Improving Text Segmentation with Two-Level Transformer and Auxiliary Coherence Modeling

Anonymous EMNLP-IJCNLP submission

Abstract

Breaking down the structure of long texts into semantically coherent segments makes the texts more readable and supports downstream applications like summarization and retrieval. Starting from an apparent link between text coherence and segmentation, we introduce a novel supervised model for text segmentation with simple but explicit coherence modeling. Our model – a neural architecture consisting of two hiearchically connected Transformer networks – is a multi-task learning model that couples the sentence-level segmentation objective with the coherence objective differentiating correct sentence sequences from corrupt ones. The proposed model, named Coherence-Aware Text Segmentation (CATS), yields state-of-the-art segmentation performance on a collection of benchmark datasets. Furthermore, by coupling CATS with cross-lingual word embeddings, we demonstrate its effectiveness in zero-shot language transfer: it successfully segment texts in languages unseen in training.

1 Introduction

Natural language texts are, more often than not, a result of a deliberate cognitive effort of an author and as such consist of semantically coherent segments. Text segmentation deals with automatically breaking down the structure of text into such topically contiguous segments, i.e., it aims to identify the points of topic shift (Hearst, 1994; Choi, 2000; Brants et al., 2002; Riedl and Biemann, 2012; Du et al., 2013; Glavaš et al., 2016; Koshorek et al., 2018). Reliable segmentation results with texts that are more readable for humans, but also facilitates downstream tasks like automated text summarization (Angheluta et al., 2002; Bokaei et al., 2016), passage retrieval (Huang et al., 2003; Prince and Labadié, 2007; Shtekh et al., 2018), or dialog modeling (Griol and Molina, 2015; Manuvinakurike et al., 2016; Zhao and Kawahara, 2017).

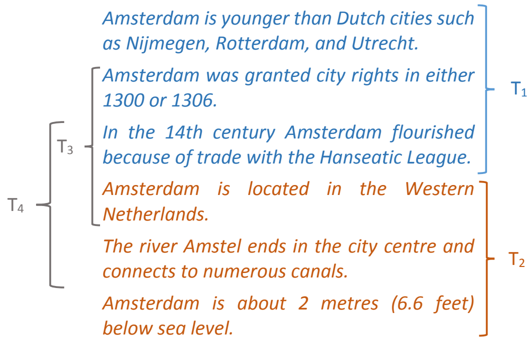

Figure 1: Snippet illustrating the relation (i.e., dependency) between text coherence and segmentation.

Text coherence is inherently tied to text segmentation – intuitively, the text within a segment is expected to be more coherent than the text spanning different segments. Consider, e.g., the text in Figure 1, with two topical segments. Snippets T_1 and T_2 are more coherent than T_3 and T_4: all T_1 sentences relate to *Amsterdam's history*, and all T_2 sentences to *Amsterdam's geography*; in contrast, T_3 and T_4 contain sentences from both topics. T_1 and T_2 being more coherent than T_3 and T_4 signals that the fourth sentence starts a new segment.

Given this duality between text segmentation and coherence, it is surprising that the methods for text segmentation capture coherence only implicitly. Unsupervised segmentation models rely either on probabilistic topic modeling (Brants et al., 2002; Riedl and Biemann, 2012; Du et al., 2013) or semantic similarity between sentences (Glavaš et al., 2016), both of which only indirectly relate to text coherence. Similarly, a recently proposed state-of-the-art supervised neural segmentation model (Koshorek et al., 2018) directly learns to predict binary sentence-level segmentation decisions and has no explicit mechanism for modeling coherence.

In this work, in contrast, we propose a supervised neural model for text segmentation that explicitly takes coherence into account: we augment the segmentation prediction objective with an auxiliary coherence modeling objective. Our proposed model, dubbed Coherence-Aware Text Segmentation (CATS), encodes a sentence sequence using two hierarchically connected Transformer networks (Vaswani et al., 2017; Devlin et al., 2018). Similar to (Koshorek et al., 2018), CATS' main learning objective is a binary sentence-level segmentation prediction. However, CATS augments the segmentation objective with an auxiliary coherence-based objective which pushes the model to predict higher coherence for original text snippets than for corrupt (i.e., fake) sentence sequences. We empirically show (1) that even without the auxiliary coherence objective, the Two-Level Transformer model for Text Segmentation (TLT-TS) yields state-of-the-art performance across multiple benchmarks, (2) that the CATS model, with the auxiliary coherence modeling, further significantly improves the segmentation, and (3) that both TLT-TS and CATS are robust in domain transfer. Furthermore, we demonstrate models' effectiveness in zero-shot language transfer. Coupled with a cross-lingual word embedding space,[1] our models trained on English Wikipedia successfully segments texts from unseen languages, outperforming the best-performing unsupervised segmentation model (Glavaš et al., 2016) by a wide margin.

