Hierarchical RNN with Static Sentence-Level Attention for Text-Based Speaker Change Detection

Zhao Meng
Key Laboratory of High Confidence Software Technologies, MoE
Software Institute, Peking University
zhaomeng.pku@outlook.com

Lili Mou
David R. Cheriton School of Computer Science
University of Waterloo
doublepower.mou@gmail.com

Zhi Jin*
Key Laboratory of High Confidence Software Technologies, MoE
Software Institute, Peking University
zhijin@sei.pku.edu.cn

ABSTRACT

Speaker change detection (SCD) is an important task in dialog modeling. Our paper addresses the problem of text-based SCD, which differs from existing audio-based studies and is useful in various scenarios, for example, processing dialog transcripts where speaker identities are missing (e.g., OpenSubtitle), and enhancing audio SCD with textual information. We formulate text-based SCD as a matching problem of utterances before and after a certain decision point; we propose a hierarchical recurrent neural network (RNN) with static sentence-level attention. Experimental results show that neural networks consistently achieve better performance than feature-based approaches, and that our attention-based model significantly outperforms non-attention neural networks.[1]

1 INTRODUCTION

Speaker change detection (SCD), or sometimes known as *speaker segmentation*, aims to find changing points of speakers in a dialog. Specifically, a speaker change occurs when the current and the next sentences are not uttered by the same speaker [10]. Detecting speaker changes plays an important role in dialog processing, and is a premise of dialog understanding, speaker clustering [14], etc.

In this paper, we address the problem of text-based speaker change detection, which differs from traditional SCD with audio input [4, 9]. Text-based SCD is important for several reasons:

• Evidence in the speech processing domain shows that text information can improve speech-based SCD [7]. However, there lacks specialized research for text-based SCD.

• In some scenarios, researchers may not have access to raw audio signals for SCD. Vinyals et al. [17] and Li et al. [5], for example, train sequence-to-sequence neural networks to automatically generate replies in an open-domain dialog system. They use OpenSubtitle [16] as the corpus, but assume every two consecutive sentences are uttered by different speakers (which brings much noise to their training data).

• The fast development of dialog analysis puts high demands on understanding textual data [17–19]—in addition to audio features alone—because human-computer conversation involves deep semantics, requiring complicated natural language processing. Text-based SCD could also serve as a surrogate task for general speaker modeling, similar to next utterance classification (NUC) being a surrogate task for general dialog generation [8].

Using only text to detect speaker changes brings new challenges. Previous audio-based SCD depends largely on acoustic features, e.g., pitch [9] and silence points [4], which provide much information of speaker changes. With textual features alone, we need deeper semantic understanding of natural language utterances.

In this paper, we formulate text-based SCD as a binary sentence-pair classification problem, that is, we would like to judge whether the speaker is changing between each consecutive sentence pair (which we call a *decision point*). We also take into consideration previous and future sentences around the current decision point as context (Figure 1), serving as additional evidence.[2]

We propose a hierarchical RNN with static sentence-level attention for text-based speaker change detection. First, we use a long short term memory (LSTM)-based recurrent neural network (RNN) to capture the meaning of each sentence. Another LSTM-RNN integrates sentence information into a vector, before and after the decision point, respectively; the two vectors are combined for prediction. To better explore the context, we further apply an attention mechanism over sentences to focus on relevant information during context integration. Compared with widely-used word-level attention, our sentence-level attention is more efficient because there could be hundreds of words in the context within only a few sentences. Also, our attention is static in that only the nearest two sentences around the decision point search for relevant information; it differs from dynamic attention [1], which buries more important sentences under less important context.

Our model was evaluated on transcripts of nearly 3,000 episodes of TV talk shows. In our experiments, modern neural networks consistently outperform traditional methods that use handcrafted features. Ablation tests confirm the effectiveness of context; the proposed hierarchical RNN with sentence-level static attention can better utilize such contextual information, and significantly outperforms non-attention neural networks. The results show that our tailored model is especially suited to the task of text-based speaker change detection.

