SECTOR: A Neural Model for Coherent Topic Segmentation and Classification

Sebastian Arnold
Rudolf Schneider
Beuth University of Applied
Sciences Berlin, Germany
{sarnold, ruschneider}@
beuth-hochschule.de

Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

Felix A. Gers
Alexander Löser
Beuth University of Applied
Sciences Berlin, Germany
{gers, aloeser}@
beuth-hochschule.de

Abstract

When searching for information, a human reader first glances over a document, spots relevant sections, and then focuses on a few sentences for resolving her intention. However, the high variance of document structure complicates the identification of the salient topic of a given section at a glance. To tackle this challenge, we present SECTOR, a model to support machine reading systems by segmenting documents into coherent sections and assigning topic labels to each section. Our deep neural network architecture learns a latent topic embedding over the course of a document. This can be leveraged to classify local topics from plain text and segment a document at topic shifts. In addition, we contribute WikiSection, a publicly available data set with 242k labeled sections in English and German from two distinct domains: diseases and cities. From our extensive evaluation of 20 architectures, we report a highest score of 71.6% F1 for the segmentation and classification of 30 topics from the English city domain, scored by our SECTOR long short-term memory model with Bloom filter embeddings and bidirectional segmentation. This is a significant improvement of 29.5 points F1 over state-of-the-art CNN classifiers with baseline segmentation.

1 Introduction

Today's systems for natural language understanding are composed of building blocks that extract semantic information from the text, such as named entities, relations, topics, or discourse structure. In traditional natural language processing (NLP), these extractors are typically applied to bags of words or full sentences (Hirschberg and Manning, 2015). Recent neural architectures build upon pretrained word or sentence embeddings (Mikolov et al., 2013; Le and Mikolov, 2014), which focus on semantic relations that can be learned from large sets of paradigmatic examples, even from long ranges (Dieng et al., 2017).

From a human perspective, however, it is mostly the authors themselves who help best to understand a text. Especially in long documents, an author thoughtfully designs a readable structure and guides the reader through the text by arranging topics into coherent passages (Glavaš et al., 2016). In many cases, this structure is not formally expressed as section headings (e.g., in news articles, reviews, discussion forums) or it is structured according to domain-specific aspects (e.g., health reports, research papers, insurance documents).

Ideally, systems for text analytics, such as topic detection and tracking (TDT) (Allan, 2002), text summarization (Huang et al., 2003), information retrieval (IR) (Dias et al., 2007), or question answering (QA) (Cohen et al., 2018), could access a document representation that is aware of both *topical* (i.e., latent semantic content) and *structural* information (i.e., segmentation) in the text (MacAvaney et al., 2018). The challenge in building such a representation is to combine these two dimensions that are strongly interwoven in the author's mind. It is therefore important to understand topic segmentation and classification as a mutual task that requires encoding both topic information and document structure coherently.

In this paper, we present SECTOR,[1] an end-to-end model that learns an embedding of latent topics

---

[1]Our source code is available under the Apache License 2.0 at `https://github.com/sebastianarnold/SECTOR`.

169

from potentially ambiguous headings and can be applied to entire documents to predict local topics on sentence level. Our model encodes topical information on a vertical dimension and structural information on a horizontal dimension. We show that the resulting embedding can be leveraged in a downstream pipeline to segment a document into coherent sections and classify the sections into one of up to 30 topic categories reaching 71.6% F_1—or alternatively, attach up to 2.8k topic labels with 71.1% mean average precision (MAP). We further show that segmentation performance of our bidirectional long short-term memory (LSTM) architecture is comparable to specialized state-of-the-art segmentation methods on various real-world data sets.

To the best of our knowledge, the combined task of segmentation and classification has not been approached on the full document level before. There exist a large number of data sets for text segmentation, but most of them do not reflect real-world topic drifts (Choi, 2000; Sehikh et al., 2017), do not include topic labels (Eisenstein and Barzilay, 2008; Jeong and Titov, 2010; Glavaš et al., 2016), or are heavily normalized and too small to be used for training neural networks (Chen et al., 2009). We can utilize a generic segmentation data set derived from Wikipedia that includes headings (Koshorek et al., 2018), but there is also a need in IR and QA for supervised structural topic labels (Agarwal and Yu, 2009; MacAvaney et al., 2018), different languages and more specific domains, such as clinical or biomedical research (Tepper et al., 2012; Tsatsaronis et al., 2012), and news-based TDT (Kumaran and Allan, 2004; Leetaru and Schrodt, 2013).

Therefore we introduce WIKISECTION,[2] a large novel data set of 38k articles from the English and German Wikipedia labeled with 242k sections, original headings, and normalized topic labels for up to 30 topics from two domains: *diseases* and *cities*. We chose these subsets to cover both clinical/biomedical aspects (e.g., symptoms, treatments, complications) and news-based topics (e.g., history, politics, economy, climate). Both article types are reasonably well-structured according to Wikipedia guidelines (Piccardi et al., 2018), but we show that they are also comple-

---

mentary: Diseases is a typical scientific domain with low entropy (i.e., very narrow topics, precise language, and low word ambiguity). In contrast, cities resembles a diversified domain, with high entropy (i.e., broader topics, common language, and higher word ambiguity) and will be more applicable to for example, news, risk reports, or travel reviews.

We compare SECTOR to existing segmentation and classification methods based on latent Dirichlet allocation (LDA), paragraph embeddings, convolutional neural networks (CNNs), and recurrent neural networks (RNNs). We show that SECTOR significantly improves these methods in a combined task by up to 29.5 points F_1 when applied to plain text with no given segmentation.

The rest of this paper is structured as follows: We introduce related work in Section 2. Next, we describe the task and data set creation process in Section 3. We formalize our model in Section 4. We report results and insights from the evaluation in Section 5. Finally, we conclude in Section 6.

2   Related Work

The analysis of emerging topics over the course of a document is related to a large number of research areas. In particular, topic modeling (Blei et al., 2003) and TDT (Jin et al., 1999) focus on representing and extracting the semantic topical content of text. Text segmentation (Beeferman et al. 1999) is used to split documents into smaller coherent chunks. Finally, text classification (Joachims 1998) is often applied to detect topics on text chunks. Our method unifies those strongly interwoven tasks and is the first to evaluate the combined topic segmentation and classification task using a corresponding data set with long structured documents.

Topic modeling is commonly applied to entire documents using probabilistic models, such as LDA (Blei et al., 2003). AlSumait et al. (2008) introduced an online topic model that captures emerging topics when new documents appear. Gabrilovich and Markovitch (2007) proposed the Explicit Semantic Analysis method in which concepts from Wikipedia articles are indexed and assigned to documents. Later, and to overcome the vocabulary mismatch problem, Cimiano et al. (2009) introduced a method for assigning latent concepts to documents. More recently, Liu et al. (2016) represented documents with vectors of

closely related domain keyphrases. Yeh et al. (2016) proposed a conceptual dynamic LDA model for tracking topics in conversations. Bhatia et al. (2016) utilized Wikipedia document titles to learn neural topic embeddings and assign document labels. Dieng et al. (2017) focused on the issue of long-range dependencies and proposed a latent topic model based on RNNs. However, the authors did not apply the RNN to predict local topics.

Text segmentation has been approached with a wide variety of methods. Early unsupervised methods utilized lexical overlap statistics (Hearst 1997; Choi 2000), dynamic programming (Utiyama and Isahara, 2001), Bayesian models (Eisenstein and Barzilay, 2008), or pointwise boundary sampling (Du et al., 2013) on raw terms.