2 Coherence-Aware Two-Level Transformer for Text Segmentation

Figure 2 illustrates the high-level architecture of the CATS model. A snippet of text – a sequence of sentences of fixed length – is an input to the model. Token encodings are a concatenation of a pretrained word embedding and a positional embedding. Sentence are first encoded from their tokens with a token-level Transformer (Vaswani et al., 2017). Next, we feed the sequence of obtained sentence representations to the second, sentence-level Transformer. Transformed (i.e., contextualized) sentence representations are next fed to the feed-forward segmentation classifier, which makes a binary segmentation prediction for each sentence. We additionally feed the encoding of the whole snippet (i.e., the sentence sequence) to the coherence regressor (a feed-forward net), which predicts

[1] See (Ruder et al., 2018; Glavaš et al., 2019) for a comprehensive overview of methods for inducing cross-lingual word embeddings.

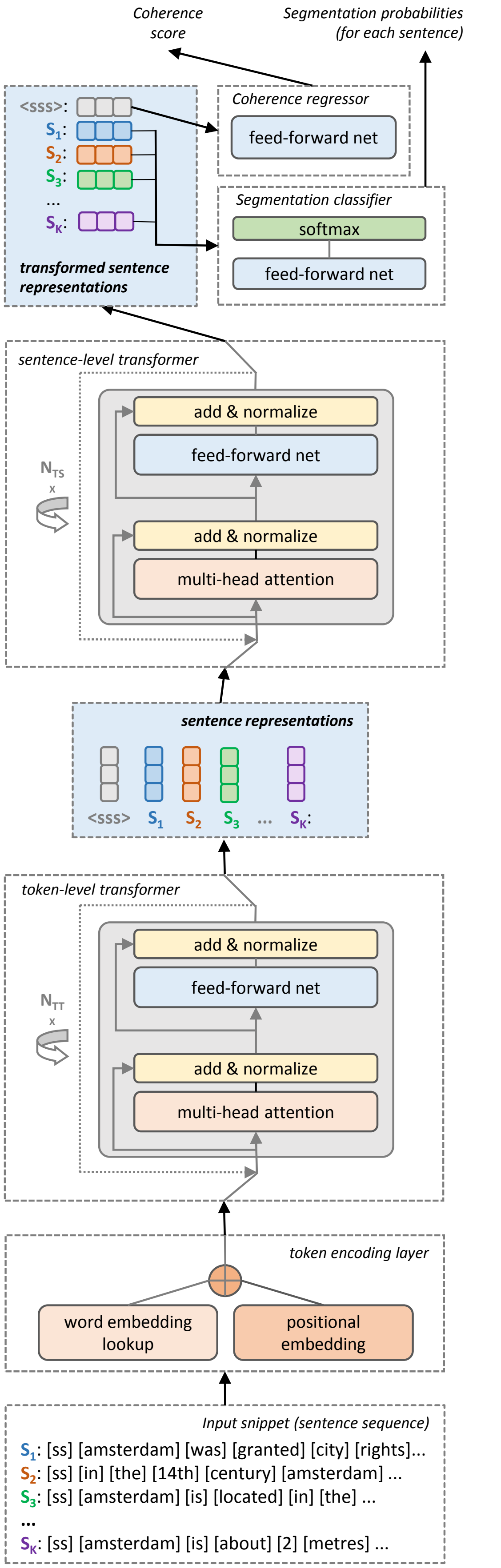

Figure 2: High-level depiction of the Coherence-Aware Text Segmentation (CATS) model.

a coherence score. In what follows, we describe each component in more detail.

2.1 Transformer-Based Segmentation

The segmentation decision for a sentence clearly does not depend only on its content but also on its context, i.e., information from neighboring sentences. In this work, we employ the encoding stack of the attention-based Transformer architecture (Vaswani et al., 2017) to contextualize both token representations in a sentence and, more importantly, sentence representations within the snippet. We choose Transfomer encoders because (1) they have recently been reported to outperform recurrent encoders on a range of NLP tasks (Devlin et al., 2018; Radford et al., 2018; Shaw et al., 2018) and (2) they are faster to train than recurrent nets.

