A Joint Model for Document Segmentation and Segment Labeling

Joe Barrow*
University of Maryland
jdbarrow@cs.umd.edu

Rajiv Jain
Adobe Research
rajijain@adobe.com

Vlad I. Morariu
Adobe Research
morariu@adobe.com

Varun Manjunatha
Adobe Research
vmanjuna@adobe.com

Douglas W. Oard
University of Maryland
oard@umd.edu

Philip Resnik
University of Maryland
resnik@umd.edu

Abstract

Text segmentation aims to uncover latent structure by dividing text from a document into coherent sections. Where previous work on text segmentation considers the tasks of document segmentation and segment labeling separately, we show that the tasks contain complementary information and are best addressed jointly. We introduce the Segment Pooling LSTM (S-LSTM) model, which is capable of jointly segmenting a document and labeling segments. In support of joint training, we develop a method for teaching the model to recover from errors by aligning the predicted and ground truth segments. We show that S-LSTM reduces segmentation error by 30% on average, while also improving segment labeling.

1 Introduction

A well-written document is rich not only in content but also in structure. One type of structure is the grouping of content into topically coherent segments. These *segmented documents* have many uses across various domains and downstream tasks. Segmentation can, for example, be used to convert unstructured medical dictations into clinical reports (Sadoughi et al., 2018), which in turn can help with medical coding (since a diagnosis mentioned in a "Medical History" might be different from a diagnosis mentioned in an "Intake" section (Ganesan and Subotin, 2014)). Segmentation can also be used downstream for retrieval (Hearst and Plaunt, 2002; Edinger et al., 2017; Allan et al., 1998), where it can be particularly useful when applied to informal text or speech that lacks explicit segment markup. Topically segmented documents are also useful for *pre-reading* (the process of skimming or surveying a text prior to careful reading), thus serving as an aid for reading comprehension (Swaffar et al., 1991; Ajideh, 2003).

Uncovering latent, topically coherent segments of text is a difficult problem because it requires solving a chicken-and-egg problem: determining the segment topics is easier if segment boundaries are given, and identifying the boundaries of segments is easier if the topic(s) addressed in parts of the document are known. Prior approaches to text segmentation can largely be split into two categories that break the cycle by sequentially solving the two problems: those that attempt to directly predict segment bounds (Koshorek et al., 2018), and those that attempt to predict topics per passage (e.g., per sentence) and use measures of coherence for post hoc segmentation (Hearst, 1997; Arnold et al.; Eisenstein and Barzilay, 2008; Riedl and Biemann, 2012; Glavaš et al., 2016). The benefit of the topic modeling approach is that it can work in unsupervised settings where collecting ground truth segmentations is difficult and labeled data is scarce (Eisenstein and Barzilay, 2008; Choi, 2000). Recent work uses Wikipedia as a source of segmentation labels by eliding the segment bounds of a Wikipedia article to train supervised models (Koshorek et al., 2018; Arnold et al.). This enables models to directly learn to predict segment bounds or to learn sentence-level topics and perform post hoc segmentation.

Our work is motivated by the observation that the segment bounds and topicality are tightly interwoven, and should ideally be considered jointly rather than sequentially. We start by examining three properties about text segmentation: (1) segment bounds and segment labels contain complementary supervisory signals, (2) segment labels are a product of lower level (e.g. sentence) labels which must be composed, and (3) the model should not only learn to label from ground-truth segmentations at training time, but instead the labeler should learn to be robust to segmentation errors. These properties build on previous work discussed in Section 2. We

* Work done while interning at Adobe.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 313–322

experimentally evaluate and verify each of these properties in Section 5 with respect to a document segmentation and segment labeling task.

Taking advantage of these properties, we propose a neural model that jointly segments and labels without committing to *a priori* segmentations, Segment Pooling LSTM (S-LSTM). It consists of three components: a segment proposal LSTM (discussed in Section 3.2), a segment pooling layer (Section 3.3), and a segment aligner for training and evaluation (Section 3.4).

Our main contribution is a model that performs segmentation and labeling jointly rather than separately. By virtue of joint inference, our model takes advantage of the complementary supervisory signals for segmentation and topic inference, considers the contribution of all sentences to the segment label, and avoids committing to early errors in low-level inference.

Our approach improves over neural and non-neural baselines of a document segmentation task. We use a dataset of Wikipedia articles described in Section 5 for training and evaluation. We show that S-LSTM is capable of reducing segmentation error by, on average, 30% while also improving segment classification. We also show that these improvements hold on out-of-domain datasets.

