Lessons Learnt from Linear Text Segmentation: a Fair Comparison of Architectural and Sentence Encoding Strategies for Successful Segmentation

Iacopo Ghinassi
Queen Mary University of London / London, UK
i.ghinassi@qmul.ac.uk

Lin Wang
Queen Mary University of London / London, UK
lin.wang@qmul.ac.uk

Chris Newell
BBC R&D / London, UK
chris.newell@bbc.co.uk

Matthew Purver
Queen Mary University of London / London, UK
Institut Jožef Stefan / Ljubljana, Slovenia
m.purver@qmul.ac.uk

Abstract

Recent works on linear text segmentation have shown new state-of-the-art results nearly every year. Most times, however, these recent advances include a variety of different elements which makes it difficult to evaluate which individual components of the proposed methods bring about improvements for the task and, more generally, what actually works for linear text segmentation. Moreover, evaluating text segmentation is notoriously difficult and the use of a metric such as P_k, which is widely used in existing literature, presents specific problems that complicates a fair comparison between segmentation models. In this work, then, we draw from a number of existing works to assess which is the state-of-the-art in linear text segmentation, investigating what architectures and features work best for the task. For doing so, we present three models representative of a variety of approaches, we compare them to existing methods and we inspect elements composing them, so as to give a more complete picture of which technique is more successful and why that might be the case. At the same time, we highlight a specific feature of P_k which can bias the results and we report our results using different settings, so as to give future literature a more comprehensive set of baseline results for future developments. We then hope that this work can serve as a solid foundation to foster research in the area, overcoming task-specific difficulties such as evaluation setting and providing new state-of-the-art results[1].

[1]code available at: https://github.com/Ighina/NSE-TopicSegmentation

1 Introduction

Linear text segmentation, also known as topic segmentation, is a well known problem in natural language processing, and the first step for a number of downstream applications. The task consists in the automatic segmentation of a text into topically coherent units and this has many use cases: a long transcript from a news show, e.g., could be divided into single news stories so as to help an end user in retrieving more relevant and specific information (Reynar, 1999) or a long article could be divided into subsections to aid its reading (Hearst, 1997).

Recent works have presented a series of advancements in the field, from which a number of conclusions could be drawn, such as the fact that Transformer architectures work better than traditional recurrent models (Lo et al., 2021) and that fine-tuned LLMs need no additional contextual information to perform the task (Lee et al., 2023).

The results of different recent works, however, can be contradictory and not pointing towards a clear direction forward in terms of what works and what does not in text segmentation. Part of the reason for this, we show, is the fact that existing and popular metrics such as P_k (Beeferman et al., 1999) might lead to very different results under different conditions and, therefore, the final results from which to draw our conclusions are unstable.

Based on this, we draw on existing literature to present our own topic segmentation models. We show that carefully designed recurrent neural networks are still relevant in the field as they can obtain state-of-the-art results in most occasions given a fixed and fair evaluation setting. We draw conclusions on why this might be the case and we show

Proceedings of Recent Advances in Natural Language Processing, pages 408–418
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_046

that this evidence makes sense given previous literature on the subject.

2 Related Work

2.1 Models for Topic Segmentation

Traditionally, text segmentation involves the segmentation of text like books or articles (Beeferman et al., 1999; Koshorek et al., 2018), business meeting or TV news transcripts (Misra et al., 2010; Purver et al., 2006; Sehikh et al., 2018).

An early text segmentation system, TextTiling, used two adjacent sliding windows over sentences and compared the two by means of cosine similarity between the relative bag-of-words vector representations (Hearst, 1994). The same algorithm was then successfully used with different, more informative sentence representations, such as Term-Frequency Inverse-Document-Frequency (TF-IDF) rescoring of bag-of-words (Galley et al., 2003) and features derived from generative topic models like Latent Dirichlet Allocation (LDA, Riedl and Biemann, 2012). More recently, these topic features have been replaced with sentence representations extracted from large language models, again apparently showing improvements (Ghinassi, 2021; Harrando and Troncy, 2021; Solbiati et al., 2021).

Recent research has also seen a surge of large annotated datasets for the task, usually exploiting the headers of Wikipedia articles to obtain large datasets without requiring human annotation. The first such dataset was proposed by Koshorek et al. (2018), but the most popular datasets in this category are the two Wikisection datasets proposed by Arnold et al. (2019), as their smaller sizes allow for faster experimentation.

With the availability of such larger, publicly available datasets, supervised methods became the preferred approach for the task. Koshorek et al. (2018) trained a hierarchical, Bidirectional Long-Short Term Memory (BiLSTM) neural network to segment paragraphs in a large Wikipedia corpus, showing good improvements over non-neural and unsupervised methods. Since then, most of the literature has focused on using hierarchical recurrent neural networks (Tsunoo et al., 2017; Lukasik et al., 2020a; Sehikh et al., 2018) or, more recently, hierarchical transformers (Lukasik et al., 2020b; Glavaš and Somasundaran, 2020). In recent works, BERT used as a sentence encoder has been included either to instill additional general knowledge to end-to-end systems (Xing et al., 2020) or to extract standalone features (Lo et al., 2021).