Sentence Encoding. Let $\mathbb{S} = \{S_1, S_2, \dots, S_K\}$ denote a single training instance – a snippet consisting of K sentences and let each sentence $S_i = \{t_1^i, t_2^i, \dots, t_T^i\}$ be a fixed-size sequence of T tokens.[2] Following (Devlin et al., 2018), we prepend each sentence S_i with a special sentence start token t_0^i = [ss], aiming to use the transformed representation of that token as the sentence encoding.[3] We encode each token t_j^i ($i \in \{1, \dots, K\}$, $j \in \{0, 1, \dots, T\}$) with a vector $\mathbf{t}_j^i$ which is the concatenation of a d_e-dimensional word embedding and a d_p-dimensional embedding of the position j. We use pretrained word embeddings and fix them in training; we learn positional embeddings as model's parameters. Let $\mathit{Transform}_T$ denote the encoder stack of the Transformer model (Vaswani et al., 2017), consisting of N_{TT} layers, each coupling a multi-head attention net with a feed-forward net.[4] We then apply $\mathit{Transform}_T$ to the token sequence of each snippet sentence:

$$\{\mathbf{tt}_j^i\}_{j=0}^T = \mathit{Transform}_T\left(\{\mathbf{t}_j^i\}_{j=0}^T\right); \quad (1)$$

The sentence encoding is then the transformed vector of the sentence start token [ss]: $\mathbf{s}_i = \mathbf{tt}_0^i$.

Sentence Contextualization. Sentence encodings $\{\mathbf{s}_i\}_{i=1}^K$ produced with $\mathit{Transform}_T$ only capture the content of the sentence itself, but not its context. We thus employ a second, sentence-level Transformer $\mathit{Transform}_S$ (with N_{TS} layers) to produce context-informed sentence representations. We prepend each sequence of non-contextualized sentence embeddings $\{\mathbf{s}_i\}_{i=1}^K$ with a fixed embedding $\mathbf{s}_0$, denoting the snippet start token <sss>, in order capture the encoding of the whole snippet (i.e., sequence of K sentences) as the transformed embedding of the <sss> token:

$$\{\mathbf{ss}_i\}_{i=0}^K = \mathit{Transform}_S\left(\{\mathbf{s}_i\}_{i=0}^K\right); \quad (2)$$

with the transformed vector $\mathbf{ss}_0$ being the encoding of the whole snippet $\mathbb{S}$.

Segmentation Classification. Finally, we feed the contextualized sentence vectors $\mathbf{ss}_i$ to the segmentation classifier, a simple single-layer feed-forward net coupled with a softmax function:

$$\hat{\mathbf{y}}_i = \mathit{softmax}(\mathbf{ss}_i \mathbf{W}_{seg} + \mathbf{b}_{seg}); \quad (3)$$

with $\mathbf{W}_{seg} \in \mathbb{R}^{(d_e+d_p)\times 2}$ and $\mathbf{b}_{seg} \in \mathbb{R}^2$ as classifier's parameters. Let $\mathbf{y}_i \in \{[0,1],[1,0]\}$ be the true segmentation label of the i-th sentence. The segmentation loss J_{seg} is then the simple negative log-likelihood over all sentences of all N snippets in the training batch:

$$J_{seg} = -\sum_{n=1}^{N}\sum_{i=1}^{K} \ln \hat{\mathbf{y}}_i^n \cdot \mathbf{y}_i^n. \quad (4)$$

2.2 Auxiliary Coherence Modeling

Given the obvious dependency between segmentation and coherence, we pair the segmentation task with an auxiliary task of predicting snippet coherence. To this effect, we couple each true snippet $\mathbb{S}$ from the original text with a corrupt (i.e., incoherent) snippet $\bar{\mathbb{S}}$, created by (1) randomly shuffling the order of sentences in $\mathbb{S}$ and (2) randomly replacing sentences with other document sentences (see §2.4 for more details on the corruption procedure).

Let $(\mathbb{S}, \bar{\mathbb{S}})$ be a pair of a true snippet and its corrupt counterpart, and $(\mathbf{ss}_0, \overline{\mathbf{ss}_0})$ their respective encodings, obtained with the Two-Level Transformer. The encodings of the correct snippet ($\mathbf{ss}_0$) and the scrambled snippet ($\overline{\mathbf{ss}_0}$) are then presented to the coherence regressor, which independently generates a coherence score for each of them. The scalar output of the coherence regressor is:

$$\hat{\mathbf{y}}_{\mathbb{S}} = \mathbf{ss}_0 \mathbf{w}_c + b_c; \quad \hat{\mathbf{y}}_{\bar{\mathbb{S}}} = \overline{\mathbf{ss}_0} \mathbf{w}_c + b_c; \quad (5)$$

[2] We trim sentences longer than T tokens and pad sentences shorter than T tokens.

[3] This strategy eliminates the need for an additional self-attention layer that would aggregate transformed token vectors into a sentence encoding.

