Tipster: A Topic-Guided Language Model for Topic-Aware Text Segmentation

Zheng Gong, Shiwei Tong, Han Wu, Qi Liu(✉), Hanqing Tao, Wei Huang, and Runlong Yu

Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei 230026, China
{gz70229,tongsw,wuhanhan,hqtao,ustc0411,yrunl}@mail.ustc.edu.cn, qiliuql@ustc.edu.cn

Abstract. The accurate segmentation and structural topics of plain documents not only meet people's reading habit, but also facilitate various downstream tasks. Recently, some works have consistently given positive hints that text segmentation and segment topic labeling could be regarded as a mutual task, and cooperating with word distributions has the potential to model latent topics in a certain document better. To this end, we present a novel model namely *Tipster* to solve text segmentation and segment topic labeling collaboratively. We first utilize a neural topic model to infer latent topic distributions of sentences considering word distributions. Then, our model divides the document into topically coherent segments based on the topic-guided contextual sentence representations of the pre-trained language model and assign relevant topic labels to each segment. Finally, we conduct extensive experiments which demonstrate that Tipster achieves the state-of-the-art performance in both text segmentation and segment topic labeling tasks.

Keywords: Text segmentation · Neural topic model · Language model

1 Introduction

Text segmentation and segment topic labeling tasks are two coupled tasks denoted henceforth **topic-aware text segmentation** (**TATS**) task, which aim to provide accurate text segmentation and structural topics of unlabeled documents. Figure 1 describes a toy example of TATS about *Paris* city. Based on topical coherence, the sentences in the document (*i.e.*, s1, s2, s3, s4) are divided into three segments, which portray various topics of *Paris*, *i.e.*, History, Geography and Culture. Accurate segmentation and structural topics can not only help understand the unlabeled documents better, but also be applied to many downstream tasks, such as passage retrieval and intent detection.

The vast majority of studies on TATS concentrate on supervised methods. SECTOR [1] and S-LSTM [2] exploit pre-trained word embeddings to predict segment boundary and assign related topics. Besides, some BERT-based

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 213–221, 2022.
https://doi.org/10.1007/978-3-031-00129-1_14

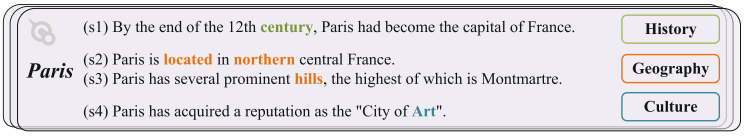

Fig. 1. A raw document includes detailed description of *Paris* covering multiple topics (*e.g.,* History). The colored words are keywords related to the corresponding topic. (Color figure online)

works [10] recently have achieved a high performance on the single text segmentation task by directly representing sentence with pre-trained BERT embedding [4]. However, the native sentence representations generated from BERT are proved to collapse into a small space and produce high similarity between most sentences [9], which hinder the sentence representations of diverse topics.

Actually, *word distributions can be regarded as evident representations of topics*. For instance, as shown in Fig. 1, the word "*century*" appears more frequently in the text related to history topic (in Sentence 1), whereas "*located*" and "*northern*" appear more frequently in geography topic (in Sentence 2). Therefore, it comes to the conclusion that word distributions vary in different topics. We could utilize the word distributions to pay more attention to the words relevant to topics, leading to the distinguishable sentence representation of topics.

However, there are many technical challenges in aligning the pre-trained sentence representation and topic-word distributions with the TATS. First, the mainstream methods of modeling word distributions adopt probabilistic topic models [13], which have a slow speed in estimating parameters. However, due to the complexity of our task, traditional inference methods will be limited by the cost of computation. Second, since the pre-trained sentence representations have some limitations [9] in capturing semantic changes in sentences, how to allocate rational attention on each word has not been explored much.

