Neural Discourse Segmentation

Jing Li
Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates
jingli.phd@hotmail.com

Abstract

Identifying discourse structures and coherence relations in a piece of text is a fundamental task in natural language processing. The first step of this process is segmenting sentences into clause-like units called elementary discourse units (EDUs). Traditional solutions to discourse segmentation heavily rely on carefully designed features. In this demonstration, we present SEGBOT, a system to split a given piece of text into sequence of EDUs by using an end-to-end neural segmentation model.[1] Our model does not require hand-crafted features or external knowledge except word embeddings, yet it outperforms state-of-the-art solutions to discourse segmentation.

1 Introduction

Identifying discourse structures and coherence relations has an important role in various natural language processing (NLP) applications such as text summarization [Durrett *et al.*, 2016; Li *et al.*, 2018d], information extraction [Li *et al.*, 2018a], question answering [Li *et al.*, 2018c] and passage retrieval [Dias *et al.*, 2007; Li *et al.*, 2018e]. The first necessary step of this process is to perform **discourse segmentation**, which refers to the task of breaking a piece of text into a sequence of elementary discourse units or EDUs [Marcu, 2000]. As exemplified in Figure 1, EDUs are clause-like units that serve as building blocks for discourse parsing in Rhetorical Structure Theory or RST [Mann and Thompson, 1988]. If the given text is wrongly segmented in this stage, the error propagates to the subsequent steps and it becomes unreliable to assign correct discourse relations. Thus, discourse segmentation is a crucial step for effective discourse parsing [Joty *et al.*, 2015]. It is also important for downstream applications like text compression [Sporleder and Lapata, 2005] and machine translation [Joty *et al.*, 2017].

Discourse segmentation can be treated as a sequence labeling problem, where the task is to predict a sequence of 'yes/no' boundary tags. Traditional approaches to sequence labeling use conditional random fields (CRFs) with hand-crafted features. Recent methods use recurrent neural networks (RNNs) with possibly a CRF layer at the output [Lample *et al.*, 2016; Xu *et al.*, 2019]. However, when it comes to EDU segmentation, studies have shown that sequence labeling does not provide any additional gain over simple binary classification (*e.g.,*, using MaxEnt) due to the sparsity of 'yes' boundary tags [Fisher and Roark, 2007; Joty *et al.*, 2015]. Therefore, in our work, we frame the task as a sequence generation task with a seq2seq model [Sutskever *et al.*, 2014]. Our goal is to generate a sequence of indices in the input text each indicating an EDU boundary; see Figure 2(a). Notice that the set of candidate positions changes at each decoding step. This is unlike the existing seq2seq model, where the output vocabulary of the decoder RNN is fixed at each decoding step.

[The bank also says]$_{\text{EDU}}$ [it will use its network]$_{\text{EDU}}$ [to channel investments]$_{\text{EDU}}$

Figure 1: A sentence with three elementary discourse units (EDUs).

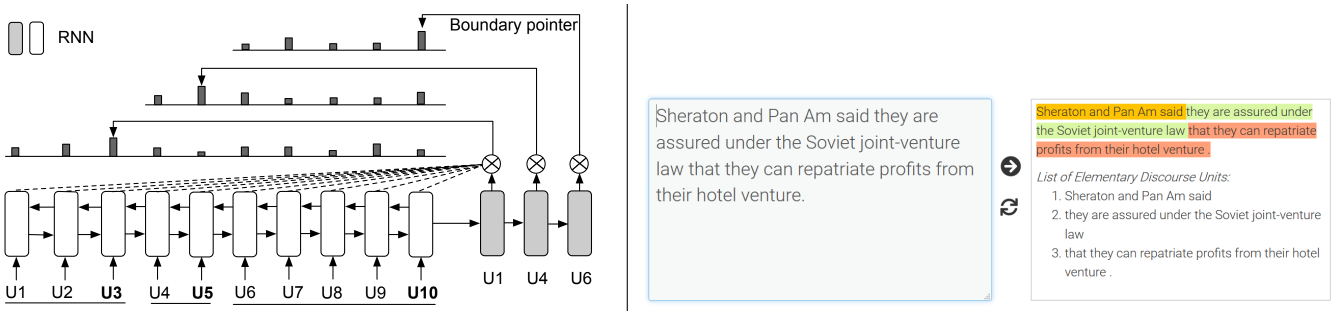
To alleviate these issues, we propose SEGBOT, an end-to-end neural system for discourse segmentation. SEGBOT uses distributed representations to better capture lexical semantics, and employs a bidirectional RNN to model sequential dependencies while encoding the given text. The decoder, which is a unidirectional RNN, uses a *pointer* mechanism [Vinyals *et al.*, 2015; Li *et al.*, 2018b] to infer the segment boundaries. SEGBOT effectively handles variable size vocabulary in the output, to produce segment boundaries depending on the input sequence. Experiments show that SEGBOT outperforms state-of-the-art results on discourse segmentation.