2 Related Work

Coherence-based Segmentation. Much work on text segmentation uses measures of coherence to find topic shifts in documents. Hearst (1997) introduced the TextTiling algorithm, which uses term co-occurrences to find coherent segments in a document. Eisenstein and Barzilay (2008) introduced BayesSeg, a Bayesian method that can incorporate other features such as cue phrases. Riedl and Biemann (2012) later introduced TopicTiling, which uses coherence shifts in topic vectors to find segment bounds. Glavaš et al. (2016) proposed GraphSeg, which constructs a semantic relatedness graph over the document using lexical features and word embeddings, and segments using cliques. Nguyen et al. (2012) proposed SITS, a model for topic segmentation in dialogues that incorporates a per-speaker likelihood to change topics.

While the above models are unsupervised, Arnold et al. introduced a supervised method to compute sentence-level topic vectors using Wikipedia articles. The authors created the Wiki-Section dataset and proposed the SECTOR neural model. The SECTOR model predicts a label for each sentence, and then performs post hoc segmentation looking at the coherence of the latent sentence representations, addressing segmentation and labeling separately. We propose a model capable of jointly learning segmentation boundaries and segment-level labels at training time. Our segmentation does not rely on measures of coherence, and can instead learn from signals in the data, such as cue phrases, to predict segment bounds, while still performing well at the segment labeling task.

Supervised Segmentation. An alternative to using measures of topical coherence to segment text is to learn to directly predict segment bounds from labeled data. This was the approach taken in Koshorek et al. (2018), where the authors used Wikipedia as a source of training data to learn text segmentation as a supervised task. However, learning only to predict segment bounds does not necessarily capture the topicality of a segment that is useful for informative labeling.

The task of document segmentation and labeling is well-studied in the clinical domain, where both segmenting and learning segment labels are important tasks. Pomares-Quimbaya et al. (2019) provide a current overview of work on clinical segmentation. Ganesan and Subotin (2014) trained a logistic regression model on a clinical segmentation task, though they did not consider the task of segment labeling. Tepper et al. (2012) considered both tasks of segmentation and segment labeling, and proposed a two-step pipelined method that first segments and then classifies the segments. Our proposed model is trained jointly on both the segmentation and segment labeling tasks.

Concurrent work considers the task of *document outline generation* (Zhang et al., 2019). The goal of outline generation is to segment and generate (potentially hierarchical) headings for each segment. The authors propose the HiStGen model, a hierarchical LSTM model with a sequence decoder. The work offers an alternative view of the joint segmentation and labeling problem, and is evaluated using exact match for segmentation and ROUGE (Lin, 2004) for heading generation if the segment is predicted correctly. In contrast, we evaluate our models using a commonly-used probabilistic segmentation measure, Pk, which assigns partial credit to incorrect segmentations (Beeferman et al., 1999). We also use an alignment technique to assign partial credit to labels of incorrect segmentations, both for

training and evaluation. In addition, we explicitly consider the problem of model transferability, evaluating the pretrained models on additional datasets.

IOB Tagging. The problem of jointly learning to segment and classify is well-studied in NLP, though largely at a lower level, with Inside-Outside-Beginning (IOB) tagging (Ramshaw and Marcus, 1999). Conditional random field (CRF) decoding has long been used with IOB tagging to simultaneously segment and label text, e.g. for named entity recognition (NER, McCallum and Li, 2003). The models that perform best at joint segmentation/classification tasks like NER or phrase chunking were IOB tagging models, typically LSTMs with a CRF decoder (Lample et al., 2016) until BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018). Tepper et al. (2012) proposed the use of IOB tagging to segment and label clinical documents, but argued for a pipelined approach.

CRF-decoded IOB tagging models are more difficult to apply to the multilabel case. Segment bounds need to be consistent across all labels, so modeling the full transition from $|L| \longrightarrow |L|$ (where $|L|$ is the size of the label space) at every time step is computationally expensive. In contrast, our joint model performs well at multilabel prediction, while also outperforming a neural CRF-decoded model on a *single*-label labeling task.

3 Modeling

In order to jointly model document segmentation and segment classification, we introduce the Segment Pooling LSTM (S-LSTM) model. S-LSTM is a supervised model trained to both predict segment bounds and pool over and classify the segments. The model consists of three components: a **sentence encoder** (Section 3.1), a **segment predictor LSTM** (Section 3.2), and a **segment pooling network** which pools over predicted segments to classify them (Section 3.3). The segment predictor is allowed to make mistakes that the labeler must learn to be robust to, a process which we refer to as exploration, and accomplish by aligning predicted and ground truth segments (Section 3.4). The full architecture is presented in Figure 1, and the loss is discussed in Section 3.5.

3.1 Encoding Sentences

The first stage is encoding sentences. S-LSTM is agnostic to the choice of sentence encoder, though in this work we use a concat pooled bi-directional LSTM (Howard and Ruder, 2018). First, the embedded words are passed through the LSTM encoder. Then, the maximum and mean of all hidden states are concatenated with the final hidden states, and this is used as the sentence encoding.