With this in mind, we bring in a neural variational topic model to capture the relation between distributions of words and topics and propose the **T**opic-gu**i**ded **p**re-trained **s**en**t**enc**e** **r**epresentation of language models (**Tipster**) for TATS. Specifically, our model has a three-stage process for this task. First, we infer the contextual representation of words in the sentence with a pre-trained language model. Second, with the help of neural topic model, we exploit the word distributions to discover informative words in the sentence and obtain topic-guided sentence representations via rational attention weights. Third, we capture the change of semantics via sentence representations of the document and predict topic labels of each segment. Our extensive experimental results show that Tipster achieves the state-of-the-art (SOTA) performance on the TATS. We also show that these improvements are in line with out-of-domain datasets.

2 Problem Formulation

In this section, we formulate the TATS formally. Given the pre-defined topic categories C and a document containing N consecutive sentences $S = [s_1, ..., s_N]$, the goals of TATS are to split S into a sequence of segments $B = [b_1, ..., b_M]$ and

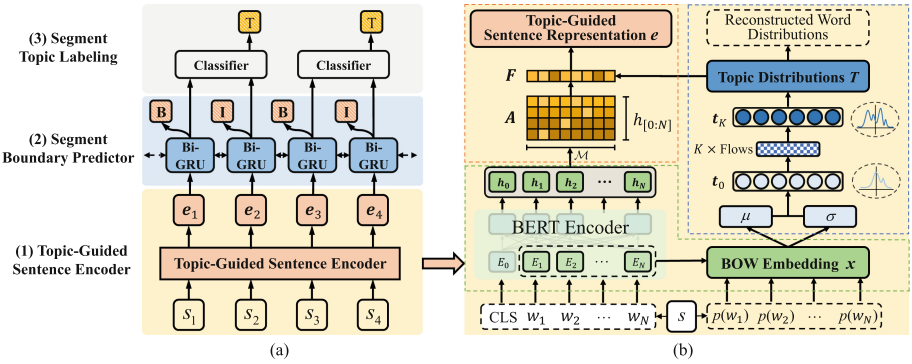

Fig. 2. (a) The Tipster model takes a given sequence of sentences as inputs, and outputs the segment boundaries and each segment topic label. (b) The topic-guided sentence encoder consists of a pre-trained language model and a neural topic model.

assign one topic t_i (single-label task) or multiple topics T_i (multi-label task) to each segment b_i. The process of topic classification is formulated as follows:

$$t_i = f_{\text{single}}(b_i) \quad \text{and} \quad T_i = f_{\text{multiple}}(b_i), \tag{1}$$

where f_{single} is the function that how we assign a single topic $t_i \in C$ to the segment b_i, while f_{multiple} is the function that how we assign multiple topics $T_i \subseteq C$ to the segment b_i. Our model handles both two circumstances.

3 Tipster for Topic-Aware Text Segmentation

3.1 Topic-Guided Sentence Encoder

Contextual and BOW Representations. The topic-guided sentence encoder is designed as a cooperative architecture with the pre-trained BERT [4] and a neural topic model (NTM), as shown in Fig. 2(b). Considering a word sequence of sentence $\boldsymbol{s} = \{w_i\}_{i=1}^{N}$ as input, where N is the number of words, we obtain contextual representations $\boldsymbol{h}_{[0:N]}$ of each word from the last hidden layer of BERT, where $\boldsymbol{h}_0, \boldsymbol{h}_j \in \mathbb{R}^d$ is the contextual representation of *CLS* token and word w_j respectively, and d is the hidden layer size of BERT.