Note that SEGBOT is a generic neural segmenter, and it can also be applied to topic segmentation (*i.e.,* breaking a document into a sequence of topically coherent segments). In this demonstration, we focus on EDU segmentation.

2 The SEGBOT Model

Encoding Phase. For the task of discourse segmentation, the units in the input ($U0$ to $U8$ in Figure 2(a)) are words in a sentence. Each word is represented with a distributed representation. We use the pretrained word vectors from GloVe [Pennington *et al.*, 2014], which are validated on various NLP tasks including text classification [Iyyer *et al.*, 2015] and reading comprehension [Wang *et al.*, 2017].

[1] The online demonstration of SEGBOT can be accessed at http://138.197.118.157:8000/segbot/



Sheraton and Pan Am said they are assured under the Soviet joint-venture law that they can repatriate profits from their hotel venture.

Sheraton and Pan Am said they are assured under the Soviet joint-venture law that they can repatriate profits from their hotel venture .

List of Elementary Discourse Units:

1. Sheraton and Pan Am said
2. they are assured under the Soviet joint-venture law
3. that they can repatriate profits from their hotel venture .

(a) The SEGBOT model

(b) The Web interface: http://138.197.118.157:8000/segbot/

Figure 2: The SEGBOT model architecture, and the Web interface for demonstration (input panel on the left and output panel on the right).

Formally, given an input sequence $\mathbf{U} = (U_1, U_2, \ldots, U_N)$ of length N, we get its distributed representations $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ by looking up the corresponding embedding matrix, where $\boldsymbol{x}_n \in \mathbb{R}^K$ is the representation for the unit U_n with K being the dimensions. Our ultimate goal is to split the input sequence into contiguous segments by identifying the boundaries (*e.g.,* $U3$, $U5$ and $U10$ in Figure 2(a)).

SEGBOT encodes the input sequence $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ by using bidirectional gated recurrent units (GRUs) [Cho *et al.*, 2014], which are similar to long short-term memory (LSTM) but are computationally cheaper. GRUs are able to capture long distance dependencies without running into the problems of gradient vanishing or explosion.

Decoding Phase. The decoder of SEGBOT takes a start unit (*i.e.,* start of a segment) U_m in the input sequence as input and transforms it to its distributed representation $\boldsymbol{x}_m$ by looking up the corresponding embedding matrix. It then passes $\boldsymbol{x}_m$ through a GRU-based (unidirectional) hidden layer. We use "*teacher forcing*" [Lamb *et al.*, 2016] to train our model by supplying the ground-truth start units to decoder RNNs.

Pointing Phase. The output layer of our decoder computes a distribution over the possible positions in the input sequence for a possible segment boundary. For example, considering Figure 2(a), as the decoder starts with input $U1$, it computes an output distribution over all positions ($U1$ to $U10$) in the input sequence. Then, for $U4$ as input, it computes an output distribution over positions $U4$ to $U10$, and finally for $U6$ as input, it computes a distribution over $U6$ to $U10$.

3 Interface and Demonstration

We demonstrate the effectiveness of SEGBOT for discourse segmentation by developing a concise web interface, as shown in Figure 2(b). The interface consists of two panels: the input panel and the output panel.

The input panel takes sentences from users. Once a user clicks the button labeled with a right arrow, the input sentence is passed to the SEGBOT model, where each word in the sentence is an input unit (*e.g.,* $U0$ to $U8$ in Figure 2(a)).

The output panel displays the segmentation results in two forms. As shown in the output panel in Figure 2(b), on the top is the color-coded sentence where each EDU is displayed in a different color. The color-coded sentence presents EDUs in its context to facilitate easy interpretation for the user. On the bottom of the output panel, is the list of segmented EDUs, as shown in Figure 2(b). The order of the EDUs are determined by their positions in the original sentence. The list of EDUs allows user to consider individual EDUs and to pay more attention to their boundaries.

Method	Precision	Recall	F-score
HILDA [Hernault *et al.*, 2010]	77.9	70.6	74.1
SPADE [Soricut and Marcu, 2003]	83.8	86.8	85.2
F&R [Fisher and Roark, 2007]	91.3	89.7	90.5
DS [Joty *et al.*, 2015]	88.0	92.3	90.1
BiLSTM-CRF [Lample *et al.*, 2016]	89.1	87.8	88.5
SEGBOT (our model)	**91.6***	**92.8***	**92.2***

Table 1: Segmentation results on RST-DT Dataset. Significant improvements over the underlined methods are marked with * (t-test, p-value < 0.01).