3.2 Predicting Segment Bounds

The second step of our model is a Segment Predictor LSTM, which predicts segment boundaries within the document. For this step we use a bidirectional LSTM that consumes each sentence vector and predicts an indicator variable, (B)eginning or (I)nside a segment. It is trained from pre-segmented documents using a binary cross entropy loss. This indicator variable determines if the sentence is the start of a new segment or not. This is similar to the approach taken by TextSeg in Koshorek et al. (2018), though we do not estimate a threshold, τ, and instead learn to to predict two classes: (B)eginning and (I)nside.

3.3 Segment Pooling

After segmenting the document, the third stage of the model pools within the predicted segments to predict a label for each segment. The sentence vectors for the predicted segments are all grouped, and a pooling function is run over them. There are several possible sequence-to-vector pooling functions that could be used, such as averaging, and more complex learned pooling functions, such as LSTMs. The full S-LSTM model uses a concat pooling LSTM, and our experimental results show that this yields a better segment label than just averaging. We then use a classifier following the output of the segment pooler, which can provide a distribution over labels for each segment.

The combination of segment prediction and pooling is one way that S-LSTM is different from previous hierarchical LSTM models. The model can predict and label segments dynamically, generating a single vector for *predicted* segments.

3.4 Segment Alignment and Exploration

Because segments can be considered dynamically at training time, we propose a method of assigning labels to potentially incorrect segments by aligning the predicted segments with ground truth segments. This label assignment allows segment-labeling loss to be propagated through the end-to-end model.

Teacher Forcing. Teacher forcing, or feeding ground truth inputs into a recurrent network as

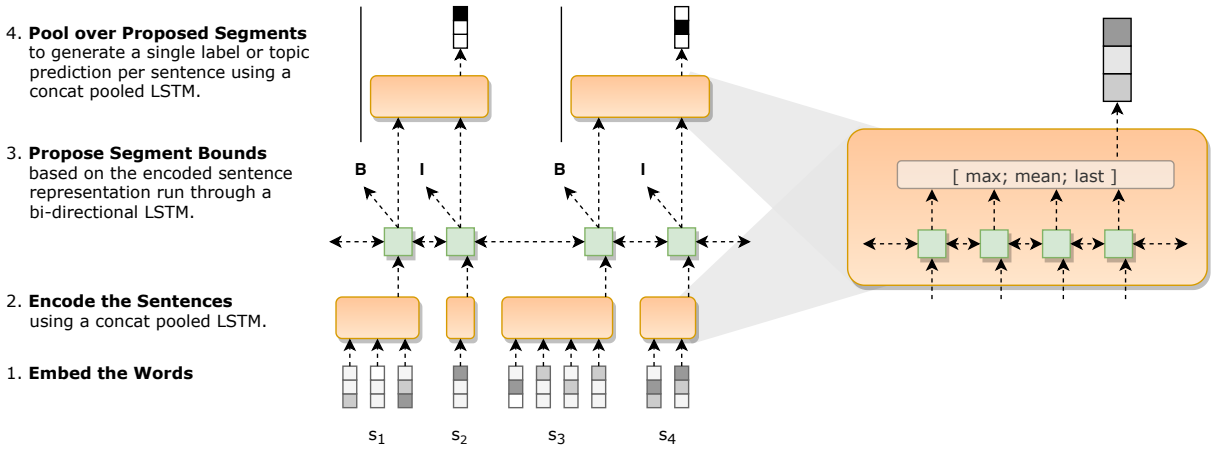

Figure 1: Segment Pooling LSTM (S-LSTM) architecture. The network first proposes segment bounds based on text, and then pools over sentence representations in the proposed segment to generate a segment label.

opposed to model predictions, was first developed in Williams and Zipser (1989). The idea is to use ground truth predictions for inputs that would normally come from model predictions for the first stages of training, to help with convergence. For S-LSTM, it is the simplest approach to segment pooling and alignment: at training time feed the ground truth segments (as opposed to the predicted segments) the segment pooler (step 3 in Figure 1). This gives us a one-to-one alignment of "predicted" (forced) segments and ground truth segments. This is opposed to only using the *predicted* segments as the bounds for segment pooler.

Exploration. Employing only teacher forcing does not allow the segment labeler to learn how to recover from errors in segmentation. The mechanism for allowing the model to explore incorrect segmentations is to align the predicted segments with overlapping ground truth segments at training time, and treat the all aligned ground truth labels as correct. While many alignments are possible, we use the one presented in Figure 2. This many-to-many alignment ensures that every ground-truth segment is mapped to at least one predicted segment and every predicted segment is mapped to at least one ground truth segment.