Inspired by Variational Encoder [7,12], we build a NTM to learn non-linear topic-word distributions from $\boldsymbol{h}_{[1:N]}$. We process the sentence $\boldsymbol{s}$ into a bag-of-words (BOW) input $\boldsymbol{s_{bow}} \in R^{|\mathcal{V}|}$, where $\mathcal{V}$ denotes the word vocabulary with the stopwords removed. Then we derive the BOW embedding $\boldsymbol{x}$ of sentence $\boldsymbol{s}$ with the distribution $p(w)$ and pre-trained word embedding matrix $\boldsymbol{E}$ of BERT:

$$p(w_i) = \frac{(\boldsymbol{s_{bow}})_i}{\sum_{w \in \mathbf{s}} (\boldsymbol{s_{bow}})_w} \quad \textit{and} \quad \boldsymbol{x} = \sum\nolimits_{w \in \mathbf{s}} (p(w)\boldsymbol{E}_w). \tag{2}$$

Latent Topic Distributions. We build a NTM to exploit the complex relation between the distributions of words and latent topics. Our NTM consists of two components: a inference network to infer the latent topic distributions and a decoder network to reconstruct the input word distributions.

1) *Normalizing Flows-Based Posterior.* The majority of NTMs design the variational distributions of topics via simplest multivariate Gaussian with diagonal covariance, which may not be expressive to approximate the true complicated posterior of latent topics. Therefore, under the assumption of keeping prior Gaussian, we utilize the normalizing flows [11] to infer the more flexible posterior distributions of latent topics. Specifically, we assume the prior of topic distribution is a multivariate Gaussian with diagonal covariance $p_\lambda(\boldsymbol{t}) \sim \mathcal{N}(0, \boldsymbol{I})$. We formulate the inference process as follows:

$$\begin{aligned} g = f_{\mathrm{MLP}}(\boldsymbol{x}),\ \boldsymbol{\mu} = l_1(g),\ \log \boldsymbol{\sigma} = l_2(g), \\ q_\phi(\boldsymbol{t} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{t} \mid \boldsymbol{\mu}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2(\boldsymbol{x}))), \end{aligned} \tag{3}$$

where f_{MLP} denotes the multi-layer perceptron (MLP), l_1 and l_2 are linear transformation functions with bias, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the mean and the standard deviation of the Gaussian posterior. To reduce the variance in stochastic estimation, we use the reparameterize trick [7,12] by sampling $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$, and reparameterizing $\boldsymbol{t} = \boldsymbol{\mu} + \epsilon \times \boldsymbol{\sigma}$.

Since the Gaussian posterior $q_\phi(\boldsymbol{t} \mid \boldsymbol{x})$ may not be sufficiently flexible with the true posterior, we utilize the normalizing flows to transform the variational posterior into more complex density via a sequence of invertible transformation $\boldsymbol{t}_K = f_K(f_{K-1}(...f_1(\boldsymbol{t}_0)))$, where we suppose that the initial sample $\boldsymbol{t}_0 = \boldsymbol{t}$ from $q_\phi(\boldsymbol{t} \mid \boldsymbol{x})$, and apply K parameterized invertible transformation f to obtain the final sample $\boldsymbol{t}_K$. The probability density of the final sample $\boldsymbol{t}_K$ is defined through the variational method as:

$$q_K\left(\boldsymbol{t}_K \mid \boldsymbol{x}\right) = q_\phi\left(\boldsymbol{t}_0 \mid \boldsymbol{x}\right) \prod_{k=1}^{K} \left| \det \frac{\partial f_k\left(\boldsymbol{t}_{k-1}; \phi_k\right)}{\partial \boldsymbol{t}_{k-1}} \right|^{-1}, \tag{4}$$

where the last term $|\circ|$ denotes the Jacobian determinant and ϕ_k is the parameter of the k-th transformation. As for our model, we consider the Planar flows [11] as the transformation:

$$f_k(\boldsymbol{t}_{k-1}; \phi_k) = \boldsymbol{t}_{k-1} + \boldsymbol{u}_k \cdot \tanh(\boldsymbol{w}_k^T \boldsymbol{t}_{k-1} + \boldsymbol{b}_k), \tag{5}$$

where $\phi_k = \{\boldsymbol{u_k}, \boldsymbol{w_k} \in \mathbb{R}^m, \boldsymbol{b}_k \in \mathbb{R}\}$, $\left|\det \frac{\partial f_k}{\partial \boldsymbol{t}_{k-1}}\right| = \left|1 + \boldsymbol{u}_k^\top \psi_k\left(\boldsymbol{t}_{k-1}\right)\right|$. The planar flows apply contractions and expansions in the perpendicular direction to the hyperplane $\boldsymbol{w}_k^T \boldsymbol{t}_{k-1} + \boldsymbol{b}_k = 0$.

