TV NEWS STORY SEGMENTATION USING DEEP NEURAL NETWORK

Zhu Liu

AT&T Labs - Research
200 South Laurel Avenue
Middletown, NJ, 07748 USA
zliu@research.att.com

Yuan Wang

Tandon School of Engineering
New York University
2 Metrotech Center, Brooklyn, NY, 11201 USA
yw1224@nyu.edu

ABSTRACT

TV news programs usually contain multiple stories on different topics, and it is essential to locate the story boundaries for the purposes of video content indexing, search, and curation. With the exponential growth of video content, story segmentation will enable the consumers to view their favorite content effortlessly, and the service providers to provide their customers with personalized services. Given the dynamic range of topics, smooth story transitions in news, and varying duration of individual story, automated news story segmentation is a challenging task. This paper focuses on using linguistic information extracted from closed caption as the initial attempt to tackle this challenge, and our future work will integrate both audio and visual. A convolutional neural network framework with attention mechanism is proposed. The model is trained and tested on TDT2 data set, and it achieves an outstanding F-measure of 0.789 on the validation set and a reasonable F-measure of 0.707 on the testing set.

Index Terms— Story segmentation, convolutional neural network, TV news, NLP

1. INTRODUCTION

Providing customers with effortless multimedia consuming experience is essential for video service providers. This paper focuses on closed caption-based TV news story segmentation, which partitions TV news programs into semantically meaningful units (stories). This technology is critical for video curation, video personalization, targeted advertising, and other value-added services [1].

Existing text-based story segmentation research can be grouped into two categories: model-based and detection-based methods. For model-based approaches, there is an explicit story or labeling variable. Story boundaries can be derived as the change of the story variable. For detection-based approaches, there is no explicit variable labeling stories. Instead, story boundaries are detected as the salient changes of textual cohesion. These methods usually share a common processing pipeline with 3 steps: 1) represent or encode each processing unit (e.g., sentence) as a fixed length vector; 2) calculate cohesion; and 3) detect story boundaries. Our proposed method belongs to the detection-based category.

Recently, Deep learning (DL), or deep neural net (DNN), elevates the performance of natural language processing (NLP) tasks to a new level. Yet research on text-based story segmentation using DNN is very limited. In this paper, a new convolutional neural network with attention is proposed. It casts boundary detection as a classification problem. Applying a sliding window on sentence sequence, it outputs the probability of story boundary for each sentence. The model is trained and tested on TDT2 data set, and it achieves an outstanding F-measure of 0.789 on the validation set and a F-measure of 0.707 on the testing set.

The rest of the paper is organized as follows. Section 2 reviews some related work, and section 3 illustrates the proposed method. Experimental results are discussed in section 4, and finally section 5 draws the conclusions.

2. RELATED WORK

In model-based category, latent semantic analysis (LSA) [2] is one of the earliest approach. Following works improved on LSA include probabilistic LSA (PLSA) [3], latent Dirichlet allocation (LDA) [4], etc. Other models include Bayesian unsupervised topic segmentation (BayesSeg) [5], locally-consistent topic modeling (LTM) [6] and hidden Markov model (HMM) [7]. These methods have their shortcomings. First, the number of topics need to be pre-fixed which is not practical for different documents. Second, these models typically use variational inference for parameter estimation, which is not compatible with DL.

The approaches in detection-based category vary by the difference in the three embedded steps. For processing unit representation, the most classical one in bag of words (BOW) vector, which tends to be high dimensional and sparse. Several embedding methods have been proposed to find short dense vector representation for the processing units such as Laplacian Eigenmaps (LE) embedding [8], isometric feature mapping (Isomap) [9], etc. These hand-crafted methods work more like dimension reduction rather

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

than learning representation from data. For cohesion calculation, cosine similarity, lexical score, cross entropy, and linear correlation are widely used. For story boundary detection, typical methods include exploration of local cohesion property heuristically, dynamic programming, watershed [10], spectral clustering [11], etc.

HMM is used jointly with DNN in [7], where DNN directly maps the word distribution into topic posterior probabilities. However, DNN fits much better in the detection framework rather than model-based framework, as it can handle all the three steps within one network. DNN have various structures to learn dense vector representations for text, such as continuous BOW (CBOW) [12], deep autoencoder [13], convolutional neural network (CNN), and recurrent neural network with attention.

3. THE PROPOSED METHOD

In this section, we first introduce the overall framework and then cover each module with details.