Transformer-based Large Language Models (LLMs) like BERT are extremely popular for many NLP tasks, often reaching state-of-the-art results. The same seemed to apply to text segmentation and recent literature has focused on the use of such models to perform text segmentation based only on local context, such as pairs of sentences, showing state-of-the-art results (Lee et al., 2023). In particular, the use of LLMs which were previously fine-tuned for sentence similarity together with additional fine-tuning of these models on the text segmentation task itself seemed to lead to best results, while the inclusion of additional context is, according to the authors, detrimental.

However, these last findings run counter to previous research, where the use of (limited) context was observed as generally beneficial (Lukasik et al., 2020a; Lo et al., 2021; Xing and Carenini, 2021; Xia et al., 2022) and the use of LLMs fine-tuned for sentence similarity did not lead to significant improvements (Solbiati et al., 2021). A more in depth exploration of state-of-the-art models shows further apparent contradictions. For example, the current second best model on Wikisection datasets shows significant improvements via the use of hierarchical transformers (Lo et al., 2021), while other sources have shown that, at least for certain datasets, BiLSTM networks can outperform transformers on this task (Lukasik et al., 2020a); this would be theoretically justified by the fact that recurrent neural networks such as BiLSTMs do give more importance to closer context, shown to be more relevant for the task (Xing and Carenini, 2021).

The current situation is therefore confusing, with different results suggesting quite different conclusions about the best choice of model architecture and settings. In this work, therefore, we focus on systematic comparison, and show that some of these discrepancies are explainable by the evaluation settings. When using a fixed evaluation setting, we can instead assess more convincingly what works best for the task and, as we show, this is indeed in line with our understanding of text segmentation as a task drawing from local coherence.

2.2 Evaluating Text Segmentation

Evaluating topic segmentation systems is itself an open problem. Classification metrics such as F1 score are not necessarily a good choice for topic segmentation: they consider a false positive boundary predicted just next to a true boundary, and one

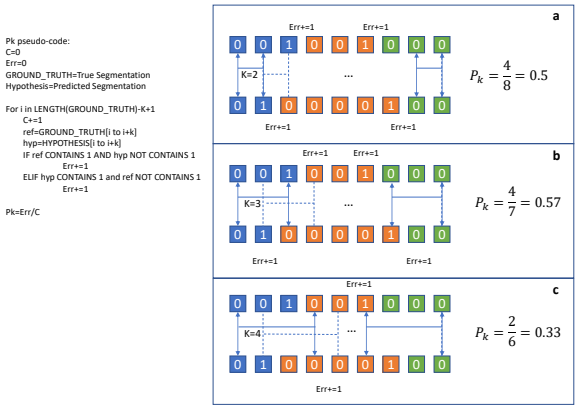

Figure 1: Pseudo-code and examples of P_k. Sub-figures a, b and c show the P_k result for the same ground truth and predicted boundaries but using $k = 2$, $k = 3$ and $k = 4$ respectively. It can be noticed how the P_k results vary greatly according to the parameter.

predicted ten sentences away, as equally bad misses. To overcome this problem Beeferman et al. (1999) proposed the P_k metric, which assesses how likely it is for two points a distance k apart (usually set to half the average true segment length) to be incorrectly separated by the hypothesized boundaries. However, P_k also has many reported problems (Pevzner and Hearst, 2002), failing to penalize incorrect separation by multiple boundaries more than single ones, and favouring false positives over true positives (Georgescul et al., 2006). Many other metrics have been proposed to overcome the limitations of P_k (Pevzner and Hearst, 2002; Scaiano and Inkpen, 2012; Fournier and Inkpen, 2012) but none of them has ever been widely adopted, and most literature still uses the P_k metric, notwithstanding its limitations.

Among the shortcomings of P_k is also the high sensitivity of the metric to its parameter k (see figure 1). This, as we will show, makes misunderstandings in the evaluation more likely, as the k parameter can be set in ways that are different from other evaluation settings, leading to differences in results that do not reflect actual meaningful differences in segmentation.

3 Methodology

3.1 Our Models

Here we describe our proposed models, which are chosen to represent the main state-of-the-art approaches in the literature and aim to find which architectural and feature factors determine a good text segmentation performance.

