Multi-head attention
Adapter-augmented transformer layer

Figure 1: Architecture of the adapter-augmented hierarchical model for supervised text segmentation.

text segmentation model (HITS) in a range of in-domain and domain-transfer segmentation experiments involving WIKI727 (Koshorek et al., 2018) and our new K12SEG dataset.

Our HITS model, illustrated in Figure 1, though similar to the hierarchical segmentation models of Glavaš and Somasundaran (2020), differs in two crucial aspects. First, we initialize the parameters of the lower (token-level) transformer with the weights of the pretrained RoBERTa (Liu et al., 2019). Secondly, aiming to prevent both (1) forgetting of distributional information captured in RoBERTa's parameters and (2) overfitting to the training domain – we augment the layers of the token-level transformer with adapter parameters (Rebuffi et al., 2018; Houlsby et al., 2019) before segmentation training. Adapter-based fine-tuning only updates the additional adapter parameters and the original transformer parameters are frozen: this fully preserves the distributional knowledge obtained in transformer's pretraining. Encoding out-of-domain segmentation knowledge (e.g., from the WIKI727 dataset) separately from the distributional information (original RoBERTa parameters) allows to combine the two types of information more flexibly during the secondary segmentation training in the target domain (e.g., on K12SEG), resulting in more effective domain transfer.

Experimental results confirm the above expectations. Our adapter-augmented HITS model trained on WIKI727, besides yielding state-of-the-art in-domain (Wikipedia) segmentation performance, facilitates domain transfer and leads to substantial gains in our target educational domain (K12SEG).

2 Segmentation Model

Hierarchical Transformer-Based Model. Our base segmentation model (Figure 1) consists of two hierarchically linked transformers: the lower transformer contextualizes tokens within sentences and yields sentence embeddings; the upper transformer then contextualizes sentence representations. An individual training instance is a sequence of N sentences, $\{s_1, \ldots, s_N\}$, each consisting of T (subword) tokens, $s_i = \{w_{i,1}, \ldots, w_{i,T}\}$. We initialize the lower transformer with the pretrained RoBERTa weights (Liu et al., 2019).[1] We then use the transformed vector of the sentence start token (<s>), $\mathbf{s}_i$, as the embedding of the sentence s_i. The purpose of the randomly initialized upper sentence-level transformer is to contextualize the sentences in the sequence with one another. Let $\mathbf{x}_i$ be the transformed representation of the sentence s_i, produced by the upper transformer. The segmentation prediction for the sentence s_i is then made by the simple feed-forward softmax classifier: $\mathbf{y}_i = \mathit{softmax}\left(\mathbf{W}\mathbf{x}_i + \mathbf{b}\right)$. We minimize the binary cross-entropy loss.

Adapter-Based Training. Unlike Koshorek et al. (2018); Glavaš and Somasundaran (2020), we initialize the lower transformer with RoBERTa weights, encoding general-purpose distributional knowledge. Full fine-tuning, in which all transformer parameters are updated in downstream training, may overwrite useful distributional signal with domain-specific artefacts, overfit the model to the training domain, and impede domain transfer for the downstream task (Rücklé et al., 2020). Alternative, adapter-based fine-tuning (Rebuffi et al., 2018; Houlsby et al., 2019; Pfeiffer et al., 2020), injects additional adapter parameters into transformer layers and updates only them during downstream fine-tuning, keeping the original transformer parameters unchanged. We adopt the bottleneck adapter architecture of Houlsby et al. (2019), reported effective for a wide range of downstream task. Concretely, in each layer of the lower transformer, we insert two bottleneck adapters: one after the multi-head attention sublayer and another after the feed-forward sublayer. Let $\mathbf{X} \in \mathbb{R}^{T \times H}$ stack contextualized vectors for the sequence of T tokens in one of the transformer layers, input to the adapter layer. The adapter then yields the following output:

[1] We use the 12-layer *Base* RoBERTa model

$$\mathbf{X}' = \mathbf{X} + g\left(\mathbf{X}\mathbf{W}_d + \mathbf{b}_d\right)\mathbf{W}_u + \mathbf{b}_u.$$

The parameter matrix $\mathbf{W}_d \in \mathbb{R}^{H \times a}$ down-projects the token vectors from $\mathbf{X}$ to the *adapter size* $a < H$, and $\mathbf{W}_u \in \mathbb{R}^{a \times H}$ up-projects the activated down-projections back to transformer's hidden size H; g is the non-linear activation function.

Training Instances and Inference. We train the model with sequences of N sentences as instances which we create by sliding the window of size N over document's sentences, with a sliding step of $N/2$. At inference, for each sentence s, we make predictions for all of the windows that contain s. This means that we obtain (at most) N segmentation probabilities for each sentence (for the i-th sentence, we get predictions from windows $[i{-}N{+}1 : i]$, $[i{-}N{+}2 : i{+}1], \ldots, [i : i{+}N{-}1]$). We average the sentence's segmentation probabilities obtained across different windows and predict that it starts a new segment if the average is above the threshold t. We treat the sequence length N and threshold t as hyperparameters and optimize them using the development datasets. For brevity, we describe the optimization details in the Appendix.