We can additionally schedule teacher forcing. At the beginning, when the segmentation prediction network performs poorly, the model pools over only ground truth segment bounds, allowing it to learn the cleanest topic representations. However, as training progresses and the segmentation accuracy begins to converge, we switch from pooling over ground truth segments to aligning predicted and ground truth segment. In this way, the segment pooler learns to be robust to segmentation errors.

3.5 Joint Training

To jointly train the model, we use a multi-task loss,

$$L(X, y; \theta) = \alpha \cdot L_{seg}(X, y_{seg}; \theta_{seg}) + (1 - \alpha) \cdot L_{cls}(X, y_{cls}; \theta_{cls}, aligner),$$

where y_{seg} are the labels for the segment prediction LSTM and y_{cls} are segment labels. In addition, we pass in an $aligner$, which determines how to align the predicted segments with the ground truth segments to compute the loss, and either teacher forces the model or allows it to explore.

4 Experimental Setup

We follow the experimental procedure of Arnold et al. to evaluate S-LSTM for the tasks of document segmentation and segment labeling.

4.1 Datasets

WikiSection. Arnold et al. introduced the WikiSection dataset, which contains Wikipedia articles across two languages (English and German) and domains (Cities and Diseases). Articles are segmented using the Wikipedia section structure. The heading of each segment is retained, as well as a normalized label for each heading type (e.g. History, Demography), drawn from a restricted label vocabulary. There are two tasks: (1) jointly segment the document and assign a single restricted-vocabulary label to the segment, and (2) predict the bag-of-words in the title of the Wikipedia section as a label. For instance, the bag-of-words label for the title of this section would be the words:

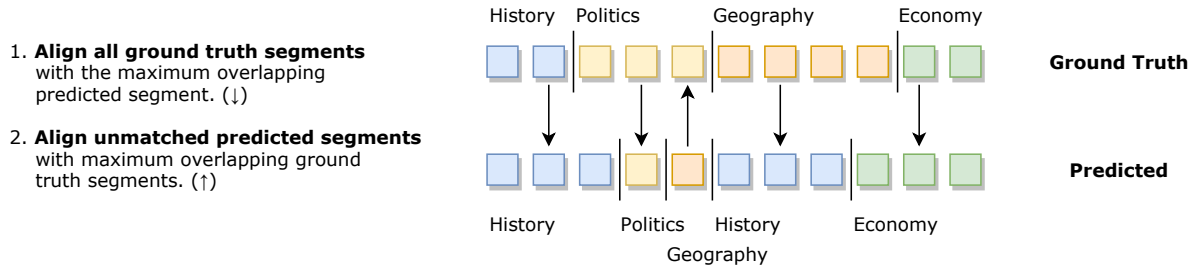

Figure 2: Greedy many-to-many alignment. This alignment is used to assign ground-truth labels to predicted segments for training. Each ground truth segment first aligns to the maximally overlapping predicted segment; each leftover predicted segment then aligns to the maximally overlapping ground truth segment.

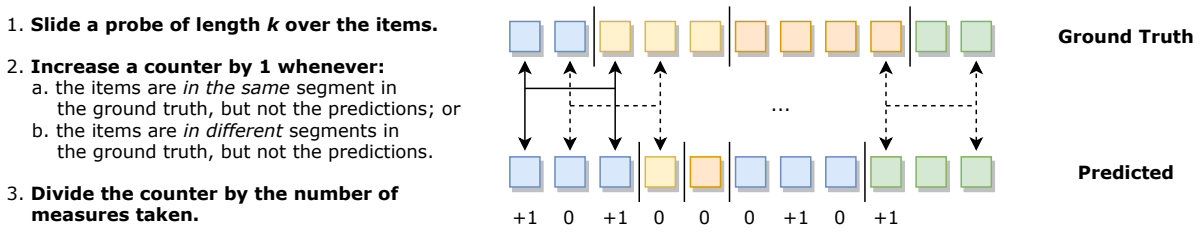

Figure 3: Computing P_k. A sliding window of length k is run over the text, and a counter increments whenever the same/different status for the two ends of the window doesn't match in the ground truth and predicted segmentation.

[Dataset, Experimental, Setup].[1] For the second task, we post-process headers to remove stopwords, numbers and punctuation. We then remove words that occur fewer than 20 times in the training data to get the final label vocabulary sizes.

Of note, we encountered a smaller label vocabulary for the bag-of-words generation task than that reported by Arnold et al.. For the four datasets, the original reported sizes of the header vocabularies were: [1.5k 1.0k, 2.8k, 1.1k]. When reproducing earlier results, we verified with the dataset authors that the actual sizes were: [179, 115, 603, 318].