Later, supervised methods included topic models (Riedl and Biemann, 2012) by calculating a coherence score using dense topic vectors obtained by LDA. Bayomi et al. (2015) exploited ontologies to measure semantic similarity between text blocks. Alemi and Ginsparg (2015) and Naili et al. (2017) studied how word embeddings can improve classical segmentation approaches. Glavaš et al. (2016) utilized semantic relatedness of word embeddings by identifying cliques in a graph.

More recently, Sehikh et al. (2017) utilized LSTM networks and showed that cohesion between bidirectional layers can be leveraged to predict topic changes. In contrast to our method, the authors focused on segmenting speech recognition transcripts on word level without explicit topic labels. The network was trained with supervised pairs of contrary examples and was mainly evaluated on artificially segmented documents. Our approach extends this idea so it can be applied to dense topic embeddings which are learned from raw section headings.

Wang et al. (2017) tackled segmentation by training a CNN to learn coherence scores for text pairs. Similar to Sehikh et al. (2017), the network was trained with short contrary examples and no topic objective. The authors showed that their pointwise ranking model performs well on data sets by Jeong and Titov (2010). In contrast to our method, the ranking algorithm strictly requires a given ground truth number of segments for each document and no topic labels are predicted.

Koshorek et al. (2018) presented a large new data set for text segmentation based on Wikipedia that includes section headings. The authors introduced a neural architecture for segmentation that is based on sentence embeddings and four layers of bidirectional LSTM. Similar to Sehikh et al. (2017), the authors used a binary segmentation objective on the sentence level, but trained on entire documents. Our work takes up this idea of end-to-end training and enriches the neural model with a layer of latent topic embeddings that can be utilized for topic classification.

Text classification is mostly applied at the paragraph or sentence level using machine learning methods such as support vector machines (Joachims, 1998) or, more recently, shallow and deep neural networks (Le et al., 2018; Conneau et al., 2017). Notably, paragraph vectors (Le and Mikolov, 2014) is an extension of word2vec for learning fixed-length distributed representations from texts of arbitrary length. The resulting model can be utilized for classification by providing paragraph labels during training. Furthermore, Kim (2014) has shown that CNNs combined with pre-trained task-specific word embeddings achieve the highest scores for various text classification tasks.

Combined approaches of topic segmentation and classification are rare to find. Agarwal and Yu (2009) classified sections of BioMed Central articles into four structural classes (introduction, methods, results, and discussion). However, their manually labeled data set only contains a sample of sentences from the documents, so they evaluated sentence classification as an isolated task. Chen et al. (2009) introduced two Wikipedia-based data sets for segmentation, one about large cities, the second about chemical elements. Although these data sets have been used to evaluate word-level and sentence-level segmentation (Koshorek et al., 2018), we are not aware of any topic classification approach on this data set.

Tepper et al. (2012) approached segmentation and classification in a clinical domain as supervised sequence labeling problem. The documents were segmented using a maximum entropy model and then classified into 11 or 33 categories. A similar approach by Ajjour et al. (2017) used sequence labeling with a small number of 3–6 classes. Their model is extractive, so it does not produce a continuous segmentation over the entire document. Finally, Piccardi et al. (2018) did not approach segmentation, but recommended an ordered set of section labels based on Wikipedia articles.
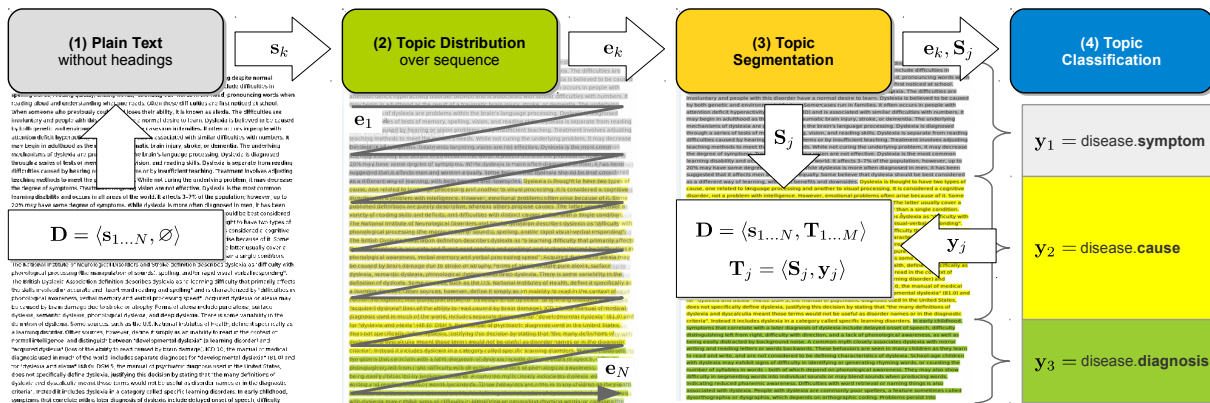
Figure 1: Overview of the WIKISECTION task: (1) The input is a plain text document $\mathbf{D}$ without structure information. (2) We assume the sentences $\mathbf{s}_{1\ldots N}$ contain a coherent sequence of local topics $\mathbf{e}_{1\ldots N}$. (3) The task is to segment the document into coherent sections $\mathbf{S}_{1\ldots M}$ and (4) to classify each section with a topic label $\mathbf{y}_{1\ldots M}$.

Eventually, we were inspired by *passage retrieval* (Liu and Croft, 2002) as an important downstream task for topic segmentation and classification. For example, Hewlett et al. (2016) proposed WikiReading, a QA task to retrieve values from sections of long documents. The objective of TREC Complex Answer Retrieval is to retrieve a ranking of relevant passages for a given outline of hierarchical sections (Nanni et al., 2017). Both tasks highly depend on a building block for local topic embeddings such as our proposed model.

3 Task Overview and Data set

We start with a definition of the WIKISECTION machine reading task shown in Figure 1. We take a document $\mathbf{D} = \langle \mathbf{S}, \mathbf{T} \rangle$ consisting of N consecutive sentences $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]$ and empty segmentation $\mathbf{T} = \varnothing$ as input. In our example, this is the plain text of a Wikipedia article (e.g., about Trichomoniasis[3]) without any section information. For each sentence $\mathbf{s}_k$, we assume a distribution of local topics $\mathbf{e}_k$ that gradually changes over the course of the document.

The task is to split $\mathbf{D}$ into a sequence of distinct topic sections $\mathbf{T} = [\mathbf{T}_1, \ldots, \mathbf{T}_M]$, so that each predicted section $\mathbf{T}_j = \langle \mathbf{S}_j, \mathbf{y}_j \rangle$ contains a sequence of coherent sentences $\mathbf{S}_j \subseteq \mathbf{S}$ and a topic label $\mathbf{y}_j$ that describes the common topic in these sentences. For the document

Data set	disease		city	
language	**en**	**de**	**en**	**de**
total docs	3.6k	2.3k	19.5k	12.5k
avg sents per doc	58.5	45.7	56.5	39.9
avg sects per doc	7.5	7.2	8.3	7.6
headings	8.5k	6.1k	23.0k	12.2k
topics	27	25	30	27
coverage	94.6%	89.5%	96.6%	96.1%

Table 1: Data set characteristics for *disease* (German: *Krankheit*) and *city* (German: *Stadt*). *Headings* denotes the number of distinct section and subsection headings among the documents. *Topics* stands for the number of topic labels after synset clustering. *Coverage* denotes the proportion of headings covered by topics; the remaining headings are labeled as other.

Trichomoniasis, the sequence of topic labels is $\mathbf{y}_{1\ldots M} = [$ symptom, cause, diagnosis, prevention, treatment, complication, epidemiology $]$.

3.1 WikiSection Data Set

For the evaluation of this task, we created WIKISECTION, a novel data set containing a gold standard of 38k full-text documents from English and German Wikipedia comprehensively annotated with sections and topic labels (see Table 1).