3.1.1 Architectures

We experiment with three different architectures (see their visual representation in figure 2):

BiLSTM: This architecture was first proposed for topic segmentation by Koshorek et al. (2018) and it has been widely used by following literature with various modifications (Xing and Carenini, 2021; Barrow et al., 2020; Badjatiya et al., 2018). In its original form, this model consists of n layers of Bidirectional Long-Short Memory (BiLSTM) recurrent neural network modelling the word-level features, a pooling layer to obtain sentence representations and n additional BiLSTM layers modelling the sentence-level features, followed by a linear layer and a Softmax activation yielding a series of probabilities $\hat{Y}$. In our case, we follow recent literature (Lukasik et al., 2020a; Xing and Carenini, 2021) and we substitute the word-level BiLSTM with embeddings extracted from sentence encoders during pre-processing. Schematically, if we define $BiLSTM$ as a series of n BiLSTM layers each having h hidden units, $W \in (R)^{h \times 1}$ as the final linear layer and $Softmax$ as the softmax activation function, our BiLSTM model predicts

$$\hat{Y} = Softmax(W^T(BiLSTM(E))) \quad (1)$$

where $E \coloneqq \{e_0, e_1, ..., e_n\}$ is the collection of all the sentence embeddings $e_i \in \mathbb{R}^d$ extracted from the given document's sentences.

At test time, we choose a threshold th by searching values between 0.05 to 0.95 with a step of 0.05 and choosing the one yielding best results on validation set. Threshold th is employed such that a topic boundary is placed after each sentence s_i for which $\hat{y}_i > th$.

Dot-BiLSTM: this architecture is similar to that of Sehikh et al. (2018) and Arnold et al. (2019), both having the intuition of separating the forward and the backward directions of the last BiLSTM layer in a network similar to the BiLSTM model described above, so as to directly compute a similarity score between the two, therefore forcing the model to exploit notions of semantic similarities more closely related to the downstream task. Having a stack of n $BiLSTM$ layers we obtain

$$H = BiLSTM(E) \quad (2)$$

Then, we separate H's forward direction $\overrightarrow{H}$ and backward direction $\overleftarrow{H}$, which are used to predict

$$\hat{Y} = 1 - Sigmoid(W_{for}^T \overrightarrow{H} \cdot W_{bac}^T \overleftarrow{H}) \quad (3)$$

with $Sigmoid$ being the sigmoid activation function, $\cdot$ being dot product and $W_{for} \in \mathbb{R}^h$ and $W_{for} \in \mathbb{R}^h$ both learnable parameters. The sigmoid-activated score is subtracted from 1, as we want the model to make sentences from two different topic segments further apart in the hidden space, thus closer to 0, while our objective labels define the identification of a topic boundary as 1.

We employ the same strategy as BiLSTM model to search the optimal threshold th.

Transformer: This architecture substitutes the BiLSTM to model sentences' context with a Transformer network (Vaswani et al., 2017). Similarly to above, we predict

$$\hat{Y} = Softmax(W^T(Transformer(E))) \quad (4)$$

where $Transformer$ represents the stack of n transformer layers substituting $BiLSTM$ from above and, in this case, $W \in \mathbb{R}^{d \times 2}$ reflecting the specific transformer architecture.

In this case, we set the threshold th to 0.5, as searching the threshold as described above consistently led to worse results.

3.1.2 Sentence Encoders

We experiment with two different sentence encoders further fine-tuned for topic segmentation.

RoBERTa last-mean (RoB): the popular RoBERTa architecture (Liu et al., 2019) consists of a 12-layer transformer encoder that was pre-trained on the masked language task in a more robust way than the original BERT architecture (Devlin et al., 2019), leading to considerable improvements on several benchmarks. Here we use the pre-trained model[2] and we obtain a single representation for each input sentence by averaging the last layer, shown to be an effective pooling strategy for sentence-level tasks (Huang et al., 2021).

All-MiniLM-L12-v2 (miniLM): this model is a version of the portable MiniLM language model, a comparatively smaller transformer encoder that is trained to mimic the last self-attention module of its larger counter-part, a process known as knowledge distillation (Wang et al., 2020). The version we use was further fine-tuned with a contrastive objective using cosine similarity between pairs of sentences that should be closer in space; it was used by Lee et al. (2023) as the backbone of their model, and here we compare it against larger, more popular transformer LLMs such as RoBERTa. Again, the sentence representation is obtained by averaging the last layer.

Both the above encoders were further fine-tuned on the topic segmentation task with this loss:

$$\mathcal{L} = ||label_{(i;i+1)} - \frac{e_i \cdot e_{i+1}}{||e_i||_2 \cdot ||e_{i+1}||_2}||_2 \quad (5)$$

where e_i and e_{i+1} are the sentence embeddings for sentences i and $i+1$, extracted by the sentence encoders. The corresponding $label_{(i;i+1)} = 1$ if they belong to the same segment, otherwise $label_{(i;i+1)} = -1$.

3.2 Other Baselines

We also report results from other baseline models for which existing implementations were available, so that the evaluation setting could be verified for each baseline. In our baseline comparisons we include $Transformer^2_{BERT}$[3] (Lo et al., 2021), $PairSeg_{MTL}$[4] (Lee et al., 2023), TextSeg[5] (Koshorek et al., 2018), BiLSTM-BERT[6] (Xing and Carenini, 2021), SECTOR[7] (Arnold et al., 2019) and TopicTiling[8] (Riedl and Biemann, 2012).