3 Evaluation

3.1 Data

WIKI727. To the best of our knowledge, WIKI727 (Koshorek et al., 2018) is the only large segment-annotated dataset designed for supervised text segmentation. It consists of 727K Wikipedia articles (train portion: 582K articles), automatically segmented according to the articles' heading structure.

K12SEG. To empirically evaluate domain transfer in supervised text segmentation, we introduce a new dataset, dubbed K12SEG, created from educational reading material designed for grade-1 to college-level students (Zeno et al., 1995). The original dataset, the *Educators Word Frequency*, was created by standardized sampling of reading materials from a variety of content areas (e.g. science, social science, home economics, fine arts, health, business etc.). Each sample is 250-325 words long. We create one synthetic K12SEG instance by selecting and concatenating two samples from (a) the same book, (b) different books from the same content area (e.g., science) or (c) different books from different content areas. In contrast to WIKI727, in which the number and sizes of segments greatly vary across Wikipedia articles, K12SEG documents are more uniform: with two segments (samples) each and minor variation in sentence length (mean: 30 sentences). Besides the different genre between WIKI727 and K12SEG, this stark difference between their distributions of segment numbers and sizes poses an additional challenge for the domain transfer. We split the total of 18,906 K12SEG documents into train (15,570 documents), development (3,000), and test portions (336). An example 2-segment document from from K12SEG is shown in Table 1.

Wikipedia-Based Test Sets. For the in-domain (Wikipedia) evaluation, we use three small-sized test sets. WIKI50 is an additional test set consisting of 50 documents, created by Koshorek et al. (2018) in the same manner as WIKI727. Chen et al. (2009) similarly created the CITIES (64 articles) and ELEMENTS (117) from Wikipedia pages of world cities and chemical elements, respectively.

3.2 Experimental Setup

Experiments. We carry out two sets of experiments. We first benchmark the performance of our HITS model "in domain", i.e., by training it on WIKI727 and evaluating it on WIKI50, ELEMENTS, and CITIES. Here we directly compare HITS (with full and adapter-based fine-tuning) with state-of-the-art segmentation models: the hierarchical Bi-LSTM (HBi-LSTM) model of Koshorek et al. (2018), and two hierarchical transformer variants of Glavaš and Somasundaran (2020) – with (CATS) and without (TLT-TS) the auxiliary coherence objective. The second set of experiments quantifies the efficacy of HITS in transfer for the educational domain. We compare the performance of "in-domain" training on K12SEG with transfer strategies: (i) *zero-shot transfer*: HITS variants (full and adapter-based fine-tuning) trained on WIKI727 and evaluated on the K12SEG test set and (ii) *sequential training*: HITS variants sequentially trained first on WIKI727 and then on the train portion of K12SEG.

Training and Optimization Details. We initialize the weights of the lower transformer network in all HITS variants with the pretrained RoBERTa Base model, having $L_L = 12$ layers (with 12 attention heads each) and hidden representations

First segment	Second segment
Traveling familiar routes in our family cars we grow so accustomed to crossing small bridges and viaducts that we forget they are there. We have to stop and think to remember how often they come along. Only when a bridge is closed for repairs and we have to take a long detour do we realize how difficult life would be without it. Try to imagine our world with all the bridges removed. In many places life would be seriously disrupted, traffic would be paralyzed, and business would be terrible. Bridges bring us together and keep us together. They are a basic necessity for civilization. The first structures human beings built were bridges. Before prehistoric people began to build even the crudest shelter for themselves, they bridged streams. Early prehistoric tribes were wanderers. Since they did not stay in one place they did not think of building themselves houses. But they could not wander far without finding a stream in their way. Nature provided the first bridges. Finding themselves confronted with some narrow but rapid river, humans noticed a tree that had fallen across the river from bank to bank. The person who first scrambled across a fallen log, perhaps after watching monkeys run across it, was the first human being to cross a bridge. Eventually, when they had learned how to chop down a tree, they also learned how to make a tree fall in the direction they wanted it to fall. The wandering tribe that first deliberately made a tree fall across a stream were the first bridge-builders.	Working in the mud and water of a river bottom was difficult and dangerous. People were often crushed or maimed by the pile driver or the piles. But the work on the foundations is the most important part of bridge-building. The part of a bridge that is underwater, the part we never see, is more important than the part we do see, because no matter how well made the superstructure may be, if the foundation is not solid the bridge will fall. Not only did the pier foundations have to be solid, they also had to be protected as much as possible from wear. A flowing river constantly stirs up the bottom, so that the water's lower depths are a thick soup filled with mud and sand and pebbles which grind against anything in the path of the current. This action is called scour. to reduce the wear and tear of the current, the Romans built the fronts of their piers in the shape of a boat's prow. The Romans used only one kind of arch, the semicircular. The arch describes a full half-circle from pier to pier. Each end of the half-circle rests on a pier, and the two piers will hold the arch up by themselves, even before the rest of the bridge is built, provided each pier is at least one third as thick as the width of the arch. Thus a bridge could be built one arch at a time, and if the work had to stop the partial structure would stay in place until work could be resumed. The Romans built their bridges during the summer and fall, when the weather was best and the water level was generally lowest, and stopped during winter and spring.