The documents originate from recent dumps in English[4] and German.[5] We filtered the collection

---

[3] https://en.wikipedia.org/w/index.php?title=Trichomoniasis&oldid=814235024.

[4] https://dumps.wikimedia.org/enwiki/20180101.

[5] https://dumps.wikimedia.org/dewiki/20180101.

using SPARQL queries against Wikidata (Tanon et al., 2016). We retrieved instances of Wikidata categories *disease* (Q12136) and their subcategories (e.g., `Trichomoniasis` or `Pertussis`) or *city* (Q515) (e.g., `London` or `Madrid`).

Our data set contains the article abstracts, plain text of the body, positions of all sections given by the Wikipedia editors with their original headings (e.g., `"Causes | Genetic sequence"`) and a normalized topic label (e.g., `disease.cause`). We randomized the order of documents and split them into 70% training, 10% validation, 20% test sets.

3.2 Preprocessing

To obtain plain document text, we used Wiki-extractor,[6] split the abstract sections and stripped all section headings and other structure tags except newline characters and lists.

Vocabulary Mismatch in Section Headings.
Table 2 shows examples of section headings from disease articles separated into head (most common), torso (frequently used), and tail (rare). Initially, we expected articles to share congruent structure in naming and order. Instead, we observe a high variance with 8.5k distinct headings in the diseases domain and over 23k for English cities. A closer inspection reveals that Wikipedia authors utilize headings at different granularity levels, frequently copy and paste from other articles, but also introduce synonyms or hyponyms, which leads to a *vocabulary mismatch problem* (Furnas et al., 1987). As a result, the distribution of headings is heavy-tailed across all articles. Roughly 1% of headings appear more than 25 times whereas the vast majority (88%) appear 1 or 2 times only.

3.3 Synset Clustering

In order to use Wikipedia headlines as a source for topic labels, we contribute a normalization method to reduce the high variance of headings to a few representative labels based on the clustering of BabelNet synsets (Navigli and Ponzetto, 2012).

We create a set $\mathcal{H}$ that contains all headings in the data set and use the BabelNet API to match[7] each heading $h \in \mathcal{H}$ to its corresponding

rank	heading h	label y	H	freq
0	Diagnosis	diagnosis	0.68	3,854
1	Treatment	treatment	0.69	3,501
2	Signs and Symptoms	symptom	0.68	2,452
	. . .			
21	Differential Diagnosis	diagnosis	0.23	236
22	Pathogenesis	mechanism	0.16	205
23	Medications	medication	0.14	186
	. . .			
8,494	Usher Syndrome Type IV	classification	0.00	1
8,495	False Melanose Lesions	other	0.00	1
8,496	Cognitive Therapy	treatment	0.00	1

Table 2: Frequency and entropy (H) of top-3 head and randomly selected torso and tail headings for category diseases in the English Wikipedia.

synsets $S_h \subset S$. For example, `"Cognitive behavioral therapy"` is assigned to synset `bn:03387773n`. Next, we insert all matched synsets into an undirected graph G with nodes $s \in S$ and edges e. We create edges between all synsets that match among each other with a lemma $h' \in \mathcal{H}$. Finally, we apply a community detection algorithm (Newman, 2006) on G to find dense clusters of synsets. We use these clusters as normalized topics and assign the sense with most outgoing edges as representative label, in our example e.g. `therapy`.

From this normalization step we obtain 598 synsets that we prune using the head/tail division rule $\text{count}(s) < \frac{1}{|S|} \sum_{s_i \in S} \text{count}(s_i)$ (Jiang, 2012). This method covers over 94.6% of all headings and yields 26 normalized labels and one `other` class in the English disease data set. Table 1 shows the corresponding numbers for the other data sets. We verify our normalization process by manual inspection of 400 randomly chosen heading–label assignments by two independent judges and report an accuracy of 97.2% with an average observed inter-annotator agreement of 96.0%.

4 SECTOR Model

We introduce SECTOR, a neural embedding model that predicts a latent topic distribution for every position in a document. Based on the task

---

[6] `http://attardi.github.io/wikiextractor/`.
[7] We match lemmas of main senses and compounds to synsets of type NOUN CONCEPT.
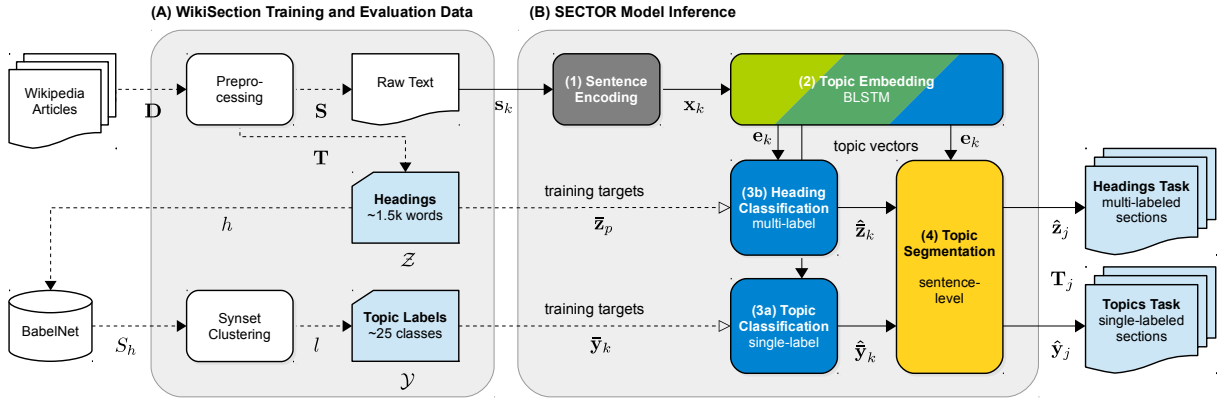
Figure 2: Training and inference phase of segmentation and topic classification (SECTOR). For training (A), we preprocess Wikipedia documents to supply a ground truth for segmentation $\mathbf{T}$, headings $\mathcal{Z}$ and topic labels $\mathcal{Y}$. During inference (B), we invoke SECTOR with unseen plain text to predict topic embeddings $\mathbf{e}_k$ on sentence level. The embeddings are used to segment the document and classify headings $\hat{\mathbf{z}}_j$ and normalized topic labels $\hat{\mathbf{y}}_j$.

described in Section 3, we aim to detect M sections $\mathbf{T}_{1\ldots M}$ in a document $\mathbf{D}$ and assign topic labels $\mathbf{y}_j = \text{topic}(\mathbf{S}_j)$, where $j = 1, \ldots, M$. Because we do not know the expected number of sections, we formulate the objective of our model on the sentence level and later segment based on the predictions. Therefore, we assign each sentence $\mathbf{s}_k$ a sentence topic label $\bar{\mathbf{y}}_k = \text{topic}(\mathbf{s}_k)$, where $k = 1, \ldots, N$. Thus, we aim to predict coherent sections with respect to document context:

$$p(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_N \mid \mathbf{D}) = \prod_{k=1}^{N} p(\bar{\mathbf{y}}_k \mid \mathbf{s}_1, \ldots, \mathbf{s}_N) \quad (1)$$

We approach two variations of this task: For WIKISECTION-topics, we choose a single topic label $\mathbf{y}_j \in \mathcal{Y}$ out of a small number of normalized topic labels. However, from this simplified classification task arises an entailment problem, because topics might be hierarchically structured. For example, a section with heading "Treatment | Gene Therapy" might describe genetics as a subtopic of treatment. Therefore, we also approach an extended task WIKISECTION-headings to capture ambiguity in a heading, We follow the CBOW approach (Mikolov et al., 2013) and assign all words in the heading $\mathbf{z}_j \subset \mathcal{Z}$ as multi-label bag over the original heading vocabulary. This turns our problem into a ranked retrieval task with a large number of ambiguous labels, similar to Prabhu and Varma (2014). It further eliminates the need for normalized topic labels. For both tasks, we aim to maximize the log likelihood

of model parameters Θ on section and sentence level:

$$\mathcal{L}(\Theta) = \sum_{j=1}^{M} \log p(\mathbf{y}_j \mid \mathbf{s}_1, \ldots, \mathbf{s}_N; \Theta)$$

$$\bar{\mathcal{L}}(\Theta) = \sum_{k=1}^{N} \log p(\bar{\mathbf{y}}_k \mid \mathbf{s}_1, \ldots, \mathbf{s}_N; \Theta) \quad (2)$$

Our SECTOR architecture consists of four stages, shown in Figure 2: sentence encoding, topic embedding, topic classification and topic segmentation. We now discuss each stage in more detail.