We also include NoPred, a baseline consisting in always predicting the majority class (i.e. no topic boundary): this simple baseline, in fact, can highlight how different k can determine very different results when using P_k, even when the predictions are just a constant value.

Other models have been variously proposed during the years and especially the ones proposed by Lukasik et al. (2020a) and Barrow et al. (2020) have been often used for baseline comparisons. As an official implementation for the two models is missing, however, we leave them out of our analysis, for the moment, leaving their inclusion in the revised ranking for future research.

3.3 Evaluation Setting

In evaluation, we used the mentioned P_k metric.

Most literature already settled on the use of half the average segment lengths when choosing k. Something that is not often specified is whether the average segment length should be computed based on the entire corpus or on single documents (therefore possibly leading to a different k for each test

[2] Model available at https://huggingface.co/roberta-base.

[3] github.com/kelvinlo-uni/Transformer-squared

[4] github.com/JHlee95/TxtSeg_MTL

[5] github.com/koomri/text-segmentation

[6] github.com/lxing532/improve_topic_seg

[7] github.com/sebastianarnold/SECTOR

[8] github.com/riedlma/topictiling

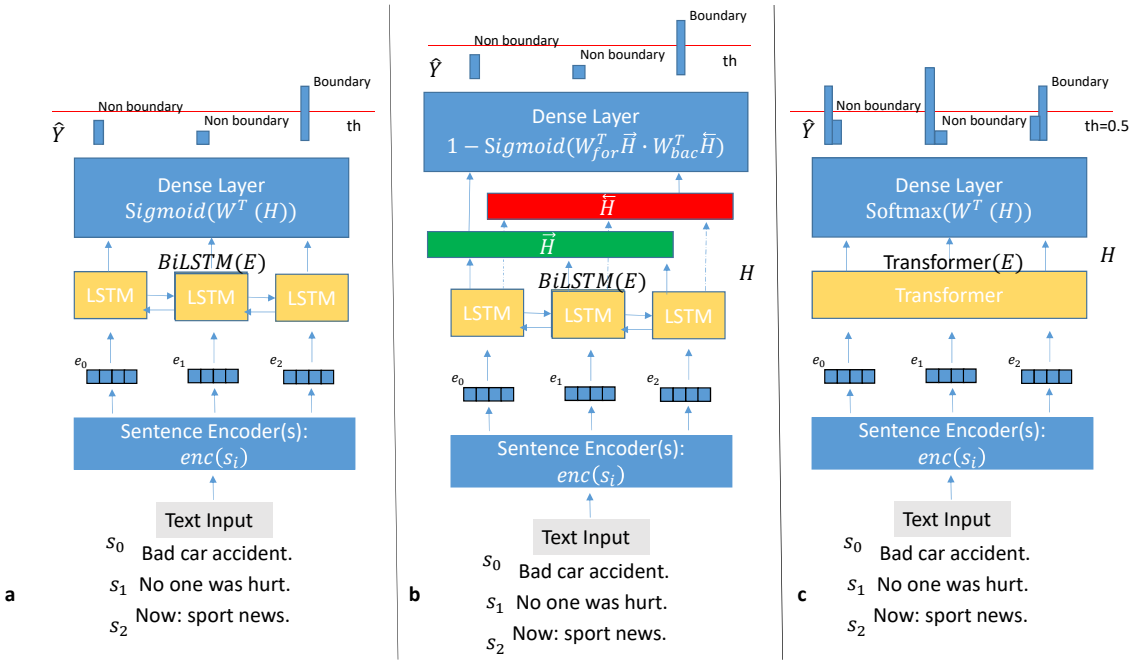

Figure 2: The three models we present: **a** BiLSTM; **b** Dot-BiLSTM; **c** Transformer.

document), but considering the existing implementations listed above it can be inferred that usually k is computed separately for each test document: this is also our default setting. Formally, given an input document doc having N segments, we compute:

$$k = \frac{\sum_{i=1}^{N} seglength_i}{2} \quad (6)$$

with $seglength_i$ being the length of the i_{th} segment in the document.

We also report results for different k to highlight how this can lead to divergent results.

3.4 Data

We use the Wikisection dataset proposed by Arnold et al. (2019). The dataset was obtained by scraping Wikipedia articles concerning specific macro-topics and using the existing headers to obtain ground truth labels for segmentation. The dataset is considerably smaller than the Wiki-757 dataset proposed by Koshorek et al. (2018) and it is therefore more popular in recent literature, as it allows for quicker experimentation. The dataset is divided in two languages, English and German, and two macro-topics for each language, cities and diseases. In our setting we follow recent literature and separate languages and macro-topics, therefore we obtain four separate datasets each having their predefined training, test and validation sets. Table 1 shows datasets statistics and general information.

Language	Macro-Topic	Abbrev.	Documents
English	Disease	en_disease	3900
English	City	en_city	19539
German	Disease	de_disease	2323
German	City	de_city	12537

Table 1: Wikisection datasets details: for more in-details information see the original paper (Arnold et al., 2019).