[4] For more details on the encoding stack of the Transformer model, see the original publication (Vaswani et al., 2017).

with $\mathbf{w}_c \in \mathbb{R}^{d_e+d_p}$ and $b_c \in \mathbb{R}$ as regressor's parameters. We then jointly softmax-normalize the coherence scores for $\mathbb{S}$ and $\bar{\mathbb{S}}$:

$$[coh(\mathbb{S}), coh(\bar{\mathbb{S}})] = softmax\left([\hat{\mathbf{y}}_{\mathbb{S}}, \hat{\mathbf{y}}_{\bar{\mathbb{S}}}]\right). \quad (6)$$

We want to force the model to produce higher coherence score for the correct snippet $\mathbb{S}$ than for its corrupt counterpart $\bar{\mathbb{S}}$. We thus define the following contrastive margin-based coherence objective:

$$J_{coh} = \max\left(0, \delta_{coh} - (coh(\mathbb{S}) - coh(\hat{\mathbb{S}}))\right) \quad (7)$$

where δ_{coh} is the margin by which we would like $coh(\mathbb{S})$ to be larger than $coh(\bar{\mathbb{S}})$.

2.3 Creating Training Instances

Our presumed training corpus contains documents that are generally longer than the snippet size K and annotated for segmentation at the sentence level. We create training instances by sliding a sentence window of size K over documents' sentences with a stride of $K/2$.

For the sake of auxiliary coherence modeling, for each original snippet S, we create its corrupt counterpart $\bar{\mathbb{S}}$ with the following corruption procedure: (1) we first randomly shuffle the order of sentences in $\mathbb{S}$; (2) for p_1 percent of snippets (random selection) we additionally replace sentences of the shuffled snippet (with the probability p_2) with randomly chosen sentences from other, non-overlapping snippets.

2.4 Inference

At inference time, given a long document, we need to make a binary segmentation decision for each sentence. Our model, however, does not take individual sentences as input, but rather sequence of K sentences (i.e., snippets) and makes in-context segmentation prediction for each sentence. Since we can create multiple different sequences of K consecutive sentences that contain some sentence S,[5] our model can obtain multiple segmentation predictions for the same sentence. As we do not know apriori which of the snippets containing a sentence S is the most reliable with respect to the segmentation prediction for S, we consider *all* possible snippets containing S. In other words, at inference time, unlike in training, we create snippets by sliding the window of K sentences over the document with the stride of 1. Let $\mathcal{S} = \{\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_K\}$ be the set of (at most) K different snippets containing a sentence S. We then average the segmentation probabilities predicted for the sentence S over all snippets in $\mathcal{S}$:[6]

$$P_{seg}(S) = \frac{1}{K} \sum_{\mathbb{S}_k \in \mathcal{S}} \hat{\mathbf{y}}_S\left(\mathbb{S}_k\right)[0] \quad (8)$$

We then predict that S starts a new segment if $P_{seg}(S) > \tau$, where τ is the confidence threshold, tuned as a hyperparameter of the model.

2.5 Cross-Lingual Zero-Shot Transfer

Models that do not require any language-specific features other than pretrained word embeddings as input can (at least conceptually) be easily transferred to another language by means of a cross-lingual word embedding space (Ruder et al., 2018; Glavaš et al., 2019). Let $\mathbf{X}_{L1}$ be the monolingual embedding space of the source language (most often English), which we use in training and let $\mathbf{X}_{L2}$ be the independently trained embedding space of the target language to which we want to transfer the segmentation model. To transfer the model, we need to project target-language vectors from $\mathbf{X}_{L2}$ to the source-language space $\mathbf{X}_{L1}$. There is a plethora of recently proposed methods for inducing projection-based cross-lingual embeddings (Faruqui and Dyer, 2014; Smith et al., 2017; Conneau et al., 2018; Artetxe et al., 2018; Hoshen and Wolf, 2018; Alvarez-Melis and Jaakkola, 2018, *inter alia*). We opt for the supervised alignment model based on solving the Procrustes problem (Smith et al., 2017), due to its simplicity and competitive performance in zero-shot language transfer of NLP models (Glavaš et al., 2019). Given a limited-size word translation training dictionary D, we obtain the linear projection matrix $\mathbf{W}_{L2\to L1}$ between $\mathbf{X}_{L2}$ and $\mathbf{X}_{L1}$ as follows:

$$\begin{aligned} \mathbf{W}_{L2\to L1} &= \mathbf{U}\mathbf{V}^\top, \text{ where} \\ \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top &= SVD(\mathbf{X}_S{\mathbf{X}_T}^\top); \end{aligned} \quad (9)$$

with $\mathbf{X}_S \subset \mathbf{X}_{L1}$ and $\mathbf{X}_T \subset \mathbf{X}_{L2}$ as subsets of monolingual spaces that align vectors from training translations pairs from D. Once we obtain $\mathbf{W}_{L2\to L1}$, the language transfer of the segmentation model is straightforward: we input the embeddings of L2 words from the projected space $\mathbf{X}'_{L2} = \mathbf{X}_{L2}\mathbf{W}_{L2\to L1}$.