*Corresponding author. This research is supported by the National Basic Research Program of China (the 973 Program) under Grant No. 2015CB352201, and the National Natural Science Foundation of China under Grant Nos. 61232015 and 61620106007.
[1]Code available at https://sites.google.com/site/textscd/

[2]Because our task is based on text (and is *not* online speaker change detection), we actually have access to the "future context" after the decision point.

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133110

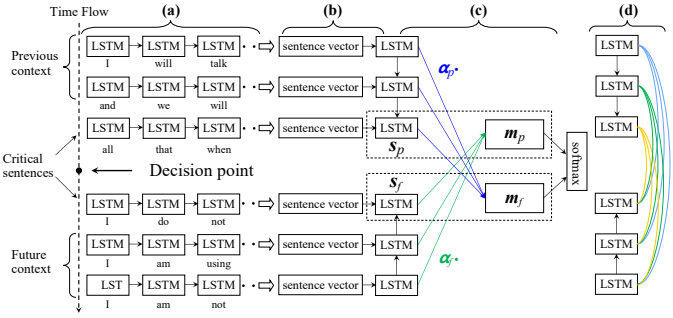

Figure 1: The proposed neural network. (a) LSTM-RNN sentence encoder. (b) Context encoder with another LSTM-RNN. (c) Sentence-level static attention. (d) C.f. Dynamic attention. Notice that we consider text-based speaker change detection in this paper, so we have future utterances as context.

2 RELATED WORK

Traditional speaker change detection (SCD) deals with audio input and is a key step for *speaker diarization* (determining "who spoke when?") [10]. A typical approach is to compare consecutive sliding windows of input with spectral features, pitch, or silence points [4, 9]. Our paper differs from the above work and focuses on textual input, which is useful in various scenarios, for example, processing dialog transcripts where speaker identities are missing (e.g., OpenSubtitle) [5], and enhancing audio SCD with textual information [7].

Nowadays, text-based dialog analysis has been increasingly important, as surface acoustic features are insufficient for semantic understanding in conversations. Previous research has addressed a variety of tasks, ranging from dialog act classification [13] to user intent modeling [3]. In our previous study, we address the problem of session segmentation in text-based human-computer conversations [15]. Without enough annotated data, we apply a heuristic matching approach, thus the task being unsupervised. Li et al. [7] enhance audio-based SCD with transcribed text, and they are also in the unsupervised regime. By contrast, this paper adopts a supervised setting as we have obtained massive, high-quality labels of speaker identities from the Cable News Network website.

As described in Section 1, we formulate our task as a sentence-pair classification problem. Previous studies have utilized convolutional/recurrent neural networks (CNNs/RNNs) to detect the relationship (e.g., paraphrase and logical entailment) between two sentences [2]; Rocktäschel et al. [11] equip RNN with attention mechanisms. These studies do not consider contextual information. In our scenario, the context appears on both sides of the decision point, and we carefully design the neural architecture to better use such contextual information.

3 APPROACH

In this section, we describe the proposed approach in detail. Figure 1 shows the overall architecture of our model, which has three main components: a sentence encoder, a context encoder, and an attention-based matching mechanism.

3.1 Sentence Encoder

We use a recurrent neural network (RNN) with long short term memory (LSTM) units to encode a sentence as a vector (also known as a *sentence embedding*), shown in Figure 1a.

An RNN is suited for processing sequential data (e.g., a sentence consisting of several words) as it keeps a hidden state, changing at each time step based on its previous state and the current input. But vanilla RNNs with perceptron-like hidden states suffer from the problem of *vanishing or exploding gradients*, being less effective to model long dependencies. LSTM units alleviate the problem by better balancing input and its previous state with gating mechanisms. For convenience, we use LSTM's final state (corresponding to the last word in a sentence) as the sentence embedding.