4.1 Sentence Encoding

The first stage of our SECTOR model transforms each sentence $\mathbf{s}_k$ from plain text into a fixed-size sentence vector $\mathbf{x}_k$ that serves as input into the neural network layers. Following Hill et al. (2016), word order is not critical for document-centric evaluation settings such as our WIKISECTION task. Therefore, we mainly focus on unsupervised compositional sentence representations.

Bag-of-Words Encoding. As a baseline, we compose sentence vectors using a weighted bag-of-words scheme. Let $\mathbb{I}(\mathbf{w}) \in \{0, 1\}^{|\mathcal{V}|}$ be the indicator vector, such that $\mathbb{I}(\mathbf{w})^{(i)} = 1$ iff $\mathbf{w}$ is the i-th word in the fixed vocabulary $\mathcal{V}$, and let $\text{tf-idf}(\mathbf{w})$ be the TF-IDF weight of $\mathbf{w}$ in the corpus. We define the sparse bag-of-words encoding $\mathbf{x}_{\text{bow}} \in \mathbb{R}^{|\mathcal{V}|}$ as follows:

$$\mathbf{x}_{\text{bow}}(\mathbf{s}) = \sum_{\mathbf{w} \in \mathbf{s}} \left(\text{tf-idf}(\mathbf{w}) \cdot \mathbb{I}(\mathbf{w}) \right) \quad (3)$$

Bloom Filter Embedding. For large $\mathcal{V}$ and long documents, input matrices grow too large to fit into GPU memory, especially with larger batch sizes. Therefore we apply a compression technique for sparse sentence vectors based on Bloom filters (Serrà and Karatzoglou, 2017). A Bloom filter projects every item of a set onto a bit array $\mathbb{A}(i) \in \{0,1\}^m$ using k independent hash functions. We use the sum of bit arrays per word as compressed Bloom embedding $\mathbf{x}_{\text{bloom}} \in \mathbb{N}^m$:

$$\mathbf{x}_{\text{bloom}}(\mathbf{s}) = \sum_{\mathbf{w} \in \mathbf{s}} \sum_{i=1}^{k} \mathbb{A}\big(\text{hash}_i(\mathbf{w})\big) \qquad (4)$$

We set parameters to $m = 4096$ and $k = 5$ to achieve a compression factor of 0.2, which showed good performance in the original paper.

Sentence Embeddings. We use the strategy of Arora et al. (2017) to generate a distributional sentence representation based on pre-trained word2vec embeddings (Mikolov et al., 2013). This method composes a sentence vector $v_{\text{emb}} \in \mathbb{R}^d$ for all sentences using a probability-weighted sum of word embeddings $v_{\mathbf{w}} \in \mathbb{R}^d$ with $\alpha = 10^{-4}$, and subtracts the first principal component u of the embedding matrix $[\, v_{\mathbf{s}} : \mathbf{s} \in \mathbf{S} \,]$:

$$v_{\mathbf{s}} = \frac{1}{|\mathbf{S}|} \sum_{\mathbf{w} \in \mathbf{s}} \Big(\frac{\alpha}{\alpha + p(\mathbf{w})} \, v_{\mathbf{w}} \Big)$$
$$\mathbf{x}_{\text{emb}}(\mathbf{s}) = v_{\mathbf{s}} - u u^T v_{\mathbf{s}} \qquad (5)$$

4.2 Topic Embedding

We model the second stage in our architecture to produce a dense distributional representation of latent topics for each sentence in the document. We use two layers of LSTM (Hochreiter and Schmidhuber, 1997) with forget gates (Gers et al., 2000) connected to read the document in the forward and backward direction (Graves, 2012). We feed the LSTM outputs to a ''bottleneck'' layer with tanh activation as topic embedding. Figure 3 shows these layers in context of the complete architecture. We can see that context from left $(k - 1)$ and right $(k + 1)$ affects forward and backward layers independently. It is therefore important to separate these weights in the embedding layer to precisely capture the difference between sentences at section boundaries. We modify our objective given in Equation (2) accordingly with long-range depen-
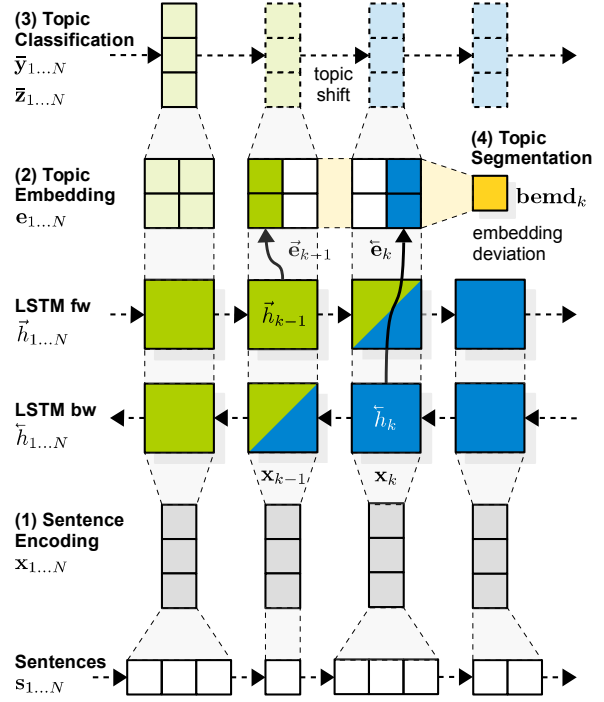

Figure 3: Neural network architecture SECTOR. The recurrent model consists of stacked LSTM, embedding and output layers that are optimized on document level and later accessed during inference in stages 1–4.

dencies from forward and backward layers of the LSTM:

$$\mathcal{L}(\Theta) = \sum_{k=1}^{N} \big(\log p(\bar{\mathbf{y}}_k \mid \mathbf{x}_{1\dots k-1}; \vec{\Theta}, \Theta') \\ + \log p(\bar{\mathbf{y}}_k \mid \mathbf{x}_{k+1\dots N}; \overleftarrow{\Theta}, \Theta')) \qquad (6)$$

Note that we separate network parameters $\vec{\Theta}$ and $\overleftarrow{\Theta}$ for forward and backward directions of the LSTM, and tie the remaining parameters Θ' for the embedding and output layers. This strategy couples the optimization of both directions into the same vector space without the need for an additional loss function. The embeddings $\mathbf{e}_{1\dots N}$ are calculated from the context-adjusted hidden states h'_k of the LSTM cells (here simplified as f_{LSTM}) through the bottleneck layer:

$$\vec{h}_k = f_{\text{LSTM}}(\mathbf{x}_k, \vec{h}'_{k-1}, \vec{\Theta})$$
$$\overleftarrow{h}_k = f_{\text{LSTM}}(\mathbf{x}_k, \overleftarrow{h}'_{k+1}, \overleftarrow{\Theta})$$
$$\vec{\mathbf{e}}_k = \tanh(W_{eh}\vec{h}_k + b_e) \qquad (7)$$
$$\overleftarrow{\mathbf{e}}_k = \tanh(W_{eh}\overleftarrow{h}_k + b_e)$$

Now, a simple concatenation of the embeddings $\mathbf{e}_k = \vec{\mathbf{e}}_k \oplus \overleftarrow{\mathbf{e}}_k$ can be used as topic vector by downstream applications.