3.5 Experimental Setup

In our experiments we used the original parameters for all the baseline models, including the two state-of-the-art models described in section 3.1.

For BiLSTM and Dot-BiLSTM we followed the conventional setting of Koshorek et al. (2018) using 2 bidirectional LSTM layers, each direction having 128 hidden units. In training we minimised a binary cross entropy loss and we used a learning rate of 0.001 and Adam optimizer (Kingma and Ba, 2015). We applied dropout between input features and the first hidden layer, as well as between hidden layers, using for both probability values in the range

$\{0.2, 0.5\}$, where the optimal dropout probability was chosen based on validation results.

For our Transformer model, we followed the setting of Lo et al. (2021) using 5 transformer layers and a hidden dimension for the feedforward layer of 1024 hidden units. We have kept the dropout probability value to 0.2 as we observed no improvement in changing it and in training we minimised the cross entropy loss between the no-boundary and boundary class (where in our BiLSTM model we had a single output probability), using a learning rate of 0.0001 and Adam optimizer.

4 Results

4.1 Baseline Comparison with Standard P_k

Table 2 shows our results for the baselines and our models on the English Wikisection datasets.

A first look immediately shows that different k values affect not only absolute performance but the ranking of models; we discuss this in more detail in Section 4.2 below. However, even by looking just at the P_k^{def} columns (containing the results with the k we defined as standard), we can see that previous rankings do not hold in this consistent evaluation setting. Specifically, $Transformer^2_{BERT}$ does not seem to perform better than Bi-LSTM+BERT for en_city, and performs worse than all the other supervised baselines for en_disease; we discuss this in more detail later when analysing the influence of the Transformer architecture. The same holds for Pair_MTL, but in this case the model also underperforms with respect to SECTOR. Both these results contradict existing literature, suggesting that in fact the improvements that were noted in this case were due to a difference in evaluation setting, rather than in actual segmentation performance.[9]

Our BiLSTM-based models all perform better than most other baselines in both datasets, while our Transformer-based model shows extremely poor performance.

4.2 Sensitivity of P_k to k

The results using different k show conflicting results. By looking at the best performing models for P_k^{10}, it is evident that changing k does not influence the results in the same way for all models: if we set $k = 10$, $Transformer^2_{BERT}$ figures as the best model, while PairSeg_MTL under-performs; when changing to $k = 2$, the Transformer-based models

[9]By looking at the implementations listed above, Lo et al. (2021) set $k = 10$ and Lee et al. (2023) set $k = 2$.

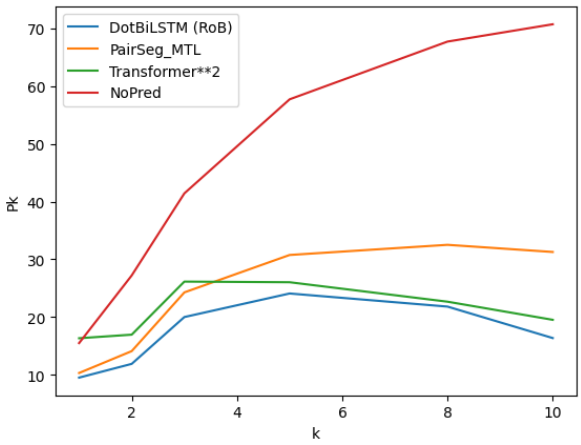

Figure 3: P_k results for different values of k and different models on en_disease test set.

are instead the worst performing ones. Even just never predicting a topic boundary produces very different P_k values according to which k we use, as shown in the first row of the table. The non-linear variation of results according to k is visually exemplified by figure 3.

4.3 Comparison of Different Architectures

Our results show that the Dot-BiLSTM architecture consistently outperforms other architectures; especially the Transformer-based model, which is consistently the worst.

The difference between Dot-BiLSTM and BiLSTM models is quite small, but this could be expected given the similarity of these two architectures. Still, Dot-BiLSTM always outperforms BiLSTM on both datasets, showing that the intuition of Sehikh et al. (2018) and of Arnold et al. (2019) was correct in the sense that forcing the model to directly modelling the similarity between adjacent units of text helps in the task of text segmentation. This was variously observed by including auxiliary losses during training (Xing and Carenini, 2021; Glavaš and Somasundaran, 2020), but here we observe how using this approach directly for segmentation works as well.