[5] Sliding the sentence window with the stride of 1, the m-th sentence will, in the general case, be found in K different snippets: $[m-K+1 : m]$, $[m-K+2 : m+1]$, …, $[m : m+K-1]$.

[6] The first element (i.e., index [0]) of the predicted vector $\hat{\mathbf{y}}$ denotes the (positive) segmentation probability.

3 Experimental Setup

We first describe datasets used for training and evaluation and then provide the details on the comparative evaluation setup and model optimization.

3.1 Data

WIKI-727K Corpus. Koshorek et al. (2018) leveraged the manual structuring of Wikipedia pages into sections to automatically create a large segmentation-annotated corpus. WIKI-727K consists of 727,746 documents created from English (EN) Wikipedia pages, divided into training (80%), development (10%), and test portions (10%). We train, optimize, and evaluate our models on respective portions of the WIKI-727K dataset.

Standard Test Corpora. Koshorek et al. (2018) additionally created a small evaluation set WIKI-50 to allow for comparative evaluation against unsupervised segmentation models, e.g., the GRAPHSEG model of Glavaš et al. (2016), for which evaluation on large datasets is prohibitively slow. For years, the synthetic dataset of Choi (2000) was used as a standard becnhmark for text segmentation models. CHOI dataset contains 920 documents, each of which is a concatenation of 10 paragraphs randomly sampled from the Brown corpus. CHOI dataset is divided into subsets containing only documents with specific variability of segment lengths (e.g., segments with 3-5 or with 9-11 sentences).[7] Finally, we evaluate the performance of our models on two small datasets, CITIES and ELEMENTS, created by Chen et al. (2009) from Wikipedia pages dedicated to the cities of the world and chemical elements, respectively.

Other Languages. In order to test the performance of our Transformer-based models in zero-shot language transfer setup, we prepared small training datasets in other languages. Analogous to the WIKI-50 dataset created by Koshorek et al. (2018) from English (EN) Wikipedia, we created WIKI-50-CS, WIKI-50-FI, and WIKI-50-TR datasets consisting of 50 randomly selected pages from Czech (CS), Finnish (FI), and Turkish (TR) Wikipedia, respectively.[8]

3.2 Comparative Evaluation

Evaluation Metric. Following previous work (Riedl and Biemann, 2012; Glavaš et al., 2016; Koshorek et al., 2018), we also adopt the standard text segmentation measure P_k (Beeferman et al., 1999) as our evaluation metric. P_k score is the probability that a model makes a wrong prediction as to whether the first and last sentence of a randomly sampled snippet of k sentences belong to the same segment (i.e., the probability of the model predicting the same segment for the sentences from different segment or different segments for the sentences from the same segment). Following (Glavaš et al., 2016; Koshorek et al., 2018), we set k to the half of the average segment size according to the gold-standard segmentation of the dataset.

Baseline Models. We compare CATS against the state-of-the-art neural segmentation model of (Koshorek et al., 2018) and against GRAPHSEG (Glavaš et al., 2016), the state-of-the-art unsupervised text segmentation model. Additionally, as a sanity check, we evaluate the RANDOM baseline – it assigns a positive segmentation label to a sentence with the probability that corresponds to the ratio of the total number of segments (according to the gold segmentation) and total number of sentences in the dataset.

3.3 Model Configuration

Model Variants. We evaluate two variants of our two-level transformer text segmentation model: with and without the auxiliary coherence modeling. The first model, TLT-TS, minimizes only the segmentation objective J_{seg}. CATS, our second model, is a multi-task learning model that alternately minimizes the segmentation objective J_{seg} and the coherence objective J_{coh}. We adopt a balanced alternate training regime for CATS in which a single parameter update based on the minimization of J_{seg} is followed by a single parameter update based on the optimization of J_{coh}.

Word Embeddings. In all our experiments we use 300-dimensional monolingual FASTTEXT word embeddings pretrained on the Common Crawl corpora of respective languages: EN, CS, FI, and TR.[9] We induce a cross-lingual word embedding space, needed for the zero-shot language transfer experiments, by projecting CS, FI, and TR monolingual

[7] Following Koshorek et al. (2018), we evaluate our models on the whole CHOI corpus and not on specific subsets.

[8] For our language transfer experiments we selected target languages from different families and linguistic typologies w.r.t English as our source language: Czech is, like English, an Indo-European language (but as a Slavic language it is, unlike English, fusional by type); Finnish is an Uralic language (fusionally-agglutinative by type); whereas Turkish is a Turkic language (agglutinative by type).