4.3 Topic Classification

The third stage in our architecture is the output layer that decodes the class labels. To learn model parameters Θ required by the embedding, we need to optimize the full model for a training target. For the WIKISECTION-topics task, we use a simple one-hot encoding $\bar{\mathbf{y}} \in \{0,1\}^{|\mathcal{Y}|}$ of the topic labels constructed in Section 3.3 with a softmax activation output layer. For the WIKISECTION-headings task, we encode each heading as lower-case bag-of-words vector $\bar{\mathbf{z}} \in \{0,1\}^{|\mathcal{Z}|}$, such that $\bar{\mathbf{z}}^{(i)} = 1$ iff the i-th word in $\mathcal{Z}$ is contained in the heading, for example, $\bar{\mathbf{z}}_k \hat{=} \{\texttt{gene}, \texttt{therapy}, \texttt{treatment}\}$. We then use a sigmoid activation function:

$$\hat{\bar{\mathbf{y}}}_k = \text{softmax}(W_{ye}\vec{\mathbf{e}}_k + W_{ye}\overleftarrow{\mathbf{e}}_k + b_y)$$
$$\hat{\bar{\mathbf{z}}}_k = \text{sigmoid}(W_{ze}\vec{\mathbf{e}}_k + W_{ze}\overleftarrow{\mathbf{e}}_k + b_z) \quad (8)$$

Ranking Loss for Multi-Label Optimization. The multi-label objective is to maximize the likelihood of every word that appears in a heading:

$$\mathcal{L}(\Theta) = \sum_{k=1}^{N} \sum_{i=1}^{|\mathcal{Z}|} \log p(\bar{\mathbf{z}}_k^{(i)} \mid \mathbf{x}_{1\ldots N}; \Theta) \quad (9)$$

For training this model, we use a variation of the logistic pairwise ranking loss function proposed by dos Santos et al. (2015). It learns to maximize the distance between positive and negative labels:

$$L = \log\big(1 + \exp(\gamma(m^+ - \text{score}^+(\mathbf{x})))\big) + \log\big(1 + \exp(\gamma(m^- + \text{score}^-(\mathbf{x})))\big) \quad (10)$$

We calculate the positive term of the loss by taking all scores of correct labels y^+ into account. We average over all correct scores to avoid a too-strong positive push on the energy surface of the loss function (LeCun et al., 2006). For the negative term, we only take the most offending example y^- among all incorrect class labels.

$$\text{score}^+(\mathbf{x}) = \frac{1}{|y^+|} \sum_{y \in y^+} s_\theta(\mathbf{x})^{(y)}$$
$$\text{score}^-(\mathbf{x}) = \arg\max_{y \in y^-} s_\theta(\mathbf{x})^{(y)} \quad (11)$$

Here, $s_\theta(\mathbf{x})^{(y)}$ denotes the score of label y for input $\mathbf{x}$. We follow the authors and set scaling factor $\gamma = 2$, margins $m^+ = 2.5$, and $m^- = 0.5$.

4.4 Topic Segmentation

In the final stage, we leverage the information encoded in the topic embedding and output layers to segment the document and classify each section.

Baseline Segmentation Methods. As a simple baseline method, we use prior information from the text and split sections at *newline* characters (NL). Additionally, we merge two adjacent sections if they are assigned the same topic label after classification. If there is no newline information available in the text, we use a *maximum label* (max) approach: We first split sections at every sentence break (i.e., $\mathbf{S}_j = \mathbf{s}_k; j = k = 1, \ldots, N$) and then merge all sections that share at least one label in the top-2 predictions.

Using Deviation of Topic Embeddings for Segmentation. All information required to classify each sentence in a document is contained in our dense topic embedding matrix $E = [\mathbf{e}_1, \ldots, \mathbf{e}_N]$. We are now interested in the vector space movement of this embedding over the sequence of sentences. Therefore, we apply a number of transformations adapted from Laplacian-of-Gaussian edge detection on images (Ziou and Tabbone, 1998) to obtain the magnitude of *embedding deviation* (emd) per sentence. First, we reduce the dimensionality of E to D dimensions using PCA, that is, we solve $E = U\Sigma W^T$ using singular value decomposition and then project E on the D principal components $E_D = EW_D$. Next, we apply Gaussian smoothing to obtain a smoothed matrix E'_D by convolution with a Gaussian kernel with variance σ^2. From the reduced and smoothed embedding vectors $\mathbf{e}'_{1\ldots N}$ we construct a sequence of deviations $\mathbf{d}_{1\ldots N}$ by calculating the stepwise difference using cosine distance:

$$\mathbf{d}_k = \cos(\mathbf{e}'_{k-1}, \mathbf{e}'_k) = \frac{\mathbf{e}'_{k-1} \cdot \mathbf{e}'_k}{\| \mathbf{e}'_{k-1} \| \| \mathbf{e}'_k \|} \quad (12)$$

Finally we apply the sequence $\mathbf{d}_{1\ldots N}$ with parameters $D = 16$ and $\sigma = 2.5$ to locate the spots of fastest movement (see Figure 4), i.e. all k where $\mathbf{d}_{k-1} < \mathbf{d}_k > \mathbf{d}_{k+1}; k = 1 \ldots N$ in our discrete case. We use these positions to start a new section.

Improving Edge Detection with Bidirectional Layers. We adopt the approach of Sehikh et al. (2017), who examine the difference between forward and backward layer of an LSTM for segmentation. However, our approach focuses on the difference of left and right topic context over time steps k, which allows for a sharper distinction between sections. Here, we obtain two smoothed embeddings $\vec{\mathbf{e}}'$ and $\overleftarrow{\mathbf{e}}'$ and define the *bidirectional*
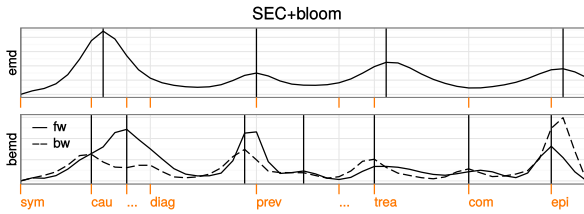
Figure 4: Embedding deviations $\mathbf{emd}_k$ and $\mathbf{bemd}_k$ of the smoothed SECTOR topic embeddings for example document `Trichomoniasis`. The plot shows the first derivative of vector movement over sentences $k = 1, \ldots N$ from left to right. Predicted segmentation is shown as black lines, the axis labels indicate ground truth segmentation.

embedding deviation (bemd) as geometric mean of the forward and backward difference:

$$\mathbf{d}'_k = \sqrt{\cos(\vec{\mathbf{e}}'_{k-1}, \vec{\mathbf{e}}'_k) \cdot \cos(\overleftarrow{\mathbf{e}}'_k, \overleftarrow{\mathbf{e}}'_{k+1})} \quad (13)$$

After segmentation, we assign each segment the mean class distribution of all contained sentences:

$$\hat{\mathbf{y}}_j = \frac{1}{|\mathbf{S}_j|} \sum_{\mathbf{s}_i \in \mathbf{S}_j} \hat{\bar{\mathbf{y}}}_i \quad (14)$$

Finally, we show in the evaluation that our SECTOR model, which was optimized for sentences $\bar{\mathbf{y}}_k$, can be applied to the WIKISECTION task to predict coherently labeled sections $\mathbf{T}_j = \langle \mathbf{S}_j, \hat{\mathbf{y}}_j \rangle$.