3.1 Overall framework

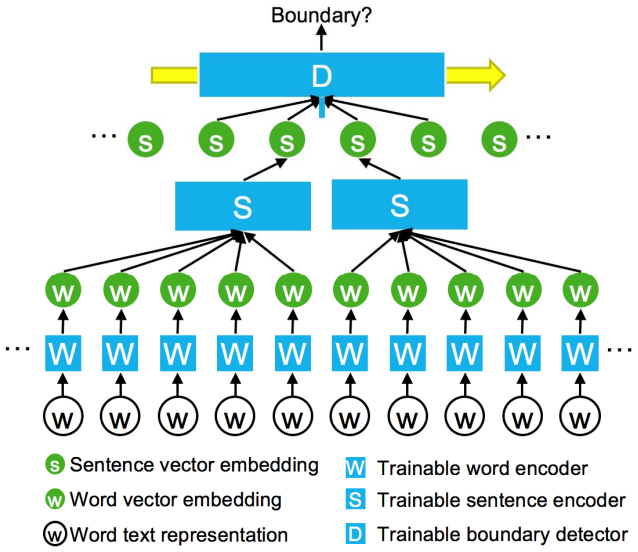


Fig. 1. Proposed framework. The blue components are configurable and trainable neural networks.

Fig. 1 shows the proposed hierarchical framework. The closed caption of a program, or the document, is treated as a list of sentences and each sentence is composed of a list of words. At the bottom of the framework, this model takes raw text as input. A word encoder converts each word into a fixed length dense vector representation (or embedding) and a sentence encoder assembles the information in the word list embeddings into a sentence embedding. A story boundary detector evaluates several continuous sentence embeddings, comparing the first half of the sentences to the second half, and claims a boundary if there is a salient lexical cohesion change between the two sets of sentences. The detector works as a sliding window on the sentence sequence and attempts to make a decision at each sentence of the sequence. The word encoder, the sentence encoder, and the boundary detector, denoted by blue rectangles in Fig. 1, are configurable and trainable neural networks.

3.2 Word encoder

In this work, we use Google word2vec as the word encoder, more specific, GoogleNews-vector-negative300. It has a vocabulary of 3 millions, with a word embedding dimension of 300. It is trained using 100 billion words in Google News dataset.

3.3. Sentence encoder

We use continuous bag of words with attention for sentence encoder. The intuition behind the attention mechanism is that not all the words are equally important for story segmentation. Some words almost carry no information about the story, such as *the*, *a*, *there*, etc. While in some situations, one can grasp the basic topic of a story with one or two words, such as *white house*, *US army*, *super bowl*, etc. The attention mechanism can filter out the unimportant words while emphasizing the informative ones.

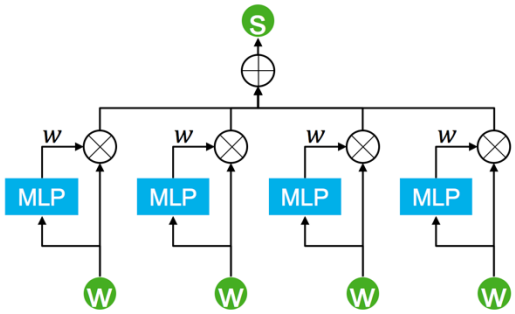


Fig. 2. Sentence encoder based on continuous bag of words with attention.

As shown in Fig. 2, the sentence embedding is a weighted summation of its word embeddings where the attention weights are obtained by a trained by-pass multi-layer perceptron (MLP) network. Apart from classical attention model, the weights for a sentence range from 0 to 1 but do not sum to 1. This attention weight is similar to the classical TF-IDF but superior to TF-IDF in the following senses. First, it is jointly optimized with the subsequent story boundary detector. Second, it implicitly regulates that semantically similar words have similar weights regardless their frequency in the training dataset since semantically similar words have similar embeddings.

3.4 Boundary detector

Fig. 3 shows the boundary detector which is a variant of [14] for sentence classification. Sentences under evaluation

are stacked into a matrix, each row representing a sentence embedding. Filters with various length but fixed width matching the sentence embedding are applied on the sentence matrix for 1D convolution along the temporal dimension. Then max pooling is applied on each feature vector obtained from previous convolution and the maximum values are concatenated into an intermediate feature vector. Finally, it passes through a MLP and a softmax layer to generate a binary label and associated probability.