[9] https://tinyurl.com/y6j4gh9a

embedding spaces to the EN embedding space. Following (Smith et al., 2017; Glavaš et al., 2019), we create training dictionaries D for learning projection matrices by machine translating 5,000 most frequent EN words to CS, FI, and TR.

Model Optimization. We optimize all hyperparameters, including the data preparation parameters like the snippet size K, via cross-validation on the development portion of the Wiki-727K dataset. We found the following configuration to lead to robust[10] performance for both TLT-TS and CATS: (1) training instance preparation: snippet size of $K = 16$ sentences with $T = 50$ tokens; scrambling probabilities $p_1 = p_2 = 0.5$; (2) configuration of Transformers: $N_{TT} = N_{TS} = 6$ layers and with 4 attention heads per layer in both transformers;[11] (3) other model hyperparameters: positional embedding size of $d_p = 10$; coherence objective contrastive margin of $\delta_{coh} = 1$. We found different optimal inference thresholds: $\tau = 0.5$ for the segmentation-only TLT-TS model and $\tau = 0.3$ for the coherence-aware CATS model. We trained both TLT-TS and CATS in batches of $N = 32$ snippets (each with $K = 16$ sentences), using the Adam optimization algorithm (Kingma and Ba, 2014) with the initial learning rate set to 10^{-4}.

4 Results and Discussion

We first present and discuss the results that our models, TLT-TS and CATS, yield on the previously introduced EN evaluation datasets. We then report and analyze models' performance in the cross-lingual zero-shot transfer experiments.

4.1 Base Evaluation

Table 1 shows models' performance on five EN evaluation datasets. Both our Transformer-based models – TLT-TS and CATS – outperform the competing supervised model of (Koshorek et al., 2018), a hierarchical encoder based on recurrent components, across the board. The improved performance that TLT-TS has with respect to the model of Koshorek et al. (2018) is consistent with improvements that Transformer-based architectures yield in comparison with models based on recurrent components in other NLP tasks (Vaswani et al., 2017; Devlin et al., 2018). The gap in performance is particularly wide (>20 P_k points) for the ELEMENTS dataset. Evaluation on the ELEMENTS test set is, arguably, closest to a true domain-transfer setting:[12] while the train portion of the WIKI-727K set contains pages similar in type to those found in WIKI-50 and CITIES test sets, it does not contain any Wikipedia pages about chemical elements (all such pages are in the ELEMENTS test set). This would suggest that TLT-TS and CATS offers more robustness in domain transfer than the recurrent model of Koshorek et al. (2018).

CATS significantly[13] and consistently outperforms TLT-TS. This empirically confirms the usefulness of explicit coherence modeling for text segmentation. Moreover, Koshorek et al. (2018) report human performance on the WIKI-50 dataset of 14.97, a mere one P_k point above the performance of our coherence-aware CATS model.

The unsupervised GRAPHSEG model of Glavaš et al. (2016) seems to outperform all supervised models on the synthetic CHOI dataset. We believe that this is primarily because (1) by being synthetic, the CHOI dataset can be accurately segmented based on simple lexical overlaps and word embedding similarities (and GRAPHSEG relies on similarities between averaged word embeddings) and because (2) by being trained on a much more challenging real-world WIKI-727K dataset – on which lexical overlap is insufficient for accurate segmentation – supervised models learn to segment based on deeper natural language understanding (and learn not to encode lexical overlap as reliable segmentation signal). Additionally, GRAPHSEG is evaluated separately on each subset of the CHOI dataset, for each of which it is provided the (gold) minimal segment size, which further facilitates and improves its predicted segmentations.

4.2 Zero-Shot Cross-Lingual Transfer

In Table 2 we show the results of our zero-shot cross-lingual transfer experiments. In this setting,

[10] Given the large number of hyperparameters and the size of the training dataset (i.e., the training time for a single model instance), we only performed a search over a limited-size grid of hyperparameter configurations. It is thus likely that a better-performing configuration than the one reported could be found with a more extensive grid search.

[11] We do not tune other transformer hyperparameters, but rather adopt the recommended values from (Vaswani et al., 2017): filter size of 1024 and dropout probabilities of 0.1 for both attention layers and feed-forward ReLu layers.

[12] The CHOI dataset – albeit from a different domain – is synthetic, which impedes direct performance comparisons with other evaluation datasets.

[13] According to the non-parametric random shuffling test (Yeh, 2000): $p < 0.01$ for WIKI-727K, CHOI and CITIES; $p < 0.05$ for WIKI-50 and ELEMENTS.