Formally, let $\boldsymbol{x}^{(t)}$ be the embedding of the t-th word in a sentence, and $\boldsymbol{h}^{(t-1)}$ be the last step's hidden state. We have

$$[\boldsymbol{i}^{(t)}; \boldsymbol{f}^{(t)}; \boldsymbol{o}^{(t)}] = \sigma(W\boldsymbol{x}^{(t)} + U\boldsymbol{h}^{(t-1)} + \boldsymbol{b}) \quad (1)$$

$$\boldsymbol{g}^{(t)} = \tanh(W_g\boldsymbol{x}^{(t)} + U_g\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_g) \quad (2)$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{i}^{(t)} \otimes \boldsymbol{g}^{(t)} + \boldsymbol{c}^{(t-1)} \otimes \boldsymbol{f}^{(t)} \quad (3)$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \otimes \tanh(\boldsymbol{c}^{(t)}) \quad (4)$$

where W's and U's are weights, and $\boldsymbol{b}$'s are bias terms. $\otimes$ denotes element-wise product, σ the sigmoid function. $\boldsymbol{i}^{(t)}$, $\boldsymbol{f}^{(t)}$, and $\boldsymbol{o}^{(t)}$ are known as gates, and $\boldsymbol{h}^{(t)}$ is the current step's hidden state.

3.2 Context Encoder

Another LSTM-RNN encodes contextual information over sentence vectors, shown in Figure 1b. Since we model our task as a matching problem, we apply the RNN to the decision point's both sides separately, the resulting vectors of which are concatenated as an input of prediction.

It should be noticed that our LSTM-RNN goes from faraway sentences to the nearest ones (called *critical sentences*) on both sides of the current decision point. We observe that nearer sentences play a more important role for prediction; that an RNN is better at keeping recent input information by its nature. Hence, our treatment is appropriate.

Besides, our neural network is hierarchical in that it composites sentences with words, and discourses with sentences. It is similar to the hierarchical autoencoder in [6]. Other studies apply a single RNN over a discourse, also achieving high performance in tasks like machine comprehension [12]. In our scenario, however, the sentences (either context or critical ones) are not necessarily uttered by a single speaker. Experiments in Section 4.3 show that hierarchical models are more suitable for SCD.

3.3 Our Attention Mechanism

We use an attention mechanism to better utilize contextual information. Attention-based neural networks are first proposed to dynamically focus on relevant words of the source sentence in machine translation [1]. In our scenario, we would like to match a critical sentence with all utterances on the other side of the decision point (Figure 1c). That is to say, the attention mechanism is applied to the sentence level, different from other work that uses word-level attention. Our method is substantially more efficient

because a context of several utterances could contain hundreds of words.

Considering t sentences (context size being $t-1$) before and after the decision point, respectively, we have $2t$ sentences in total, namely $\boldsymbol{s}_p^{(1)} \rightarrow \boldsymbol{s}_p^{(2)} \rightarrow \cdots \rightarrow \boldsymbol{s}_p^{(t)}$ $\boldsymbol{s}_f^{(t)} \leftarrow \cdots \leftarrow \boldsymbol{s}_f^{(2)} \leftarrow \boldsymbol{s}_f^{(1)}$, where subscripts p and f refer to previous and future utterances around the current decision point; the arrows indicate RNN's directions.

In our attention mechanism, a critical sentence, e.g., $\boldsymbol{s}_p^{(t)}$, focuses on all sentences on the other side of the decision point $\boldsymbol{s}_f^{(1)}, \cdots, \boldsymbol{s}_f^{(t)}$, and aggregates information weighted by a probabilistic distribution $\boldsymbol{\alpha}_p \in \mathbb{R}^t$, i.e.,

$$\tilde{\alpha}_p^{(i)} = \boldsymbol{u}_a^\top \tanh\left(W_a\left[\boldsymbol{s}_p^{(t)}; \boldsymbol{s}_f^{(i)}\right]\right) \tag{5}$$

$$\alpha_p^{(i)} = \text{softmax}\left(\tilde{\alpha}_p^{(i)}\right) = \frac{\exp\left\{\tilde{\alpha}_p^{(i)}\right\}}{\sum_{j=1}^{t}\exp\left\{\tilde{\alpha}_p^{(j)}\right\}} \tag{6}$$