The first task aligns closely with the clinical domain, in which headers are typically drawn from a fixed label set (Tepper et al., 2012). The second aligns more closely with learning to segment and label from naturally labeled data, such as contracts or Wikipedia articles, which can potentially then be transferred (Koshorek et al., 2018).

Wiki-50. The *Wiki-50* dataset was introduced as a test set in Koshorek et al. (2018), which also introduced the full *Wiki-727k* dataset. The dataset contains 50 randomly sampled Wikipedia articles, segmented and with their headers, and was used to evaluate computationally expensive methods such as BAYESSEG (Eisenstein and Barzilay, 2008).

Cities and Elements. The *Cities* and *Elements* datasets were introduced in Chen et al. (2009). They provide two additional Wikipedia datasets with both segmentation and segment headers.

Clinical. We use the *Clinical Textbook* dataset from Eisenstein and Barzilay (2008), which has segment boundaries but no headings.

4.2 Experimental Design

We evaluate S-LSTM with previous document segmentation and segment labeling approaches on all four WikiSection datasets— English-language Diseases (**en_disease**), German-language Diseases (**de_disease**), English-language Cities (**en_city**), and German-language Cities (**de_city**)—for both the single label and multi-label tasks.

Model Ablation. In order to understand the effect of our proposed segment pooling and segment exploration strategies, we also include results for simpler baselines for each of these modules. For the segment labeling we report not only the full S-LSTM model with LSTM pooling, but also additionally a mean pooling model, which we denote with "-pool". For the segment exploration we report not only the model with exploration, but also a model only trained using teacher forcing, which we denote with "-expl".

Model Transferability. To evaluate model transferability, we test models trained on the English

[1] Subsection bags-of-words labels include the dominating section heading.

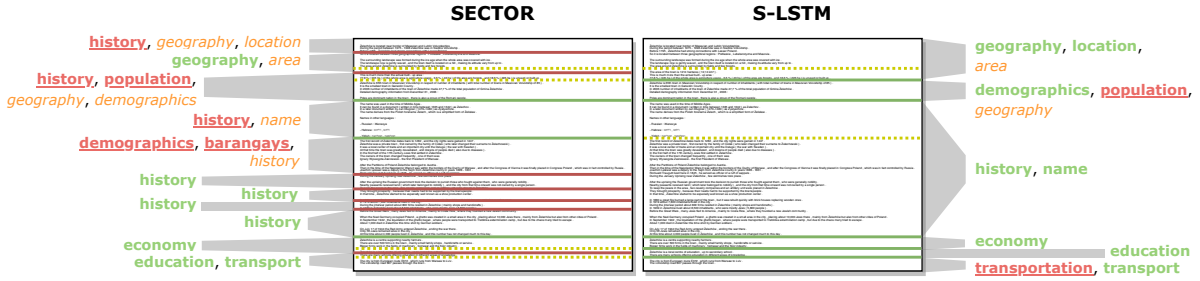

Figure 4: A randomly selected document from the **en_cities** test set, with the output of SECTOR (left) and S-LSTM (right). Green lines are a correctly predicted segment bound, red lines are false positive bound predictions, and yellow dashed lines are false negatives. For each segment, the top 1-2 predicted terms are also shown. Terms are **bold green** if they appear in the maximally overlapping segment in the ground truth, **underlined red** if they are false positive terms, and *italicized yellow* if they are false negatives. S-LSTM does not predict any false positive segment bounds, and makes only a small number of labeling errors compared with the SECTOR baseline.

WikiSection tasks (**en_disease** and **en_city**) on the *Cities*, *Elements*, *Wiki-50*, and *Clinical* datasets.

4.3 Evaluation Measures

Segmentation: Pk. P_k is a probabilistic measure (Beeferman et al., 1999) that works by running a sliding window of width k over the predicted and ground truth segments, and counting the number of times there is disagreement about the ends of the probe being in the same or different sections (see Figure 3). The number of disagreements is then divided by the total number of window positions, resulting in a score normalized between 0 and 1. Our segmentation results are reported setting k to half the average size of ground truth segments.

Classification: F1, MAP, and Prec@1. For classification, we report three different measures, depending on the task. For the single label tasks, we report F_1 and Mean Average Precision (MAP). For evaluating the bag-of-words (multi-label) tasks, we report Precision at the first rank position (Prec@1) and MAP. In both cases, these are computed by first aligning the predicted segments with the ground truth segments as shown in Figure 2 and described in Section 3.4. In all cases, the metrics are micro-averaged.