2) *Word Distribution Reconstruction.* After we obtain the posterior distribution of topics $\boldsymbol{t}_K$, we explicitly reconstruct the sentence $\boldsymbol{s}$ by independently generating the word $e_{\mathrm{NVTM}} \rightarrow w_i$ via a dense layer θ:

$$p_\theta\left(w_i \mid \boldsymbol{t}_K\right) = \mathrm{Softmax}\left(E\left(w_i; \boldsymbol{t}_K, \theta\right)\right), \tag{6}$$

where $E(w_i; \boldsymbol{t}_K, \theta) = \boldsymbol{t}_K^T \boldsymbol{W} w_i + \boldsymbol{b}_{w_i}$, $\theta = \{\boldsymbol{W} \in \mathbb{R}^{m \times |\mathcal{V}|}, \boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}|}\}$. This reconstruction process encourages the latent topic distributions to contain the most important words in the sentence.

3) *Evidence Lower Bound.* To maximize the marginal likelihood $\log p(\boldsymbol{x})$ of sentence $\boldsymbol{s}$, we derive the loss function of NTM from *evidence lower bound* [11]:

$$\begin{aligned} \mathcal{L}_{\text{NTM}} = &- \sum_{i=1}^{|\mathcal{V}|} p(w_i) \log p_\theta(w_i \mid \boldsymbol{t}_K) + \text{KL}\left(q_\phi(\boldsymbol{t}_0 \mid \boldsymbol{x}) \| p_\lambda(\boldsymbol{t}_0)\right) \\ &- \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k(\boldsymbol{t}_{k-1})}{\partial \boldsymbol{t}_{k-1}} \right|. \end{aligned} \tag{7}$$

where the first term of $\mathcal{L}_{\text{NTM}}$ is the reconstruction loss of word distributions, and the second term of $\mathcal{L}_{\text{NTM}}$ is the Kullback-Leibler divergence between posterior and prior of the first layer topic distributions. Based on the sample $\boldsymbol{t}_0 \sim q_\phi(\boldsymbol{t}_0 \mid \boldsymbol{x})$, the parameters of our NTM can be optimized by back propagating the stochastic gradients. Afterwards, we obtain the latent topic distributions $\boldsymbol{T} \in (0,1)^m$ of sentence $\boldsymbol{s}$ via a Softmax function on the output topic $\boldsymbol{t}_K$.

Topic-Guided Sentence Representation. Considering that the topics $\boldsymbol{T}$ in the NTM have different meanings, we build a memory bank $\mathcal{M} = \langle \boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_{|\boldsymbol{T}|} \rangle$ to represent each topic. Moreover, such a memory bank facilitates reasonable attention on the meaningful words by projecting the space of NTM to the contextual representations $\boldsymbol{h}_{[1:N]}$. We hypothesise that each word has a different contribution to each topic. The memory bank of topics is used to determine the attention matrix $\boldsymbol{A} \in \mathbb{R}^{(N+1) \times m}$ between contextual representations of words $\boldsymbol{h}_{[0:N]} \in \mathbb{R}^{(N+1) \times d}$ and topic embeddings $\mathcal{M} \in \mathbb{R}^{m \times d}$, and then we obtain multiple topic facets $\boldsymbol{F} \in \mathbb{R}^{m \times d}$ through $\boldsymbol{A}$ and $\boldsymbol{h}_{[0:N]}$:

$$\boldsymbol{A}_{jk} = \frac{\exp(\cos(\boldsymbol{m}_k, \boldsymbol{h}_j))}{\sum_{j'=0}^{N} \exp(\cos(\boldsymbol{m}_k, \boldsymbol{h}_{j'}))} \quad \textit{and} \quad \boldsymbol{F} = \boldsymbol{A}^T \cdot \boldsymbol{h}_{[0:N]}, \tag{8}$$

where cos denotes the cosine similarity. The topic guided sentence representation $\boldsymbol{e} \in \mathbb{R}^d$ is obtained by aggregating $\boldsymbol{F}$ weighted by topic distributions $\boldsymbol{T}$:

$$\boldsymbol{e} = \sum_{i=1}^{m} \boldsymbol{T}_i \boldsymbol{F}_i. \tag{9}$$

3.2 Segment Boundary Predictor

After getting the sentence embedding $\boldsymbol{e}$, we use a two-layer bidirectional Gated Recurrent Units (Bi-GRU) to predict segment boundaries in the document. This sub-network takes a sequence of sentence embedding $\boldsymbol{e}_{1,2,...,M} \in R^d$ as input, feeds them into Bi-GRU and predicts two classes, B or I, representing whether the sentence t is (B)eginning or (I)nside of a segment. We formulate the segment boundary predictor as:

$$\overleftrightarrow{\boldsymbol{h}_t} = f_{\text{Bi-GRU}}\left(\boldsymbol{e}_t, \overrightarrow{\boldsymbol{h}_{t-1}}, \overleftarrow{\boldsymbol{h}_{t+1}}\right), \quad p_t(\text{B;I}) = \text{Softmax}\left(W_s \overleftrightarrow{\boldsymbol{h}_t} + b_s\right). \tag{10}$$

3.3 Segment Topic Labeling

The last step of our model is to assign related topic labels to each segment. We obtain the embedding H_i of segment i by concatenating the mean pooling, max pooling and attention pooling [14] of contained sentences embedding $H_i = [\text{Mean}(\overleftrightarrow{\boldsymbol{h}_j}); \text{Max}(\overleftrightarrow{\boldsymbol{h}_j}); \text{Attn}(\overleftrightarrow{\boldsymbol{h}_j})]$. Then we assign each segment one topic t with a Softmax activation output layer for single topic classification. While for the multi-topic classification, we assign each segment multiple topics T by replacing the Softmax with Sigmoid function. We formulate segment labeling process as:

$$t_i = \text{Softmax}(W_y H_i + b_y) \quad \text{and} \quad T_i = \text{Sigmoid}(W_z H_i + b_z). \tag{11}$$

3.4 Joint Learning

We jointly train Tipster with multi-task loss function:

$$\mathcal{L} = (1 - \alpha - \beta)\mathcal{L}_{\text{Seg}} + \alpha\mathcal{L}_{\text{Topic}} + \beta\mathcal{L}_{\text{NTM}}, \tag{12}$$

where $\mathcal{L}_{\text{Seg}}$ and $\mathcal{L}_{\text{Topic}}$ are segmentation loss and topic labeling loss, which are cross entropy calculated at sentence level and segment level respectively.

4 Experiments

Experimental Setup. We conduct extensive experiments to demonstrate the effectiveness of Tipster on five public datasets: *Wiki-Section* [1], *Wiki-50* [8], *Cities* [3], *Elements* [3] and *Clinical* [5]. *Wiki-Section* contains four Wikipedia datasets across various languages and domains: English Language City(**en_city**), English Language Disease (**en_disease**), German Language City (**de_city**) and German Language Disease (**de_disease**). Table 1 shows their Statistics. We compare our model with seven popular methods: TopicTiling [13], Textseg [8], SEC>T+emb [1], CATS [6], S-LSTM [2], Cross-Segment BERT (CS-BERT) and BERT-LSTM [10]. For Tipster, we use pre-trained model *bert-base-uncased* for English Datasets as bert embedding, *bert-base-german-cased* for German Datasets. For topic-guided sentence embedding, we set both the number of topics

Table 1. Statistics of all datasets used in our experiments.