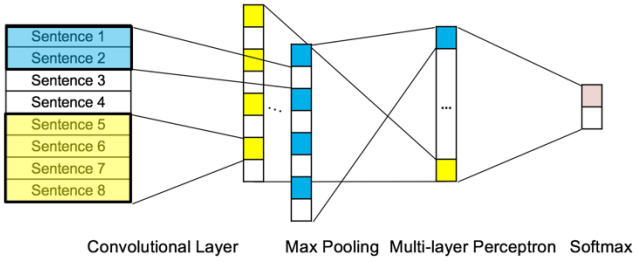


Fig. 3. Boundary detector based on CNN.

The boundary detector works as a sliding window on sentence sequence. At each sentence, it outputs a binary label and the associated probability. Once a candidate boundary is declared, its neighborhood is searched for a maximum boundary probability. The location with the maximum value is claimed as boundary and other boundary candidates in the neighborhood are depressed. More implementation details of the network structures are presented in Section 4.2.

4. EXPERIMENTS

4.1. Dataset

In this study, we use TDT2 corpus, the 1998 Topic Detection and Tracking data set, for training and testing. TDT2 is created to develop technologies for retrieval and automatic organization of Broadcast News and Newswire stories and to evaluate the performance of those technologies. It consists of data collected during the first half of 1998 from 6 sources, including 2 newswires, 2 radio programs and 2 television programs. Only English stories labeled as "NEWS" are included in our study. In total, we have 2,690 programs, which contain 55.6 thousand stories, 1.05 million sentences, and 23.1 million words. The data set is randomly partitioned into training, validation, and testing sets, with a ratio of 16:1:1.

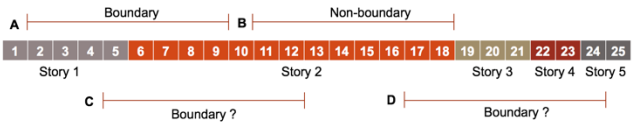


Fig. 4. Assigning binary label to sentence sequence segments.

Although story-level labels are provided, the boundary detector makes a decision on every sentence which means sample-level labels are needed. Assigning each sample a boundary/non-boundary label can be tricky. An example program with 25 sentences and 5 stories are shown in Fig. 4. In case A, there is a boundary right in the middle of the sentence sequence segment, it is obviously a 'boundary' sample. In case B, the whole sentence sequence segment is in one story, it is clearly a 'non-boundary' sample. While in case C, the segment covers a boundary but not in the middle, and in case D, the segment covers multiple stories. It is arguable on how to assign binary label to these two cases. As a result, we only keep types A and B in the training set and validation set. During testing, because our model works as a sliding window on the sentence sequence, it is unavoidable to include types C and D.

4.2 Implementation details

The window length of the boundary detector is set to 8 sentences. The unit numbers of the MLP in attention model are 300, 64, 32, 1 and the activation functions are Relu, Relu, Relu, and Reule+tanh. Its input dimension matches the sentence embedding dimension and it outputs one scalar as the attention weight. The filter sizes of the CNN layer in detector are 2x300, 4x300, and 8x300 with kernel numbers of 64, 32, and 16. The unit numbers of the MLP in detector are 112, 32, 2 and the activation functions are Relu, Relu, and None.

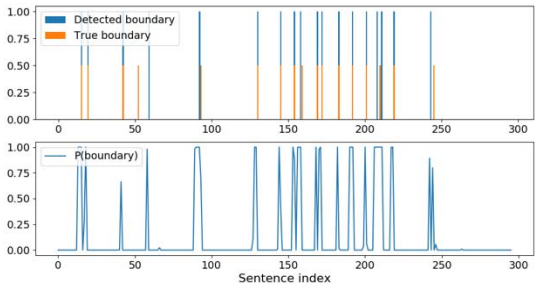
Our system is implemented using Google Tensorflow, and the model is trained with the adaptive gradient descend method. It holds one step size for each parameter and the step sizes are automatically tuned to keep the loss function decreasing. It typically eliminates the burden of choosing learning rate. To avoid overfitting, we ensure that the number of training samples is much larger than the number of parameters. In terms of regulation, we use dropout at the MLP in detector with 50% drop rate. We did not observe increasing of the loss value on validation set as training with more epochs. Our model takes about 12 hours to converge on a NVIDIA Tesla K40 GPU.

Precision, recall, and F-measure are used to evaluate the performance numerically. Precision is defined as the ratio between the number of true detections and the number of all detections. Recall is the ratio between the number of true detections and the number of true boundaries. F-measure is defined as twice the product of precision and recall divided by the sum of the two. In evaluation, if a detected boundary is within 3 sentences from a true boundary, it is treated as a true detection; otherwise false.