4.4 Baselines

We report C99 (Choi, 2000), TopicTiling (Riedl and Biemann, 2012), and TextSeg (Koshorek et al., 2018) as baselines on WikiSection segmentation. For a neural baseline, we report the SECTOR model (Arnold et al.) with pre-trained embeddings, denoted in the paper as SEC>T,H+emb. For the additional datasets, we report GraphSeg (Glavaš et al., 2016), BayesSeg (Eisenstein and Barzilay, 2008) and pretrained TextSeg and SECTOR models.

In addition, we implemented an LSTM-LSTM-CRF IOB tagging model following Lample et al. (2016). This is only used for the single-label experiments, as CRF-decoded IOB tagging models are more difficult to apply to the multilabel case.

4.5 Model Setup

For each task and dataset, we use the same set of hyperparameters: Adam optimizer (Kingma and Ba, 2015) with learning rate 0.001 and weight decay 0.9. Dropout (Srivastava et al., 2014) is applied after each layer except the final classification layers; we use a single dropout probability of 0.1 for every instance. For models with exploration, we employ teacher forcing for 10 epochs. Model weights are initialized using Xavier normal initialization (Glorot and Bengio, 2010). All LSTM hidden-layer sizes are set to 200. We use fixed 300-dimensional FastText embeddings (Bojanowski et al., 2017) for both English and German, and project them down to 200 dimensions using a trainable linear layer.

5 Results and Analysis

There are five major takeaways from the experimental results and analysis. First, the jointly trained S-LSTM model shows major improvement over prior work that modeled document segmentation and segment labeling tasks separately. Second, segment alignment and exploration during training reduces error rates. Third, the segment pooling layer leads to improvements for both segmentation and segment labeling. Fourth, S-LSTM outperforms an IOB-tagging CRF-decoded model for single label segment labeling, and *also* generalizes easily

WikiSection-topics single-label classification	**en_disease** 27 topics			**de_disease** 25 topics			**en_city** 30 topics			**de_city** 27 topics		
model configuration	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP
C99	37.4	n/a	n/a	42.7	n/a	n/a	36.8	n/a	n/a	38.3	n/a	n/a
TopicTiling	43.4	n/a	n/a	45.4	n/a	n/a	30.5	n/a	n/a	41.3	n/a	n/a
TextSeg	24.3	n/a	n/a	35.7	n/a	n/a	19.3	n/a	n/a	27.5	n/a	n/a
SEC>T+emb	26.3	55.8	69.4	27.5	48.9	65.1	15.5	71.6	81.0	16.2	71.0	81.1
LSTM-LSTM-CRF	23.9	57.2	n/a	23.6	51.4	n/a	9.7	**77.5**	n/a	10.2	74.0	n/a
S-LSTM	**20.0**	**59.3**	**72.4**	**18.8**	**55.6**	**69.0**	**9.1**	76.1	**83.5**	**9.5**	**76.5**	**84.5**

Table 1: WikiSection results. Baselines are TopicTiling (Riedl and Biemann, 2012), TextSeg (Koshorek et al., 2018), and C99 (Choi, 2000), and the best neural SECTOR models from Arnold et al..

WikiSection-headings multi-label classification	**en_disease** 179 topics			**de_disease** 115 topics			**en_city** 603 topics			**de_city** 318 topics		
model configuration	↓ P_k	↑ Prec@1	↑ MAP	↓ P_k	↑ Prec@1	↑ MAP	↓ P_k	↑ Prec@1	↑ MAP	↓ P_k	↑ Prec@1	↑ MAP
C99	37.4	n/a	n/a	42.7	n/a	n/a	36.8	n/a	n/a	38.3	n/a	n/a
TopicTiling	43.4	n/a	n/a	45.4	n/a	n/a	30.5	n/a	n/a	41.3	n/a	n/a
TextSeg	24.3	n/a	n/a	35.7	n/a	n/a	19.3	n/a	n/a	27.5	n/a	n/a
SEC>H+emb	30.7	50.5	57.3	32.9	26.6	36.7	17.9	72.3	**71.1**	19.3	68.4	70.2
S-LSTM	**19.8**	**53.5**	**60.3**	**18.6**	**36.2**	**46.1**	**9.0**	**73**	**71.3**	**8.2**	**74.1**	**75.1**
S-LSTM, -expl	20.8	52.1	59	19.1	34.7	44.8	9.2	72.7	70.8	8.5	73.8	74.4
S-LSTM, -expl, -pool	21.2	52.3	59.5	19.8	34.4	45	10.4	69.7	67.2	10.2	64.1	66.7

Table 2: WikiSection headings task results, which predicts a multi-label bag-of-words drawn from section headers. To show the effect of the segment pooling and model exploration used in S-LSTM we report two variants where -expl uses only teacher forcing and -pool uses only mean pooling.

and tractably to multi-labeling. Fifth, a deeper analysis of the joint modeling demonstrates that segment labeling and segment bound prediction contain complementary information.