Dataset	*Wiki-Section*				*Wiki-50*	*Cities*	*Elements*	*Clinical*
	en_disease	de_disease	en_city	de_city				
# Docs	3.6k	2.3k	19.5k	12.5k	50	100	118	227
# Seg/Doc	7.5	7.2	8.3	7.6	3.5	12.2	6.8	5.0
# Sent/Seg	7.8	6.3	6.8	5.3	13.6	5.2	3.3	28.0
Topic label	✓	✓	✓	✓	✓	✓	✓	✗

Table 2. Results for text segmentation, single-label and multi-label classification on *WikiSection* datasets. "n/a" denotes the model is inapplicable to the subtask.

WikiSection	en_disease			de_disease			en_city			de_city		
single-label	27 topics			25 topics			30 topics			27 topics		
model	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP	↓ P_k	↑ F_1	↑ MAP
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
S-LSTM	21.2	57.5	70.9	19.7	52.3	67.1	10.5	74.5	82.2	10.2	75.8	83.6
CS-BERT	18.7	n/a	n/a	20.5	n/a	n/a	11.7	n/a	n/a	11.6	n/a	n/a
BERT-LSTM	16.8	n/a	n/a	15.3	n/a	n/a	9.3	n/a	n/a	9.8	n/a	n/a
Tipster	**14.2**	**62.2**	**74.6**	**13.7**	**57.0**	**70.8**	**8.3**	**79.8**	**86.2**	**7.9**	**78.8**	**86.3**
Tipster-NTM	15.5	59.3	72.1	15.3	55.8	69.3	9.0	77.4	84.7	10.7	78.4	85.9
multi-label	179 topics			115 topics			603 topics			318 topics		
model	↓ P_k	↑ P@1	↑ MAP	↓ P_k	↑ P@1	↑ MAP	↓ P_k	↑ P@1	↑ MAP	↓ P_k	↑ P@1	↑ MAP
SEC>T+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	22.1	52.7	59.8	19.5	35.4	45.2	10.2	73.1	71.4	10.7	73.7	74.5
Tipster	**13.7**	**60.8**	**66.2**	**15.9**	**46.4**	**56.9**	**8.4**	**79.0**	**76.2**	**7.5**	**78.6**	**79.3**
Tipster-NTM	14.8	58.4	63.5	17.6	45.9	54.8	9.1	77.5	74.3	8.9	76.9	78.1

Table 3. Results for transferring evaluation. Models marked with △ are trained on the big corpus *Wiki-727K*, while the models marked with □ are trained on en_city for *Wiki-50* and *Cities*, en_disease for *Elements* and *Clinical*.

Segmentation	Wiki-50		Cities		Elements		Clinical
multi-label	↓ P_k	↑ MAP	↓ P_k	↑ MAP	↓ P_k	↑ MAP	↓ P_k
TextSeg△	18.2	n/a	19.7	n/a	41.6	n/a	**30.8**
SEC>H+emb□	40.5	13.4	33.3	53.6	43.3	9.5	36.5
S-LSTM□	22.7	16.6	21.2	54.2	30.2	19.1	36.1
Tipster□	19.2	**20.8**	18.2	**59.4**	27.3	**22.5**	32.9
CATS△	**16.5**	n/a	**16.9**	n/a	**18.4**	n/a	-

and normalized flow as 5. We set the GRU layer size to 128. We train our model with ADAM optimizer, 5e-4 learning rate. The tradeoff α and β are both set to 0.1 in all experiments. Following [1,2], we adopt P_k as the segmentation measure. As for segment topic labeling, we report *F1* and *Mean Average Precision* (MAP) for single-label task, *Precision@1* and MAP for multi-label task.