SEGBOT allows all valid English sentences as input. In our demonstration, we provide an example sentence "*Sheraton and Pan Am said they are assured under the Soviet joint-venture law that they can repatriate profits from their hotel venture*" for users to begin with. The three elementary discourse units are displayed in the output panel as shown in Figure 2(b). Clicking the reset button results in clearing the input panel to take in the next sentence.

4 Effectiveness Evaluation

The RST Discourse Treebank (RST-DT) [Carlson *et al.*, 2002] is a publicly available corpus manually annotated with EDUs and relations according to RST. We use GloVe 300-dimensional pre-trained word embeddings released by Stanford, and the word vectors are ketp fixed during training. Following previous work [Hernault *et al.*, 2010; Joty *et al.*, 2015], we measure segmentation performance using Precision, Recall, and F-score, reported in Table 1.

Comparing against six baselines, SEGBOT consistently outperforms all baselines on all measures. The improvement against baselines is from 0.3% to 17.6% on precision, and 0.5%-31.4% on recall, 1.8%-24.4% on F-score. It is worth mentioning that SEGBOT does not require any feature engineering. Simply taking pre-trained word embeddings as input, SEGBOT outperforms all models with carefully designed features, such as HILDA, SPADE, F&R and DS. The state-of-the-art neural model (BiLSTM-CRF) takes the same input as our model, *i.e.,* pre-trained word embeddings. SEGBOT beats BiLSTM-CRF with an absolute F-score improvement of 4.2% (p-value < 0.01).

References

[Carlson *et al.*, 2002] Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania, 2002.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Dias *et al.*, 2007] Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. Topic segmentation algorithms for text summarization and passage retrieval: an exhaustive evaluation. In *AAAI*, pages 1334–1339, 2007.

[Durrett *et al.*, 2016] Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Fisher and Roark, 2007] Seeger Fisher and Brian Roark. The utility of parse-derived features for automatic discourse segmentation. In *ACL*, volume 45, page 488, 2007.

[Hernault *et al.*, 2010] Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. A sequential model for discourse segmentation. In *CICLing*, volume 6008, pages 315–326, 2010.

[Iyyer *et al.*, 2015] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *ACL*, pages 1681–1691, 2015.

[Joty *et al.*, 2015] Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 2015.

[Joty *et al.*, 2017] Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. Discourse Structure in Machine Translation Evaluation. *Computational Linguistics*, 43:4:683–722, 2017.

[Lamb *et al.*, 2016] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*, pages 4601–4609, 2016.

[Lample *et al.*, 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[Li *et al.*, 2018a] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.

[Li *et al.*, 2018b] Jing Li, Aixin Sun, and Shafiq Joty. Segbot: A generic neural text segmentation model with pointer network. In *IJCAI*, pages 4166–4172, 2018.

[Li *et al.*, 2018c] Jing Li, Aixin Sun, and Zhenchang Xing. Learning to answer programming questions with software documentation through social context embedding. *Information Sciences*, 448:36–52, 2018.

[Li *et al.*, 2018d] Jing Li, Aixin Sun, and Zhenchang Xing. To do or not to do: Distill crowdsourced negative caveats to augment api documentation. *Journal of the Association for Information Science and Technology*, 69(12):1460–1475, 2018.

[Li *et al.*, 2018e] Jing Li, Zhenchang Xing, and Aixin Sun. Linklive: discovering web learning resources for developers from q&a discussions. *World Wide Web*, pages 1–27, 2018.

[Mann and Thompson, 1988] W. Mann and S. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[Marcu, 2000] D. Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA, 2000.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Soricut and Marcu, 2003] Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *NAACL*, pages 149–156, 2003.

[Sporleder and Lapata, 2005] Caroline Sporleder and Mirella Lapata. Discourse Chunking and its Application to Sentence Compression. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT-EMNLP'05, pages 257–264, Vancouver, British Columbia, Canada, 2005. ACL.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[Vinyals *et al.*, 2015] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NIPS*, pages 2692–2700, 2015.

[Wang *et al.*, 2017] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, pages 189–198, 2017.

[Xu *et al.*, 2019] Canwen Xu, Jing Li, Xiangyang Luo, Jiaxin Pei, Chenliang Li, and Donghong Ji. Dlocrl: A deep learning pipeline for fine-grained location recognition and linking in tweets. *arXiv preprint arXiv:1901.07005*, 2019.