5 Evaluation

We conduct three experiments to evaluate the segmentation and classification task introduced in Section 3. The WIKISECTION-topics experiment constitutes segmentation and classification of each section with a single topic label out of a small number of clean labels (25–30 topics). The WIKISECTION-headings experiment extends the classification task to multi-label per section with a larger target vocabulary (1.0k–2.8k words). This is important, because often there are no clean topic labels available for training or evaluation. Finally, we conduct a third experiment to see how SECTOR performs across existing segmentation data sets.

Evaluation Data Sets. For the first two experiments we use the WIKISECTION data sets introduced in Section 3.1, which contain documents about diseases and cities in both English and German. The subsections are retained with full granularity. For the third experiment, text segmentation results are often reported on artificial data sets (Choi, 2000). It was shown that this

scenario is hardly applicable to topic-based segmentation (Koshorek et al., 2018), so we restrict our evaluation to real-world data sets that are publicly available. The *Wiki-727k* data set by Koshorek et al. (2018) contains Wikipedia articles with a broad range of topics and their top-level sections. However, it is too large to compare exhaustively, so we use the smaller *Wiki-50* subset. We further use the *Cities* and *Elements* data sets introduced by Chen et al. (2009), which also provide headings. These sets are typically used for word-level segmentation, so they don't contain any punctuation and are lowercased. Finally, we use the *Clinical Textbook* chapters introduced by Eisenstein and Barzilay (2008), which do not supply headings.

Text Segmentation Models. We compare SECTOR to common text segmentation methods as baseline, *C99* (Choi, 2000) and *TopicTiling* (Riedl and Biemann, 2012) and the state-of-the-art *TextSeg* segmenter (Koshorek et al., 2018). In the third experiment we report numbers for *BayesSeg* (Eisenstein and Barzilay, 2008) (configured to predict with unknown number of segments) and *GraphSeg* (Glavaš et al., 2016).

Classification Models. We compare SECTOR to existing models for single and multi-label sentence classification. Because we are not aware of any existing method for combined segmentation and classification, we first compare all methods using given prior segmentation from newlines in the text (*NL*) and then additionally apply our own segmentation strategies for plain text input: maximum label (*max*), embedding deviation (*emd*) and bidirectional embedding deviation (*bemd*).

For the experiments, we train a Paragraph Vectors (*PV*) model (Le and Mikolov, 2014) using all sections of the training sets. We utilize this model for single-label topic classification (depicted as PV>T) by assigning the given topic labels as paragraph IDs. Multi-label classification is not possible with this model. We use the paragraph embedding for our own segmentation strategies. We set the layer size to 256, window size to 7, and trained for 10 epochs using a batch size of 512 sentences and a learning rate of 0.025. We further use an implementation of *CNN* (Kim, 2014) with our pre-trained word vectors as input for single-label topics (CNN>T) and multi-label

headings (CNN>H). We configured the models using the hyperparameters given in the paper and trained the model using a batch size of 256 sentences for 20 epochs with learning rate 0.01.

SECTOR Configurations. We evaluate the various configurations of our model discussed in prior sections. SEC>T depicts the single-label topic classification model which uses a softmax activation output layer, SEC>H is the multi-label variant with a larger output and sigmoid activations. Other options are: bag-of-words sentence encoding (+*bow*), Bloom filter encoding (+*bloom*) and sentence embeddings (+*emb*); multi-class cross-entropy loss (as default) and ranking loss (+*rank*).

We have chosen network hyperparameters using grid search on the en_disease validation set and keep them fixed over all evaluation runs. For all configurations, we set LSTM layer size to 256, topic embeddings dimension to 128. Models are trained on the complete train splits with a batch size of 16 documents (reduced to 8 for bag-of-words), 0.01 learning rate, 0.5 dropout, and ADAM optimization. We used early stopping after 10 epochs without MAP improvement on the validation data sets. We pretrained word embeddings with 256 dimensions for the specific tasks using word2vec on lowercase English and German Wikipedia documents using a window size of 7. All tests are implemented in Deeplearning4j and run on a Tesla P100 GPU with 16GB memory. Training a SEC+bloom model on en_city takes roughly 5 hours, inference on CPU takes on average 0.36 seconds per document. In addition, we trained a SEC>H@fullwiki model with raw headings from a complete English Wikipedia dump,[8] and use this model for cross-data set evaluation.

Quality Measures. We measure text segmentation at sentence level using the *probabilistic P_k error score* (Beeferman et al., 1999), which calculates the probability of a false boundary in a window of size k, lower numbers mean better segmentation. As relevant section boundaries we consider all section breaks where the topic label changes. We set k to half of the average segment length. We measure classification performance on section level by comparing the topic labels of all ground truth sections with predicted sections. We

---

select the pairs by matching their positions using maximum boundary overlap. We report *micro-averaged F_1* score for single-label or *Precision@1* for multi-label classification. Additionally, we measure *Mean Average Precision* (MAP), which evaluates the average fraction of true labels ranked above a particular label (Tsoumakas et al., 2009).

5.1 Results

Table 3 shows the evaluation results of the WIKISECTION-topics single-label classification task, Table 4 contains the corresponding numbers for multi-label classification. Table 5 shows results for topic segmentation across different data sets.

SECTOR Outperforms Existing Classifiers. With our given segmentation baseline (NL), the best sentence classification model CNN achieves 52.1% F_1 averaged over all data sets. SECTOR improves this score significantly by 12.4 points. Furthermore, in the setting with plain text input, SECTOR improves the CNN score by 18.8 points using identical baseline segmentation. Our model finally reaches an average of 61.8% F_1 on the classification task using sentence embeddings and bidirectional segmentation. This is a total improvement of 27.8 points over the CNN model.

Topic Embeddings Improve Segmentation. SECTOR outperforms C99 and TopicTiling significantly by 16.4 and 18.8 points P_k, respectively, on average. Compared to the maximum label baseline, our model gains 3.1 points by using the bidirectional embedding deviation and 1.0 points using sentence embeddings. Overall, SECTOR misses only 4.2 points P_k and 2.6 points F_1 compared with the experiments with prior newline segmentation. The third experiments reveals that our segmentation method in isolation almost reaches state-of-the-art on existing data sets and beats the unsupervised baselines, but lacks performance on cross-data set evaluation.

Bloom Filters on Par with Word Embeddings. Bloom filter encoding achieves high scores among all data sets and outperforms our bag-of-words baseline, possibly because of larger training batch sizes and reduced model parameters. Surprisingly, word embeddings did not improve the model significantly. On average, German models gained 0.7 points F_1 and English models declined by 0.4 points compared with Bloom filters. However,

---

[8]Excluding all documents contained in the test sets.