Model	Model Type	WIKI-727K	WIKI-50	CHOI	CITIES	ELEMENTS
RANDOM	unsupervised	53.09	52.65	49.43	47.14	50.08
Glavaš et al. (2016)	unsupervised	–	63.56	**5.6–7.2***	39.95	49.12
Koshorek et al. (2018)	supervised	22.13	18.24	26.26	19.68	41.63
TLT-TS	supervised	19.41	17.47	23.26	19.21	20.33
CATS	supervised	**15.95**	**16.53**	18.50	**16.85**	**18.41**

Table 1: Performance of text segmentation models on five English evaluation datasets. Baseline results taken from (Glavaš et al., 2016) and (Koshorek et al., 2018). GRAPHSEG model (Glavaš et al., 2016) was evaluated independently on different subcorpora of the CHOI dataset (indicated with an asterisk).

Model	CS	FI	TR
RANDOM	52.92	52.02	45.04
Glavaš et al. (2016)	49.47	49.28	39.21
TLT-TS	24.27	25.99	25.89
CATS	**22.32**	**22.87**	**24.20**

Table 2: Performance of text segmentation models in zero-shot language transfer setting on the WIKI-50-X ($X \in \{CS, FI, TR\}$) datasets.

we use our Transformer-based models, trained on the English WIKI-727K dataset, to segment texts from the WIKI-50-X ($X \in \{CS, FI, TR\}$) datasets in other languages. As a baseline, we additionally evaluate the unsupervised GRAPHSEG (Glavaš et al., 2016) model, since it is language-agnostic and requires only pretrained word embeddings of the test language as input.

Both our Transformer-based models, TLT-TS and CATS, outperform the unsupervised GRAPHSEG model (which seems to be only marginally better than the random baseline) by a wide margin. The coherence-aware CATS model is again significantly better ($p < 0.01$ for FI and $p < 0.05$ for CS and TR) than the TLT-TS model which was trained to optimize only the segmentation objective. While the results on the WIKI-50-{CS, FI, TR} datasets are not directly comparable to the results reported on the EN WIKI-50 (see Table 1) because the datasets in different languages do not contain mutually comparable Wikipedia pages, results in Table 2 still suggest that the drop in performance due to the cross-lingual transfer is not big. This is quite encouraging as it suggests that it is possible to, via the zero-shot language transfer, rather reliably segment texts from under-resourced languages lacking sufficiently large gold-segmented data needed to directly train language-specific segmentation models (that is, robust neural segmentation models in particular).

5 Related Work

In this work we address the task of text segmentation – we thus provide a detailed account of existing segmentation models. Because our CATS model has an auxiliary coherence-based objective, we additionally provide a brief overview of research on modeling text coherence.

5.1 Text Segmentation

Text segmentation tasks come in two main flavors: (1) linear (i.e., sequential) text segmentation and (2) hierarchical segmentation in which top-level segments are further broken down into sub-segments. While the hierarchical segmentation received a non-negligible research attention (Yaari, 1997; Eisenstein, 2009; Du et al., 2013), the vast majority of the proposed models (including this work) focus on linear segmentation (Hearst, 1994; Beeferman et al., 1999; Choi, 2000; Brants et al., 2002; Misra et al., 2009; Riedl and Biemann, 2012; Glavaš et al., 2016; Koshorek et al., 2018, *inter alia*).

In one of the pioneering segmentation efforts, Hearst (1994) proposed an unsupervised TextTiling algorithm based on the lexical overlap between adjacent sentences and paragraphs. Choi (2000) computes the similarities between sentences in a similar fashion, but renormalizes them within the local context; the segments are then obtained through divisive clustering. Utiyama and Isahara (2001) and Fragkou et al. (2004) minimize the segmentation cost function through exhaustive search with dynamic programming algorithms.

Following the assumption that topical cohesion guides the segmentation of the text, a number of segmentation approaches based on topic models have been proposed. Brants et al. (2002) induce latent representations of text snippets using probabilistic latent semantic analysis (Hofmann, 1999) and segment based on similarities between latent representations of adjacent snippets. Misra et al. (2009) and Riedl and Biemann (2012) leverage topic vectors of snippets obtained with the Latent Dirichlet Allocation model (Blei et al., 2003). While Misra et al. (2009) finds a globally optimal segmentation based on the similarities of snippets' topic vectors using dynamic programming, Riedl and Biemann (2012) adjust the TextTiling model of (Hearst, 1994) to use topic vectors instead of sparse lexicalized representations of snippets.

Malioutov and Barzilay (2006) proposed a first graph-based model for text segmentation. They segment lecture transcripts by first inducing a fully connected sentence graph with edge weights corresponding to cosine similarities between sparse bag-of-word sentence vectors and then running a minimum normalized multiway cut algorithm to obtain the segments. Glavaš et al. (2016) propose GRAPHSEG, a graph-based segmentation algorithm similar in nature to (Malioutov and Barzilay, 2006), which uses dense sentence vectors, obtained by aggregating word embeddings, to compute intra-sentence similarities and performs segmentation based on the cliques of the similarity graph.