Table 1: An example 2-segment document from the K12SEG dataset.

of size $H = 768$. Our upper-level transformer for sentence contextualization has $L_U = 6$ layers (with 6 attention heads each), and the same hidden size $H = 768$. We apply a dropout ($p = 0.1$) on the outputs of both the lower and upper transformer outputs. In adapter-based fine-tuning we set the adapter size to $a = 64$ and use GeLU (Hendrycks and Gimpel, 2016) as the activation function. In all experiments, we limit the sentence input to $T = 128$ subword tokens (shorter sentences are padded, longer sentences trimmed). We optimize models' parameters using the Adam algorithm (Kingma and Ba, 2015) with the initial learning rate of 10^{-5}. We train for at most 30 epochs over the respective training set (WIKI727 or K12SEG), with early stopping based on the loss on the respective development set. We train in batches of 32 instances (i.e., 32 sequences of N sentences) and have found (via cross-validation on respective development sets) the optimal sequence length to be $N = 16$ sentences and the optimal average segmentation probability threshold at inference time to be $t = 0.35$.

3.3 Results and Discussion

We report the results in terms of P_K, the standard evaluation metric for text segmentation (Beeferman et al., 1999). P_K is the percentage of wrong predictions on whether or not the first and last sentence in a sequence of K consecutive sentences belong to the same segment. As in previous work (Koshorek et al., 2018; Glavaš and Somasundaran, 2020), we set K to one half of the average gold-standard segment size of the evaluation dataset.

In-Domain Wikipedia Evaluation. We report the results of the in-domain Wikipedia evaluation on WIKI50, CITIES, and ELEMENTS in Table 2. Our HITS model variants, which we start training with the pretrained RoBERTa as the token-level transformer, outperform the hierarchical neural models from (Koshorek et al., 2018) and (Glavaš

Model	Fine-tuning	WIKI50	CITIES	ELEM.
HBi-LSTM	–	18.24	19.68	41.63
TLT-TS	–	17.47	19.21	20.33
CATS	–	16.53	16.85	18.41
HITS (ours)	Full	**14.50**	15.03	17.06
HITS (ours)	Adapter	15.17	**14.11**	**14.67**

Table 2: "In-domain" performance of hierarchical neural segmentation models, trained on the large WIKI727 dataset, on three Wikipedia-based test sets (smaller values of the error measure P_K mean better performance).

and Somasundaran, 2020), which start from a randomly initialized token-level encoder. This is consistent with findings from many other tasks: fine-tuning pretrained transformers yields better results than task-specific training from scratch, even if the dataset is large (as is the case with WIKI727). Full fine-tuning produces better results on WIKI50, whereas adapter-based fine-tuning exhibits stronger performance on CITIES and ELEMENTS. Since the articles in WIKI50 come from a range of Wikipedia categories, much like in the training set WIKI727, whereas CITIES and ELEMENTS each contain articles from a single category, we believe these results already indicate that full fine-tuning is more prone to domain (genre, topic) overfitting than adapter-based tuning. Remarkably, HITS (Full) surpasses the human WIKI50 performance, reported to stand at 14.97 P_K points (Koshorek et al., 2018).

Domain Transfer Results. Table 3 shows the performance of both in-domain and transferred HITS model variants on the K12SEG test set. Interestingly, with Full fine-tuning, we observe the same performance regardless of whether we train the model on the out-of-domain (but much larger) WIKI727 dataset or the (smaller) in-domain K12SEG training set. More interestingly, adapter-based fine-tuning in the zero-shot domain transfer yields better performance than in-domain adapter fine-tuning. Poor performance of in-domain training could mean that K12SEG is either (a) insufficiently large or (b) contains such versatile segmentation examples over which it is hard to generalize. Gains from *sequential* domain transfer, in which the model is exposed to exactly the same K12SEG training set but only *after* it was trained on a much larger out-of-domain WIKI727 dataset, point to (a) as the more likely explanation. In both in-domain and zero-shot setups, adapter-based fine-tuning produces better segmentation than full fine-tuning, contributing to the conclusion that adapter-based fine-tuning curbs overfitting to domain-specific artefacts, improving the model's generalization ability. Finally, the sequential training in which we freeze the lower transformer's parameters (including the adapters) during the (secondary) in-domain training, gives the best result overall. We speculate that the relatively small K12SEG train set gives the advantage to the model variant that uses that limited-size data to fine-tune fewer parameters (i.e., only the upper, sentence-level transformer).