Given the consistent failure of the Transformer architecture, the relative success of $Transformer^2_{BERT}$ is more likely attributable to the use of pairwise embeddings from BERT, rather than some advantage of Transformer over BiLSTM on these datasets. If improvements using Transformer have previously been shown (Glavaš and Somasundaran, 2020; Lukasik et al., 2020a), such improvements were obtained on the much bigger

	en_city			en_disease		
Model	P_k^{def}	P_k^{10}	P_k^2	P_k^{def}	P_k^{10}	P_k^2
NoPred	32.93	32.39	22.13	40.53	70.71	27.21
TopicTiling	30.5	-	-	43.4	-	-
TextSeg	19.3	-	-	24.3	-	-
SECTOR	15.5	-	-	26.3	-	-
Bi-LSTM+BERT	9.3	-	-	21.1	-	-
$Transformer^2_{BERT}$	12.37	**8.2**	7	32.20	18.8	16.95
PairSeg_MTL	16.92	12.15	**4.9**	26.97	31.27	14.1
$BiLSTM_{RoB}$	8.97	5.33	5.32	22.29	**13.26**	12.51
$BiLSTM_{miniLM}$	8.9	8.49	5.19	22.75	16.8	13.03
$Transformer_{RoB}$	22.31	14.07	15.86	43.72	19.2	30.03
$Transformer_{miniLM}$	21.94	14.36	15.81	41.59	20.78	28.27
Dot-$BiLSTM_{RoB}$	**8.68**	8.62	5.12	**20.69**	16.36	**11.89**
Dot-$BiLSTM_{miniLM}$	8.77	8.39	5.17	22.49	15.8	12.7

Table 2: Results for all the presented models on en_city and en_disease datasets. For $Transformer^2_{BERT}$, $PairSeg_MTL$ and our models we present P_k results with the fixed k we established in section 3.3 (P_k^{def}), with $k = 10$ as used by Lo et al. (2021) (P_k^{10}) and with $k = 2$ as used by Lee et al. (2023)(P_k^2). In all cases, the lower the better. Best results for each dataset are highlighted in bold.

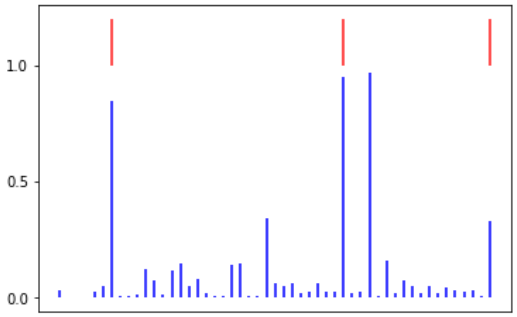

Figure 4: Probability of topic boundary output by Dot-$BiLSTM_{RoB}$ model for a test document. True boundaries are marked by the fixed-length vertical red lines at the top of the plot, while the output probabilities are represented by the variable-length blue lines.

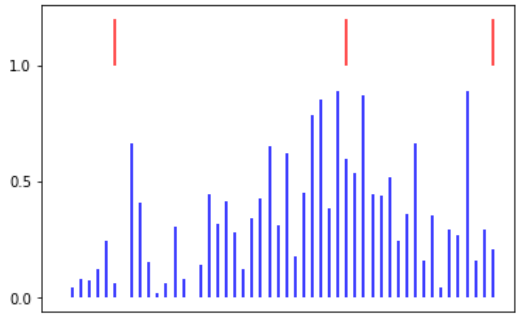

Figure 5: Probability of topic boundary output by $Transformer_{RoB}$ model for the same test document of figure 4. True boundaries are marked by the fixed-length vertical red lines at the top of the plot. The blue lines are the output probabilities.

Wiki-727 dataset. We hypothesise that the Wiki-section datasets are too small to effectively train a Transformer model, especially considering that the setting by Lo et al. (2021) is considerably deeper and bigger than the BiLSTM setting.

However, preliminary experiments with reducing the size of the Transformer model did not show any improvement either, and there could be some additional explanation to this. The role of local context in text segmentation is well known and has been exploited by much previous literature (Xia et al., 2022; Hearst, 1997; Choi et al., 2001). In this context, the advantage of the Transformer architecture in capturing long-distance dependencies (Vaswani et al., 2017) may not add any useful information for the task at hand, but instead potentially add noise, making the learning more difficult especially on small datasets.

This intuition is also confirmed by a qualitative comparison of the output from the best performing architecture shown in figure 4 against the output from the Transformer model using the same encoder (figure 5). In the first case, in fact, probabilities appear to be quite low everywhere but for the places in which the model is confident in outputting a boundary (which is mostly correct). The Transformer model clearly outputs noisier probabilities, with clusters of high probabilities rather than isolated peaks. Following the above reasoning, we hypothesise that this is an effect of the global self attention module introducing noise in the form of similarities between far away sentences, which are irrelevant for the task.