Performance in Single-Label and Multi-label Task. As shown in Table 2, we evaluate Tipster on *WikiSection* for both the single-label and multi-label TATS. Regarding segmentation error, our model reduced P_k significantly by 1.5 points on average compared to BERT-LSTM. As for the single-topic classification, our model reached 69.5% on F_1 and 79.5% on MAP on average, which improved over 4.5 and 3.5 points compared to S-LSTM, respectively. While for the multi-topic classification, which is more closely to real-world scenarios,

Tipster has a consistent improvement for the single-topic classification, which demonstrates the favorable robustness of our model under complex scenarios. From the overview, our model achieves the SOTA performance, which suggests that incorporating word distributions and assign rational attention weight to meaningful words contribute to the task. To validate that the proposed module of topic model embedding incorporating word distributions contributes to the TATS, we remove the NTM and obtain the sentence representations via average pooling, which we denote as "Tipster-NTM". Table 2 shows that Tipster without the NTM reduces the performance in segmentation and topic classification. Therefore, performing reasonable attention on the meaningful words assists with TATS.

Transferring Evaluation. We evaluate Tipster transferability on *Wiki-50*, *Cities*, *Elements* and *Clinical* by training it on the *WikiSection* datasets of corresponding domains. Table 3 shows the results for transferring evaluation on four existing datasets. Owing to the large-scale training corpus *Wiki-727k* [8], CATS outperforms other models on *Wiki-50*, *Cities* and *Elements* datasets. Our model, Tipster outperforms the other supervised models trained on the tiny *WikiSection* and other mainstream unsupervised models. This result illustrates that our model is equipped with a favorable transferability.

5 Conclusion

In this paper, we introduced Tipster, a topic-guided model explicitly incorporating word distributions for topic-aware segmentation task. Our model not only achieved the SOTA performance on TATS, but also showed a novel pooling attention mechanism for TATS. Finally, we conduct extensive experiments to demonstrate the effectiveness and transferability of Tipster.

Acknowledgement. This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 61922073 and U20A20229), and the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, P. R. China (No. CI0S-2020SC05).

References

1. Arnold, S., Schneider, R., Cudré-Mauroux, P., Gers, F.A., Löser, A.: SECTOR: a neural model for coherent topic segmentation and classification. Trans. Assoc. Comput. Linguist. **7**, 169–184 (2019)
2. Barrow, J., Jain, R., Morariu, V., Manjunatha, V., Oard, D.W., Resnik, P.: A joint model for document segmentation and segment labeling. In: Proceedings of ACL (2020)
3. Chen, H., Branavan, S., Barzilay, R., Karger, D.R.: Content modeling using latent permutations. J. Artif. Intell. Res. **36**, 129–163 (2009)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of ACL (2019)

5. Eisenstein, J., Barzilay, R.: Bayesian unsupervised topic segmentation. In: Proceedings of EMNLP (2008)
6. Glavaš, G., Somasundaran, S.: Two-level transformer and auxiliary coherence modeling for improved text segmentation. In: Proceedings of AAAI (2020)
7. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: Proceedings of ICLR (2014)
8. Koshorek, O., Cohen, A., Mor, N., Rotman, M., Berant, J.: Text segmentation as a supervised learning task. In: Proceedings of ACL (2018)
9. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: Proceedings of EMNLP (2020)
10. Lukasik, M., Dadachev, B., Papineni, K., Simões, G.: Text segmentation by cross segment attention. In: Proceedings of EMNLP (2020)
11. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: Proceedings of ICML (2015)
12. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of ICML (2014)
13. Riedl, M., Biemann, C.: TopicTiling: a text segmentation algorithm based on LDA. In: Proceedings of ACL 2012 Student Research Workshop (2012)
14. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of ACL (2016)