Finally, Koshorek et al. (2018) identify Wikipedia as a free large-scale source of manually segmented texts that can be used to train a supervised segmentation model. They train a neural model that hierarchically combines two bidirectional LSTM networks and report massive improvements over unsupervised segmentation on a range of evaluation datasets. The model we presented in this work has a similar hierarchical architecture, but uses Transfomer networks instead of recurrent encoders. Crucially, our CATS model additionally defines an auxiliary coherence modeling objective, which we couple with the primary segmentation objective in a multi-task learning setup.

5.2 Text Coherence

Measuring text coherence amounts to predicting a score that indicates how meaningful the order of the information in the text is. The majority of the proposed text coherence models are grounded in formal theories of text coherence, among which the entity grid model (Barzilay and Lapata, 2008), based on the centering theory of Grosz et al. (1995), is arguably the most popular. The entity grid model represent texts as matrices encoding the grammatical roles that the same entities have in different sentences. The entity grid model, as well as its extensions (Elsner and Charniak, 2011; Feng and Hirst, 2012; Feng et al., 2014; Nguyen and Joty, 2017) require text to be preprocessed – entities extracted and grammatical roles assigned to them – which prohibits an end-to-end model training.

In contrast, Li and Hovy (2014) train a neural model that couples recurrent and recursive sentence encoders with a convolutional encoder of sentence sequences in an end-to-end fashion on limited-size datasets with gold coherence scores. Our models' architecture is conceptually similar, but we use Transformer networks to both encode sentences and sentence sequences. Furthermore, with the goal of supporting text segmentation and not aiming to predict exact coherence scores, our model does not require gold coherence labels; instead we devise a coherence objective that contrasts original text snippets against corrupted sentence sequences.

6 Conclusion

Though segmentation of text is inherently tied to its (local) coherence, text segmentation models capture coherence only implicitly, through lexical or semantic similarities of (adjacent) sentences. In this work, we presented CATS, a novel supervised model for text segmentation that couples segmentation prediction with explicit auxiliary coherence modeling. The proposed model is a neural architecture consisting of two hierarchically connected Transformer networks: the lower-level sentence encoder generates input for the higher-level encoder of sentence sequences. We train the model in a multi-task learning setup by learning to predict (1) segmentation labels of sentences and (2) that original text snippets are more coherent than corrupt sentence sequences. We show that CATS yields state-of-the-art performance on several text segmentation benchmarks and that it can – in a zero-shot language transfer setting, coupled with a cross-lingual word embedding space – successfully segment texts from languages not seen in training.

Although effective for text segmentation, our coherence modeling is still rather simple: we use only fully randomly shuffled sequences as examples of (highly) incoherent text. In subsequent work, we will investigate negative instances of different degree of incoherence as well as better objectives for (auxiliary) modeling of text coherence.

References

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-Wasserstein alignment of word embedding spaces. In *Proceedings of EMNLP*, pages 1881–1890.

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

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of ACL*, pages 789–798.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

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

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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 *Proceedings of the eleventh international conference on Information and knowledge management*, pages 211–218. ACM.

Harr Chen, SRK Branavan, Regina Barzilay, and David R Karger. 2009. Global models of document structure using latent permutations. In *Proceedings 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*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lan Du, Wray Buntine, and Mark Johnson. 2013. Topic segmentation with a structured topic model. In *Proceedings 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 *Proceedings of HLT-NAACL*, pages 353–361. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, pages 462–471.

Vanessa Wei Feng and Graeme Hirst. 2012. Extending the entity-based coherence model with multiple ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 315–324. Association for Computational Linguistics.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 940–949.

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 *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

David Griol and José Manuel Molina. 2015. Do human-agent conversations resemble human-human conversations? In *Distributed Computing and Artificial Intelligence, 12th International Conference*, pages 159–166. Springer.

Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.

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

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of EMNLP*, pages 469–478.

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. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. 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)*, pages 469–473.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of COLING-ACL*, pages 25–32. Association for Computational Linguistics.

Ramesh Manuvinakurike, Maike Paetzel, Cheng Qu, David Schlangen, and David DeVault. 2016. Toward incremental dialogue act segmentation in fast-paced interactive dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 252–262.

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

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330.

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.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Technical Report. Preprint.*

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

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2018. A survey of cross-lingual embedding models. *arXiv preprint arXiv:1706.04902*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

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.

Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR*.

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

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.

Yaakov Yaari. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of RANLP*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING*, pages 947–953.

Tianyu Zhao and Tatsuya Kawahara. 2017. Joint learning of dialog act segmentation and recognition in spoken dialog using neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–712.