We further tested this hypothesis by re-training our Transformer models for all our settings, but restricting the context window of the self-attention

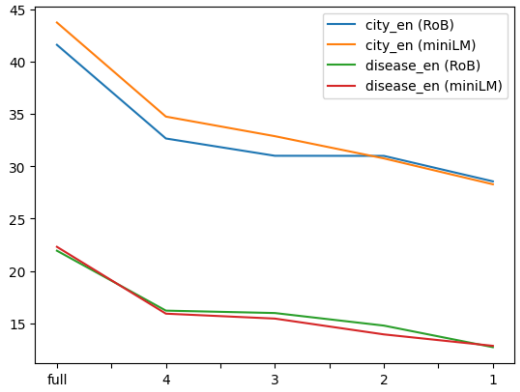

Figure 6: Effect of restricting the window of the self attention in our Transformer model. Y axis includes P_k values, while x axis includes the n parameter, representing the left and right context in the self attention module.

module to n sentences: at each time step, each sentence will have the information just from the n adjacent sentences. Figure 6 shows the results: for the Transformer architecture, restricting the available context always leads to better segmentation results, confirming our intuition. Still, the BiLSTM models outperform even the best performing Transformer setting, which might suggest that some characteristic of the BiLSTM architecture makes it more suitable for capturing the type of local context required for this task. Whether this is an effect of dataset size being too small for properly training a Transformer, or there is indeed some specific characteristic giving an edge to recurrent networks in this task, is an interesting question that we leave for future research.

4.4 Comparison of Different Encoders

Figure 7 shows the differences between encoders when using Dot-BiLSTM on the two English datasets. In the figure we also included the results for using the encoders without fine-tuning them, so as to isolate the effect of fine-tuning.

The differences between fine-tuned RoB and miniLM are small for en_city, while RoB performs more convincingly better on en_disease, even though the bigger difference could be an effect of bigger variation due to the dataset's smaller size.

In general, the choice of encoders does not seem to be extremely important when fine-tuning the encoders on the task. However, this changes when we do not fine-tune the encoders: in this case RoB outperforms miniLM by a larger margin on both datasets.

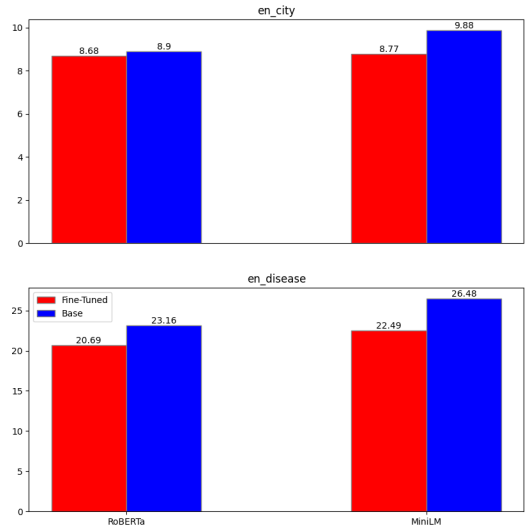

Figure 7: Comparison of results in terms of Pk_{def} for the DotBiLSTM model using RoB and miniLM encoders on en_city (top) and en_disease (bottom). We include results for both fine-tuned and base version of the encoders to evaluate the effect of fine-tuning.

The two versions of RoB (i.e. fine-tuned and base model) do not seem to present relevant differences for en_city, while fine-tuning seems to have a bigger effect on en_disease. When looking at miniLM, instead, the differences between fine-tuned and base models are much more noticeable for both datasets and this adds to the evidence from the comparison between RoB and miniLM in suggesting that RoB is probably a better encoder for text segmentation on these datasets.

Fine-tuning the encoders for text segmentation confirms itself as somewhat useful, but not at the level previously suggested by Lee et al. (2023).

4.5 Results on German Dataset

Model	de_city	de_disease
TopicTiling	41.3	45.4
TextSeg	27.5	35.7
SECTOR	16.2	27.5
Bi-LSTM+BERT	11.3	28
$Transformer^2_{DeBERT}$	13.30	27.89
PairSeg_MTL	41.08	33.40
$BiLSTM_{DeBERT}$	10.35	**22.61**
$Transformer_{DeBERT}$	26.11	37.46
Dot-$BiLSTM_{DeBERT}$	**10.27**	23.69

Table 3: Results using P_k^{def} for all the presented models on de_city and de_disease datasets. In all cases, the lower the better. Best results for each dataset are highlighted in bold.

Here we include the results obtained on de_city and de_disease. In carrying out these experiments

we used the German version of BERT, DEBERT,[10] so as to match the setting in Lo et al. (2021). For our models, we previously fine-tuned the base model on each training set as previously described.

The results on the German subsets of Wikisection (table 3) mostly confirm the observations from their English counterparts. Particularly, we also see here that the BiLSTM models are better than the Transformer-based ones, including the reported state-of-the-art, $Transformer^2_{DeBERT}$.

It is interesting to notice how in this case the PairSeg_MTL model seems to fail completely. This might be caused by more specific characteristics of these datasets rather than the difference in language, but it is an effect that could be investigated further in future. Finally, the simple BiLSTM model in this case outperforms the Dot-BiLSTM for de_disease; the results from the two models are always very similar given the similarity in the architecture and it is likely that this difference is not significant.

5 Conclusion

In this work, we have given a systematic, fair comparison of three state-of-the-art models for linear text segmentation with two fine-tuned sentence encoders as feature extractors for the task, so as to highlight what techniques proposed by recent literature work in a fair setting.

Consistent with existing literature, we have shown that the popular P_k metric is not very stable. Specifically, the influence of different k used in the metric is noticeable; with the result that if models are compared under different evaluation settings, the conclusions that could be drawn are very different and potentially misleading.