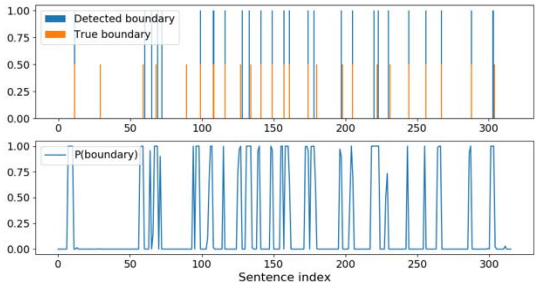
4.3 Experimental results

We show two program segmentation examples in Fig. 5. In some cases, the boundary detector can locate a boundary as a single impulse in boundary probability such as the

boundary just before sentence 150 in Fig. 5 (a). But in other cases, it assigns high boundary probability to the entire neighborhood of a boundary. This suggests that, the detector trained on samples of types A and B (explained in section 4.1) can handle types C and D well in some cases, but not all of them. Overall, the detected boundaries and the true boundaries are highly correlated.



(a)



(b)

Fig. 5. Two examples of program level segmentation results. The horizontal axis is sentence index. In each example, the bottom figure plots the probability of boundary for each sentence; the top figure plots the detected and the true boundaries in blue and orange.

Numerical performance measurements of the proposed model are list in Table 1. It shows that the model achieves similar precision on both validation and testing sets, while the recall is better in validation set. We believe that the performance gap between the validation set and the testing set is mainly due to the difference of samples included in each set: the samples in validation set are cleaner than the ones in testing. It suggests that better preparation of the training samples may further improves the overall result. Considering that the best F-measure on TDT2 reported in the latest literature is about 0.77 [15], our method achieved a great performance on the validation set but there is still room to improve on the testing set.

Table 1. Numerical performance metrics of the proposed model

Measurement	Validation set	Testing set
Precision	0.766	0.767
Recall	0.813	0.682
F-measure	0.789	0.707

5. CONCLUSIONS

We proposed a novel closed caption-based news story segmentation algorithm using deep neural network. It uses google word2vec as word encoder, continuous bag of words with attention as sentence encoder, and a sliding window boundary detector with convolutional neural network structure. The proposed method is trained and tested on TDT2 data set. It achieves an outstanding performance on the evaluation set with an F-measure of 0.789 and a good performance on the testing set with an F-measure of 0.707. As a potential future work, we will incorporate audio and visual components in TV programs for story segmentation.

6. REFERENCES

[1] D. Gibbon and Z. Liu, Introduction to Video Search Engines, Springer, 2008.

[2] F. Choi, P. Wiemer-Hastings, and J. Moore, "Latent semantic analysis for text segmentation," *Proc. of EMNLP'01*, pp. 109-117, 2001.

[3] T. Hofmann, "Probabilistic latent semantic indexing," *Proc. 22nd Annu. Int. ACM SIGIR'99*, pp. 50-57, 1999.

[4] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *JMLR*, vol. 3, pp. 993-1022, 2003.

[5] J. Eisenstein and R. Barzilay, "Bayesian unsupervised topic segmentation," *Proc. of EMNLP*, 2008.

[6] C. Yang, L. Xie, and X. Zhou, "Unsupervised broadcast news story segmentation using distance dependent Chinese restaurant processes," *ICASSP*, 2014.

[7] J. Yu, X. Xiao, L. Xie, E. S. Chng, and H. Li, "A DNN-HMM approach to story segmentation," *Proc. of INTERSPEECH*, pp. 1527–1531, 2016.

[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Journal of Neural Computation*, vol. 15, 2003.

[9] J. Tenenbaum, V. De Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[10] V. Claveau and S. Lefevre, "Topic segmentation of TV-streams by watershed transform and vectorization," *Computer Speech & Language*, vol. 29, no. 1, 2015.

[11] L. Xie, L. Zheng, Z. Liu, and Y. Zhang, "Laplacian eigenmaps for automatic story segmentation of broadcast news," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, 2012.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *NIPS*, 2013.

[13] G. Kumar and L. F. D'Haro, "Deep autoencoder topic model for short texts," *Proc. of IWES*, 2015.

[14] Y. Kim, "Convolutional neural networks for sentence classification," *Proc. Of EMNLP*, pp. 1746–1751, 2014.

[15] J. Yu, L. Xie, X. Xiao, and E. Siong, "A hybrid neural network hidden Markov model approach for automatic story segmentation," *JAIHC*, vol. 8, 2017.