WikiSection-topics single-label classification		en_disease 27 topics			de_disease 25 topics			en_city 30 topics			de_city 27 topics		
model configuration	segm.	P_k	F_1	MAP	P_k	F_1	MAP	P_k	F_1	MAP	P_k	F_1	MAP
Classification with newline prior segmentation													
PV>T*	NL	35.6	31.7	47.2	36.0	29.6	44.5	22.5	52.9	63.9	27.2	42.9	55.5
CNN>T*	NL	31.5	40.4	55.6	31.6	38.1	53.7	13.2	66.3	76.1	13.7	63.4	75.0
SEC>T+bow	NL	25.8	54.7	68.4	25.0	**52.7**	**66.9**	21.0	43.7	55.3	20.2	40.5	52.2
SEC>T+bloom	NL	22.7	**59.3**	**71.9**	27.9	50.2	65.5	**9.8**	**74.9**	**82.6**	11.7	73.1	81.5
SEC>T+emb*	NL	**22.5**	58.7	71.4	**23.6**	50.9	66.8	10.7	74.1	82.2	**10.7**	**74.0**	**83.0**
Classification and segmentation on plain text													
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
PV>T*	max	43.6	20.4	36.5	44.3	19.3	34.6	31.1	28.1	43.1	36.4	20.2	35.5
PV>T*	emd	39.2	32.9	49.3	37.4	32.9	48.7	24.9	53.1	65.1	32.9	40.6	55.0
CNN>T*	max	40.1	26.9	45.0	40.7	25.2	43.8	21.9	42.1	58.7	21.4	42.1	59.5
SEC>T+bow	max	30.1	40.9	58.5	32.1	38.9	56.8	24.5	28.4	43.5	28.0	26.8	42.6
SEC>T+bloom	max	27.9	49.6	64.7	35.3	39.5	57.3	**12.7**	63.3	74.3	26.2	58.9	71.6
SEC>T+bloom	emd	29.7	52.8	67.5	35.3	44.8	61.6	16.4	65.8	77.3	26.0	65.5	76.7
SEC>T+bloom	bemd	26.8	56.6	**70.1**	31.7	47.8	63.7	14.4	**71.6**	80.9	16.8	70.8	80.1
SEC>T+bloom+rank*	bemd	26.8	**56.7**	68.8	33.1	44.0	58.5	15.7	71.1	79.1	18.0	66.8	76.1
SEC>T+emb*	bemd	26.3	55.8	69.4	**27.5**	**48.9**	**65.1**	15.5	71.6	**81.0**	16.2	71.0	**81.1**

Table 3: Results for topic segmentation and single-label classification on four WIKISECTION data sets. $n = 718$ / 464 / 3,907 / 2,507 documents. Numbers are given as P_k on sentence level, micro-averaged F_1 and MAP at segment-level. For methods without segmentation, we used newlines as segment boundaries (NL) and merged sections of same classes after prediction. Models marked with * are based on pre-trained distributional embeddings.

WikiSection-headings multi-label classification		en_disease 1.5k topics			de_disease 1.0k topics			en_city 2.8k topics			de_city 1.1k topics		
model configuration	segm.	P_k	P@1	MAP	P_k	P@1	MAP	P_k	P@1	MAP	P_k	P@1	MAP
CNN>H*	max	40.9	36.7	31.5	41.3	14.1	21.1	36.9	43.3	46.7	42.2	40.9	46.5
SEC>H+bloom	bemd	35.4	35.8	38.2	36.9	31.7	37.8	20.0	65.2	62.0	23.4	49.8	53.4
SEC>H+bloom+rank	bemd	40.2	47.8	49.0	42.8	28.4	33.2	41.9	66.8	59.0	34.9	59.6	54.6
SEC>H+emb*	bemd	30.7	**50.5**	**57.3**	**32.9**	26.6	**36.7**	17.9	**72.3**	**71.1**	19.3	68.4	**70.2**
SEC>H+emb+rank*	bemd	**30.5**	47.6	48.9	42.9	**32.0**	36.4	**16.1**	65.8	59.0	**18.3**	**69.2**	58.9
SEC>H+emb@fullwiki*	bemd	42.4	9.7	17.9	42.7	(0.0)	(0.0)	20.3	59.4	50.4	38.5	(0.0)	(0.1)

Table 4: Results for segmentation and multi-label classification trained with raw Wikipedia headings. Here, the task is to segment the document and predict multi-word topics from a large ambiguous target vocabulary.

model training and inference using pre-trained embeddings is faster by an average factor of 3.2.

Topic Embeddings Perform Well on Noisy Data. In the multi-label setting with unprocessed Wikipedia headings, classification precision of SECTOR reaches up to 72.3% P@1 for 2.8k labels. This score is in average 9.5 points lower compared to the models trained on the small number of 25–30 normalized labels. Furthermore, segmentation performance only misses 3.8 points P_k compared with the topics task. Ranking loss could not

improve our models significantly, but achieved better segmentation scores on the headings task. Finally, the cross-domain English *fullwiki* model performs only on baseline level for segmentation, but still achieves better classification performance than CNN on the English cities data set.

5.2 Discussion and Model Insights

Figure 5 shows classification and segmentation of our SECTOR model compared to the PV baseline.

Segmentation	Wiki-50		Cities		Elements		Clinical
and multi-label classification	P_k	MAP	P_k	MAP	P_k	MAP	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>C+emb@en_disease	—	—	—	—	45.1	n/a	35.6
SEC>H+emb@en_city	30.0	31.4	28.2	**56.5**	41.0	7.9	—
SEC>C+emb@en_city	31.3	n/a	22.9	n/a	48.8	n/a	—
SEC>H+emb@cities	33.3	15.3	21.4*	52.3*	39.2	12.1	37.7
SEC>H+emb@fullwiki	28.6*	**32.6***	33.4	40.5	42.8	**14.4**	36.9

Table 5: Results for cross-data set evaluation on existing data sets. Numbers marked with * are generated by models trained specifically for this data set. A value of 'n/a' indicates that a model is not applicable to this problem.
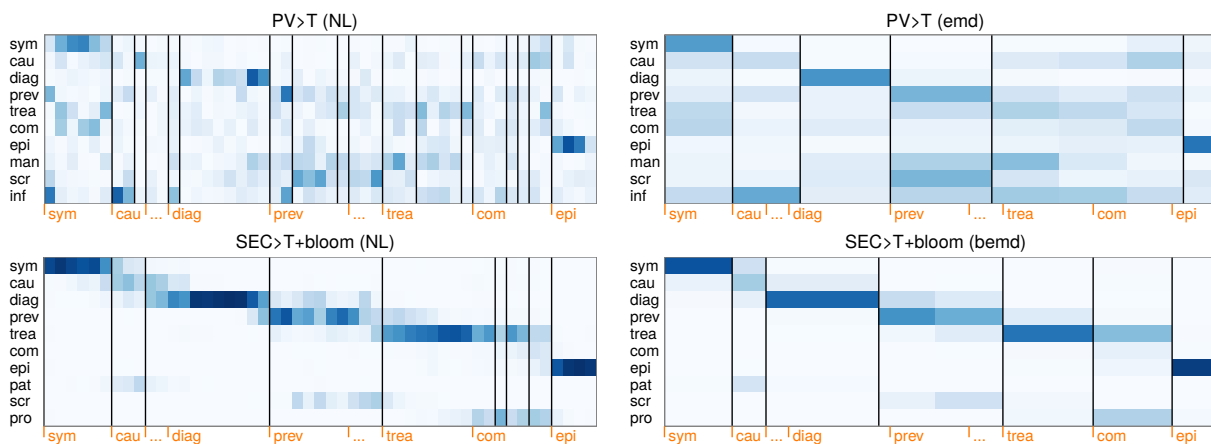

Figure 5: Heatmaps of predicted topic labels $\hat{y}_k$ for document `Trichomoniasis` from PV and SECTOR models with newline and embedding segmentation. Shading denotes probability for 10 out of 27 selected topic classes on Y axis, with sentences from left to right. Segmentation is shown as black lines, X axis shows expected gold labels. Note that segments with same class assignments are merged in both predictions and gold standard ('...').

SECTOR Captures Latent Topics from Context. We clearly see from NL predictions (left side of Figure 5) that SECTOR produces coherent results with sentence granularity, with topics emerging and disappearing over the course of a document. In contrast, PV predictions are scattered across the document. Both models successfully classify first (`symptoms`) and last sections (`epidemiology`). However, only SECTOR can capture `diagnosis`, `prevention`, and `treatment`. Furthermore, we observe additional `screening` predictions in the center of the document. This section is actually labeled `"Prevention | Screening"` in the source document, which explains this overlap.