By keeping the evaluation setting fixed, however, we show that BiLSTM-based models actually outperform Transformers, at least on the current datasets, and that fine-tuning the sentence encoders does bring improvements but not necessarily as big as previously suggested. Restricting the context available to Transformer models leads to performance gains, as previously noticed by Lukasik et al. (2020a) and Lee et al. (2023); but Bi-LSTM-based systems always outperform even the best performing Transformer models, perhaps suggesting that some architectural element of LSTMs makes them more apt for the task at hand. This is indeed interesting evidence, which we aim to develop further in future work.

[10] https://huggingface.co/bert-base-german-cased

References

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. 2019. SECTOR: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184.

Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *Advances in Information Retrieval*, pages 180–193, Cham. Springer International Publishing.

Joe Barrow, Rajiv Jain, Vlad Morariu, Varun Manjunatha, Douglas Oard, and Philip Resnik. 2020. A joint model for document segmentation and segment labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 313–322, Online. Association for Computational Linguistics.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34.

Freddy Y Y Choi, Peter Wiemer-hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 102.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1.

Chris Fournier and Diana Inkpen. 2012. Segmentation similarity and agreement. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 152–161, Montréal, Canada. Association for Computational Linguistics.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, page 562–569, USA. Association for Computational Linguistics.

Maria Georgescul, Alexander Clark, and Susan Armstrong. 2006. Word distributions for thematic segmentation in a support vector machine approach. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X*.

Iacopo Ghinassi. 2021. Unsupervised text segmentation via deep sentence encoders: a first step towards a common framework for text-based segmentation, summarization and indexing of media content. In *2nd International Workshop on Data-driven Personalisation of Television (DataTV-2021) at the ACM*

International Conference on Interactive Media Experiences (IMX 2021) (DataTV-2021).

Goran Glavaš and Swapna Somasundaran. 2020. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*.

Ismail Harrando and Raphaël Troncy. 2021. And cut! exploring textual representations for media content segmentation and alignment. In *2nd International Workshop on Data-driven Personalisation of Television (DataTV-2021) at the ACM International Conference on Interactive Media Experiences (IMX 2021) (DataTV-2021)*.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL '94, page 9–16, USA. Association for Computational Linguistics.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 2.

Jeonghwan Lee, Jiyeong Han, Sunghoon Baek, and Min Song. 2023. Topic segmentation model focusing on local context. *ArXiv*, abs/2301.01935.

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*.

Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray L. Buntine. 2021. Transformer over pre-trained transformer for neural text segmentation with enhanced topic coherence. In *EMNLP*.

Michael Lukasik, Boris Dadachev, Gonçalo Simões, and Kishore Papineni. 2020a. Text segmentation by cross segment attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4707–4716.

Michal Lukasik, Boris Dadachev, Gonçalo Simões, and Kishore Papineni. 2020b. Text segmentation by cross segment attention. *arXiv*.

Hemant Misra, Frank Hopfgartner, Anuj Goyal, P. Punitha, and Joemon M. Jose. 2010. TV news story segmentation based on semantic coherence and content similarity. In *Advances in Multimedia Modeling*, pages 347–357, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lev Pevzner and Marti A. Hearst. 2002. A Critique and Improvement of an Evaluation Metric for Text Segmentation. *Computational Linguistics*, 28(1):19–36.

Matthew Purver, Konrad P. Körding, Thomas L. Griffiths, and Joshua B. Tenenbaum. 2006. Unsupervised topic modelling for multi-party spoken discourse. In *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, volume 1.

Jeffrey C. Reynar. 1999. Statistical models for topic segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, page 357–364, USA. Association for Computational Linguistics.

Martin Riedl and Chris Biemann. 2012. Text segmentation with topic models. *Journal for Language Technology and Computational Linguistics*, 27.

Martin Scaiano and Diana Inkpen. 2012. Getting more from segmentation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 362–366, Montréal, Canada. Association for Computational Linguistics.

Imran Sehikh, Dominique Fohr, and Irina Illina. 2018. Topic segmentation in ASR transcripts using bidirectional RNNs for change detection. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017 - Proceedings*, volume 2018-January.

Alessandro Solbiati, Kevin Hefferman, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. *arXiv*.

Emiru Tsunoo, Peter Bell, and Steve Renals. 2017. Hierarchical recurrent neural network for story segmentation. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2017-August.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MINILM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Jinxiong Xia, Cao Liu, Jiansong Chen, Yuchen Li, Fan Yang, Xunliang Cai, Guanglu Wan, and Houfeng Wang. 2022. Dialogue topic segmentation via parallel extraction network with neighbor smoothing. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2126–2131, New York, NY, USA. Association for Computing Machinery.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 167–177, Singapore and Online. Association for Computational Linguistics.

Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. 2020. Improving context modeling in neural topic segmentation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 626–636, Suzhou, China. Association for Computational Linguistics.