Here, $\boldsymbol{s}_p^{(t)}$ is concatenated with $\boldsymbol{s}_f^{(i)}$, processed by a two-layer perceptron (with parameters W_a and $\boldsymbol{u}_a$). $\tilde{\alpha}_i$ is a real-valued measure, normalized by softmax to give the probability α_i. The aggregated information, known as an *attention vector*, is

$$\boldsymbol{m}_p = \sum_{i=1}^{t} \alpha_p^{(i)} \cdot \boldsymbol{s}_f^{(i)} \tag{7}$$

Likewise, the other critical sentence $\boldsymbol{s}_f^{(t)}$ yields an attention vector $\boldsymbol{m}_f$. They are concatenated along with LSTM's output of the critical sentences for prediction. In other words, the input of softmax is $[\boldsymbol{s}_p^{(t)}; \boldsymbol{s}_f^{(t)}; \boldsymbol{m}_p; \boldsymbol{m}_f]$.

It should be pointed out that, our attention is static, as only critical sentences search for relevant information using Equations (5)–(7). It resembles a variant in [11], but differs from common attention where two LSTMs interact dynamically along their propagations (Figure 1d). Such approach may bury the critical sentences; it leads to slight performance degradation in our experiment, as we shall see in Section 4.3.

4 EXPERIMENTS

4.1 Dataset Collection

We crawled transcripts of nearly 3,000 TV talk shows from the Cable News Network (CNN) website,[3] and extracted main contents from the original html files. The transcripts contain speaker identities, with which we induced speaker changes in the dialog.

The crawled dataset comprises 1.5M utterances. We split training, validation, and test sets by episodes (TV shows) at a ratio of 8:1:1. In other words, each episode appears in either the training set, or the val/test sets. This prevents utterance overlapping between training and prediction, and thus is a more realistic setting than splitting by utterances.

We notice that, our corpus is larger than previous ones by magnitudes: the 1997 HUB4 dataset, for example, is 97 hours long, whereas our TV shows are estimated to be 3,000 hours (each episode roughly lasting for an hour), which are more suitable for training deep neural networks.

[3] http://transcripts.cnn.com (*CNN* here should not be confused with a convolutional neural network.)

Model	Acc.	F_1	P	R
Random guess	61.8	25.4	26.0	25.0
Logistic regression w/ (uni+bi)-gram	80.5	50.9	73.0	39.0
DNN w/ (uni+bi)-gram	76.6	56.5	54.4	58.8
CNN w/o context	77.8	57.8	56.8	58.9
RNN w/o context	83.3	63.9	72.5	57.1
RNN w/ context (non-hierarchical)	83.7	65.7	72.6	60.0
RNN w/ context (hierarchical)	85.1	69.2	74.6	64.6
+ static attention	**89.2**	**78.4**	**81.5**	**75.6**

Table 1: Model performance (in %). Here, LSTM units are used in RNN, but omitted in the table for brevity. The context size is 8, chosen by validation (deferred to Figure 2).

In the dataset, speaker changes count to 25%. We thus used F_1-measure in addition to accuracy as metrics, i.e., $F_1 = 2P \cdot R/(P+R)$, where $P = \frac{\#\text{correctly detected changes}}{\#\text{detected changes}}$ is the *precision*, and $R = \frac{\#\text{correctly detected changes}}{\#\text{all changes}}$ is the *recall*.

4.2 Settings

We set all neural layers, including word embeddings, to 200 dimensional. Since our dataset is large, we randomly initialized word embeddings, which were tuned during training. We used the Adam optimizer with mini-batch update (batch size being 100). Other hyperparameters were chosen by validation: dropout rate from $\{0.1, 0.3\}$ and initial learning rate from $\{3\times10^{-4}, 9\times10^{-4}\}$.

We had several baselines with handcrafted features: we extracted unigram and bigram features of the critical sentences as two vectors, which are concatenated for prediction. We applied logistic regression and a 3-layer deep neural network (DNN) as the classifier; the former is a linear model whereas the latter is nonlinear. DNN's hidden dimension was set to 200, which is the same as our neural network.

A convolutional neural network (CNN) is also included for comparison. It adopts a window size of 3, and a max-pooling layer aggregates extracted features. All competing neural models (including DNN and CNN) were tuned in the same manner, so our comparison is fair.