Furthermore, we observe low confidence in the second section labeled `cause`. Our multi-class model predicts for this section {`diagnosis`,

`cause`, `genetics`}. The ground truth heading for this section is `"Causes | Genetic sequence,"` but even for a human reader this assignment is not clear. This shows that the multi-label approach fills an important gap and can even serve as an indicator for low-quality article structure.

Finally, both models fail to segment the `complication` section near the end, because it consists of an enumeration. The embedding deviation segmentation strategy (right side of Figure 5) completely solves this issue for both models. Our SECTOR model is giving nearly perfect segmentation using the bidirectional strategy, it only misses the discussed part of `cause` and is off by one sentence for the start of `prevention`. Furthermore, averaging over sentence-level predictions reveals clearly distinguishable section class labels.

6 Conclusions and Future Work

We presented SECTOR, a novel model for coherent text segmentation and classification based on latent topics. We further contributed WIKISECTION, a collection of four large data sets in English and German for this task. Our end-to-end method builds on a neural topic embedding which is trained using Wikipedia headings to optimize a bidirectional LSTM classifier. We showed that our best performing model is based on sparse word features with Bloom filter encoding and significantly improves classification precision for 25–30 topics on comprehensive documents by up to 29.5 points F_1 compared with state-of-the-art sentence classifiers with baseline segmentation. We used the bidirectional deviation in our topic embedding to segment a document into coherent sections without additional training. Finally, our experiments showed that extending the task to multi-label classification of 2.8k ambiguous topic words still produces coherent results with 71.1% average precision.

We see an exciting future application of SECTOR as a building block to extract and retrieve topical passages from unlabeled corpora, such as medical research articles or technical papers. One possible task is WikiPassageQA (Cohen et al., 2018), a benchmark to retrieve passages as answers to non-factoid questions from long articles.

Acknowledgments

References

Shashank Agarwal and Hong Yu. 2009. Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, 25(23):3174–3180.

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128.

Alexander A. Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *CoRR*, cs.CL/1503.05543v1.

James Allan. 2002, Introduction to topic detection and tracking. In *Topic Detection and Tracking*, pages 1–16. Springer.

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. 2008. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Eighth IEEE International Conference on Data Mining*, pages 3–12. IEEE.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR 2017: 5th International Conference on Learning Representations*.

M. Bayomi, K. Levacher, M. R. Ghorab, and S. Lawless. 2015. OntoSeg: A novel approach to text segmentation using ontological similarity. In *2015 International Conference on Data Mining Workshop*, pages 1274–1283. IEEE.

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

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2016. Automatic labelling of topics with neural embeddings. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 953–963.

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

Harr Chen, S. R. K. 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. ACL.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 26–33. ACL.

Philipp Cimiano, Antje Schultz, Sergej Sizov, Philipp Sorg, and Steffen Staab. 2009. Explicit versus latent concept models for cross-language information retrieval. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, volume 9, pages 1513–1518.

Daniel Cohen, Liu Yang, and W. Bruce Croft. 2018. WikiPassageQA: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1165–1168. ACM.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 1, pages 1107–1116. ACL.

Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. 2007. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, volume 7, pages 1334–1340.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2017. TopicRNN: A recurrent neural network with long-range semantic dependency. In *ICLR 2017: 5th International Conference on Learning Representations*.

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 and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 334–343. ACL.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1606–1611.

Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Compututation*, 12(10):2451–2471.

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. ACL.

Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385, Springer, Berlin Heidelberg.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1535–1545. ACL.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

Julia Hirschberg and Christopher D. Manning. 2015. Advances in natural language processing. *Science*, 349(6245):261–266.

Hoa T. Le, Christophe Cerisara, and Alexandre Denis. 2018. Do convolutional networks need to be deep for text classification? In *Association for the Advancement of Artificial Intelligence 2018 Workshop on Affective Content Analysis*, pages 29–36.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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.

Minwoo Jeong and Ivan Titov. 2010. Multi-document topic segmentation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1119–1128. ACM.

Bin Jiang. 2012. Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution. *The Professional Geographer*, 65(3):482–494.

Hubert Jin, Rich Schwartz, Sreenivasa Sista, and Frederick Walls. 1999. Topic tracking for radio, TV broadcast and newswire. In *Proceedings of the DARPA Broadcast News Workshop*, pages 199–204. Morgan Kaufmann.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142. Springer.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

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)*, volume 2, pages 469–473.

Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 297–304. ACM.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1188–1196.

Yann LeCun, Sumit Chopra, Raia Hadsell, M. Ranzato, and F. Huang. 2006, A tutorial on energy-based learning. In *Predicting Structured Data*, volume 1, MIT.

Kalev Leetaru and Philip A. Schrodt. 2013. GDELT: Global data on events, location, and tone, 1979–2012. In *ISA Annual Convention*, volume 2, pages 1–49. Citeseer.

Jialu Liu, Xiang Ren, Jingbo Shang, Taylor Cassidy, Clare R. Voss, and Jiawei Han. 2016. Representing documents via latent keyphrase inference. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1057–1067.

Xiaoyong Liu and W. Bruce Croft. 2002. Passage retrieval based on language models. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 375–382. ACM.

Sean MacAvaney, Andrew Yates, Arman Cohan, Luca Soldaini, Kai Hui, Nazli Goharian, and Ophir Frieder. 2018. Characterizing question facets for complex answer retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1205–1208. ACM Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, cs.CL/1301.3781v3.

Marwa Naili, Anja Habacha Chaïbi, and Henda Hajjami Ben Ghézala. 2017. Comparative study of word embedding methods in topic segmentation. In *Proceedings of the 21st International Conference Knowledge-Based and*

Intelligent Information & Engineering Systems, volume 112 of *Procedia Computer Science*, pages 340–349. Elsevier.

Federico Nanni, Bhaskar Mitra, Matt Magnusson, and Laura Dietz. 2017. Benchmark for complex answer retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 293–296. ACM.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Mark E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104.

Tiziano Piccardi, Michele Catasta, Leila Zia, and Robert West. 2018. Structuring Wikipedia articles with section recommendations. In *Proceedings of the 41th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–674. ACM.

Yashoteja Prabhu and Manik Varma. 2014. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 263–272. ACM.

Martin Riedl and Chris Biemann. 2012. Topic-Tiling: A text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42. ACL.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 626–634. ACL.

Imran Sehikh, Dominique Fohr, and Irina Illina. 2017. Topic segmentation in ASR transcripts using bidirectional RNNs for change detection. In *Automatic Speech Recognition and Understanding Workshop*, pages 512–518. IEEE.

Joan Serrà and Alexandros Karatzoglou. 2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 279–287. ACM.

Thomas Pellissier Tanon, Denny Vrandecic, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From Freebase to Wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428.

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 Eighth International Conference on Language Resources and Evaluation*, pages 2001–2008.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R. Alvers, and Matthias Zschunke. 2012. BioASQ: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI Technical Report FS-12-05 Information Retrieval and Knowledge Discovery in Biomedical Text*, pages 92–98.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2009. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 499–506. ACL.

Liang Wang, Sujian Li, Yajuan Lyu, and Houfeng Wang. 2017. Learning to rank semantic coherence for topic segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1340–1344. ACL.

Jui-Feng Yeh, Yi-Shan Tan, and Chen-Hsien Lee. 2016. Topic detection and tracking for conversational content by using conceptual dynamic latent Dirichlet allocation. *Neurocomputing*, 216:310–318.

Djemel Ziou and Salvatore Tabbone. 1998. Edge detection techniques – An overview. *Pattern Recognition and Image Analysis*, 8:537–559.