5.1 Structure Predicts Better Structure

Tables 1 and 2 show that by explicitly predicting segment bounds we can improve segmentation by a large margin. On the header prediction task (Table 2), we reduced P_k by an average of over 30% across the WikiSection datasets. P_k was consistent across both WikiSection tasks, and did not degrade when going from single-label to multi-label prediction, as Arnold et al. had found. This shows that we can achieve a more robust segmentation through jointly modeling segmentation and labeling. This is also clear from Figure 4, where S-LSTM predicts a much more accurate segmentation.

5.2 Exploration Allows Error Recovery

The results of an ablation experiment (Table 2, bottom) show that there is an additional classification gain by allowing the model to explore recovering from segmentation errors. Exploration has the important property of allowing the model to optimize more closely to how it is being evaluated. This follows from a long line of work in NLP that shows that for tasks such as dependency parsing (Ballesteros et al., 2016), constituency parsing (Goodman, 1996), and machine translation (Och, 2003), all show improvements by optimizing on a loss that aligns with evaluation.

The teacher forcing was important at the beginning of model training. When training variants of S-LSTM that did not use teacher forcing at the beginning, which instead could explore the bad segmentation, the segmentation failed to converge and the model performed universally poorly.

5.3 S-LSTM Can Take Advantage of Both of These, Plus Segment Pooling

S-LSTM is capable of taking advantage of the complementary information by jointly learning to segment and label. It is capable of learning to recover from segmentation errors by exploring towards the end of training. But the ablation study shows that there is one more important component of S-LSTM that allows it to improve over previous baselines: LSTM pooling over segments. The addition of the segment pooling layer improves MAP and Prec@1 across all four datasets in the heading prediction task (Table 2), comparing the model without exploration (S-LSTM,-expl) with the model without exploration (which uses average pooling: S-LSTM,-

Segmentation	**Wiki-50**		**Cities**		**Elements**		**Clinical**
and multi-label classification	$\downarrow P_k$	$\uparrow$ MAP	$\downarrow P_k$	$\uparrow$ MAP	$\downarrow P_k$	$\uparrow$ MAP	$\downarrow P_k$
GraphSeg	63.6	n/a	40.0	n/a	49.1	n/a	–
BayesSeg	49.2	n/a	36.2	n/a	35.6	n/a	57.8
TextSeg	**18.2***	n/a	**19.7***	n/a	41.6	n/a	**30.8**
SEC>H+emb@en_disease	–	–	–	–	43.3	9.5	36.5
SEC>H+emb@en_city	40.5	13.4	33.3	53.6	41.0	7.9	–
S-LSTM@en_city	22.7	**16.6**	21.2	**54.2**	34.5	11.0	–
S-LSTM@en_disease	–	–	–	–	**30.2**	**19.1**	36.1

Table 3: Transfer results across four datasets. Those marked * are trained on the training portion of the corresponding dataset, whereas those without are either unsupervised or trained on a different dataset. For the *Wiki-50*, *Cities*, and *Elements* datasets, S-LSTM outperforms all models not trained on corresponding training set.

WikiSection-headings multi-label classification	**de_disease** 115 topics		
model configuration	$\downarrow P_k$	$\uparrow$ P@1	$\uparrow$ MAP
S-LSTM, w/o Segment Prediction	n/a	42.3	52.1
S-LSTM, w/ Segment Prediction	19.1	**43.3**	**53.3**

Table 4: A model trained to jointly predict segment bounds and segment labels improves classification over a baseline which only predicts labels. Both are given oracle segment bounds and do not use exploration.

WikiSection-headings document segmentation	**de_disease** 115 topics		
model configuration	$\downarrow P_k$	$\uparrow$ P@1	$\uparrow$ MAP
S-LSTM, w/o Segment Labeling	21.8	n/a	n/a
S-LSTM, w/ Segment Labeling	**19.1**	34.7	44.8

Table 5: Inverse of the experiment in Table 4. A model that jointly predicts segment bounds and labels outperforms a model that only predicts segment bounds.

expl,-pool). It is the combination of these three improvements that comprise the full S-LSTM.

5.4 S-LSTM Outperforms a CRF Baseline

In Table 1, the results demonstrate that S-LSTM outperforms LSTM-LSTM-CRF baseline in almost every case for single-labeling, and in every case for segmentation. This makes S-LSTM a useful model choice for cases like clinical segmentation and labeling, where segments are drawn from a small fixed vocabulary. S-LSTM also generalizes easily to multi-label problems, in contrast to an IOB-tagging LSTM-LSTM-CRF, since it only requires changing the segment-pooling loss from cross-entropy to binary cross-entropy.