Setup	Fine-tuning	Freeze	K12SEG (test)
In domain	Full	–	25.5
	Adapter	–	23.9
Zero-shot	Full	–	25.5
	Adapter	–	20.7
Sequential	Full	No	12.9
	Full	Yes	14.8
	Adapter	No	13.3
	Adapter	Yes	**10.4**

Table 3: Segmentation performance in domain transfer. Evaluation on K12SEG test set. *In domain*: training on the K12SEG train set; *Zero-shot*: training on the train portion of WIKI727; *Sequential*: sequential training, first on WIKI727 and then on the train portion of K12SEG. For *Sequential* training, the column *Freeze* specifies whether the the lower transformer's parameters were frozen during secondary, in-domain fine-tuning (on the train portion of K12SEG).

4 Conclusion

Supervised text segmentation has been limited to a single large-scale segmentation dataset, WIKI727, built automatically from Wikipedia. In this work, we studied domain transfer for supervised text segmentation: we introduce K12SEG, a new dataset for supervised text segmentation built from educational reading materials (grade-1 to college-level students), and use it together with WIKI727 in our domain transfer experiments. Our hierarchical segmentation model (HITS) couples the pretrained RoBERTa with the upper-level transformer that provides sentence contextualization. We show that HITS obtains state-of-the-art performance on standard Wikipedia-based evaluation datasets, but overfits to the training domain (Wikipedia). We finally substantially improve model's transfer capabilities through adapter-based fine-tuning.

References

Roxana Angheluta, Rik De Busser, and Marie-Francine Moens. 2002. The use of topic segmentation for automatic summarization. In *Proc. of the ACL-2002 Workshop on Automatic Summarization*, pages 11–12.

Mostafa Bayomi and Séamus Lawless. 2018. C-hts: A concept-based hierarchical text segmentation approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1-3):177–210.

Mohammad Hadi Bokaei, Hossein Sameti, and Yang Liu. 2016. Extractive summarization of multi-party meetings through discourse segmentation. *Natural Language Engineering*, 22(1):41–72.

Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. 2002. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proc. of CIKM*, pages 211–218. ACM.

Harr Chen, SRK Branavan, Regina Barzilay, and David R Karger. 2009. Global models of document structure using latent permutations. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 371–379. Association for Computational Linguistics.

Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Lan Du, Wray Buntine, and Mark Johnson. 2013. Topic segmentation with a structured topic model. In *Proc. of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proc. of HLT-NAACL*, pages 353–361. Association for Computational Linguistics.

Pavlina Fragkou, Vassilios Petridis, and Ath Kehagias. 2004. A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information Systems*, 23(2):179–197.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. In *Proc. of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130.

Goran Glavaš and Swapna Somasundaran. 2020. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In *Proceedings of the The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7797–7804.

Marti A Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (GELUs). *CoRR*, abs/1606.08415.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799.

Xiangji Huang, Fuchun Peng, Dale Schuurmans, Nick Cercone, and Stephen E Robertson. 2003. Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, 6(3-4):333–362.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473.

Jing Li, Billy Chiu, Shuo Shang, and Ling Shao. 2020. Neural text segmentation and its application to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hemant Misra, François Yvon, Joemon M Jose, and Olivier Cappe. 2009. Text segmentation via topic modeling: An analytical study. In *Proc. of CIKM*, pages 1553–1556. ACM.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Violaine Prince and Alexandre Labadié. 2007. Text segmentation based on document understanding for information retrieval. In *International Conference on Application of Natural Language to Information Systems*, pages 295–304. Springer.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In *CVPR*.

Martin Riedl and Chris Biemann. 2012. Topictiling: a text segmentation algorithm based on lda. In *Proc. of ACL 2012 Student Research Workshop*, pages 37–42. Association for Computational Linguistics.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020. Multicqa: Zero-shot transfer of self-supervised text matching models on a massive scale.

Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. 2018. Exploring influence of topic segmentation on information retrieval quality. In *International Conference on Internet Science*, pages 131–140. Springer.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proc. of ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Huosong Xia, Min Tao, and Yi Wang. 2010. Sentiment text classification of customers reviews on the web based on svm. In *2010 Sixth International Conference on Natural Computation*, volume 7, pages 3633–3637. IEEE.

Susan Zeno, Stephen H Ivens, Robert T Millard, and Raj Duvvuri. 1995. *The educator's word frequency guide*. Touchstone Applied Science Associates.