5.5 Predicting Structure Predicts Better Labels (and vice versa)

Though we compare with TextSeg (a neural model that predicts segment bounds) and SECTOR (a neural model that predicts sentence labels and post hoc segments them) and show improvements compared to both models, we also directly test the hypothesis that the segmentation and segment labeling tasks contain complementary information. To do so, we conduct two experiments: (1) we fix the segment bounds at training and evaluation time, only training the model to label known segments (results in Table 5); and (2) we only have the model predict segment bounds (results in Table 4).

In both cases, the addition of the loss from the companion task improves performance on the main task. This shows that the two tasks contain complementary information, and directly validates our core hypothesis that the two tasks are tightly interwoven. Thus, considering them jointly improves performance on both tasks.

6 Conclusion and Future Work

In this paper we introduce the Segment Pooling LSTM (S-LSTM) model for joint segmentation and segment labeling tasks. We find that the model dramatically reduces segmentation error (by 30% on average across four datasets) while improving segment labeling accuracy compared to previous neural and non-neural baselines for both single-label and multi-label tasks. Experiments demonstrate that jointly modeling the segmentation and segment labeling, segmentation alignment and exploration, and segment pooling each contribute to S-LSTM's improved performance.

S-LSTM is agnostic as to the sentence encoder used, so we would like to investigate the potential

usefulness of transformer-based language models as sentence encoders. There are additional engineering challenges associated with using models such as BERT as sentence encoders, since encoding entire documents can be too expensive to fit on a GPU without model parallelism. We would also like to investigate the usefulness of an unconsidered source of document structure: the hierarchical nature of sections and subsections. Like segment bounds and headers, this structure is naturally available in Wikipedia. Having shown that segment *bounds* contain useful supervisory signal, it would be interesting to examine if segment *hierarchies* might also contain useful signal.

Acknowledgements

The authors would like to thank Sebastian Arnold for his feedback and responsiveness. We would also like to thank others for their feedback, including Franck Dernoncourt, Sasha Spala, Nick Miller, Han-Chin Shing, Pedro Rodriguez, Denis Peskov, and Yogarshi Vyas. This work was supported through Adobe Gift Funding, which supports an Adobe Research-University of Maryland collaboration. It was completed while the primary author was interning at Adobe Research.

References

Parviz Ajideh. 2003. Schema theory-based pre-reading tasks: A neglected essential in the esl reading class. *The Reading Matrix*, 3(1).

James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*.

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

Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A Smith. 2016. Training with exploration improves a greedy stack-LSTM parser. In *Proceedings of Empirical Methods in Natural Language Processing*.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5.

Harr Chen, SRK Branavan, Regina Barzilay, and David R Karger. 2009. Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, 36.

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

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

Tracy Edinger, Dina Demner-Fushman, Aaron M Cohen, Steven Bedrick, and William Hersh. 2017. Evaluation of clinical text segmentation to facilitate cohort retrieval. In *AMIA Annual Symposium Proceedings*.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Kavita Ganesan and Michael Subotin. 2014. A general supervised approach to segmentation of clinical texts. In *IEEE International Conference on Big Data*.

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

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of Artificial Intelligence and Statistics*.

Joshua Goodman. 1996. Parsing algorithms and metrics. In *Proceedings of the Association for Computational Linguistics*.

Marti Hearst and Christian Plaunt. 2002. Subtopic structuring for full-length document access. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1).

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the Association for Computational Linguistics*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Chin-Yew Lin. 2004. Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough? In *NTCIR*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. 2012. SITS: A hierarchical nonparametric model using speaker identity for topic segmentation in multiparty conversations. In *Proceedings of the Association for Computational Linguistics*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Alexandra Pomares-Quimbaya, Markus Kreuzthaler, and Stefan Schulz. 2019. Current approaches to identify sections within clinical narratives from electronic health records: a systematic review. *BMC medical research methodology*, 19(1).

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer.

Martin Riedl and Chris Biemann. 2012. TopicTiling: a text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*.

Najmeh Sadoughi, Greg P Finley, Erik Edwards, Amanda Robinson, Maxim Korenevsky, Michael Brenndoerfer, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018. Detecting section boundaries in medical dictations: Toward real-time conversion of medical dictations to clinical reports. In *International Conference on Speech and Computer*. Springer.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1).

Janet K Swaffar, Katherine Arens, and Heidi Byrnes. 1991. *Reading for meaning: An integrated approach to language learning*. Pearson College Division.

Michael Tepper, Daniel Capurro, Fei Xia, Lucy Vanderwende, and Meliha Yetisgen-Yildiz. 2012. Statistical section segmentation in free-text clinical records. In *Proceedings of the Language Resources and Evaluation Conference*.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2).

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.