4.3 Performance

Table 1 presents the performance of our model as well as baselines. As shown, all modern neural networks (CNNs/RNNs with word embeddings) are consistently better than methods using handcrafted features of unigrams and bigrams. Because we have applied a 3-layer DNN to these features, we believe the performance improvement is not merely caused by using a better classifier, but the automatic feature/representation learning nature of modern neural networks.

For neural network-based sentence encoders, we compared LSTM-RNN with CNN. Results show that LSTM-RNN outperforms CNN by 6% in terms of both accuracy and F_1-measure.

To cope with context, the simplest approach, perhaps, is to use an RNN to go through surrounding utterances of the critical sentences,

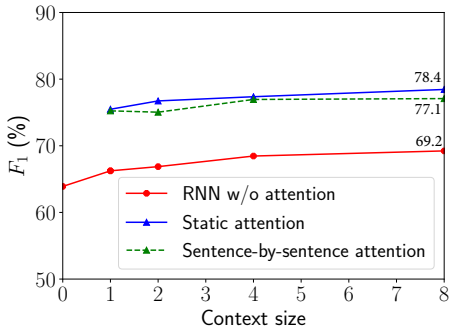

Figure 2: Effect of the context size and different attention mechanisms. The context size is the number of utterances, excluding the critical sentence, on each side of the decision point; a size of 0 refers to no context.

denoted as *non-hierarchical* in Table 1. Using contextual information yields an F_1 improvement of 2%. This controlled experiment validates the usefulness of context for SCD. The hierarchical RNN introduced in Sections 3.1 and 3.2 further improves the F_1-measure by 3%. With sentence-level static attention, our model achieves the highest performance of 89.2% accuracy and 78.4% F_1-measure.

We would like to have in-depth analysis regarding how the context size and different attention mechanisms affect our model. The context size was chosen by validation from $\{1, 2, 4, 8\}$.[4] As shown in Figure 2, even a single context sentence (on each side of the decision point) improves the performance by 2%; with more surrounding utterances, the performance grows gradually. Moreover, attention-based neural networks significantly outperform non-attention models by a margin of 10%. We also tried a dynamic sentence-by-sentence attention mechanism, similar to most existing work [1]. As analyzed in Section 3.3, such model buries critical sentences and thus slightly hurts the performance by 1–2% F_1-measure (green dashed line in Figure 2). The experiments verify the effectiveness of our hierarchical RNN with sentence-level, static attention.

Case study. Table 2 showcases a dialog snippet. In the example, our model makes an error as it fails to detect the change between Sentences 4–5. However, it is even hard for humans to judge this particular change, because the word *absolutely* also goes fluently into the next sentence. For other utterances with more substance, the neural network correctly joins Sentences 1–2 and 5–6, as well as segments Sentences 2–3 and 3–4, showing that our proposed model can effectively capture the semantics of these sentences.

5 CONCLUSION

In this paper, we proposed a static sentence-level attention LSTM-RNN for text-based speaker change detection. Our model uses an LSTM-RNN to encode each utterance into a vector, based on which another LSTM-RNN integrates contextual information, before and after a particular decision point, respectively. A static sentence-level attention mechanism is also applied to enhance information interaction. We crawled dialog transcripts from Cable News Network

[4]Due to efficiency concerns, we did not try larger context sizes.

ID	Utterances	Speaker Changes? Predicted	Truth	Correct?
1	there 's no question the deficit halts on both sides of the aisles .	No	No	✓
2	cbo , wall street , everyone will have a say into this , including workers and future retirees .	Yes	Yes	✓
3	and your saying it 's both parties ?	Yes	Yes	✓
4	absolutely .	No	Yes	✗
5	the key , though , is the first six months .	No	No	✓
6	you say it 's not going to get through in these first couple months .			

Table 2: Case study. Colors indicate speaker identities (they are used to infer groundtruth speaker changes, but cannot be seen during prediction).

TV talk shows for evaluation. Experimental results demonstrate the effectiveness of our approach. In particular, in-depth analysis validates that contextual information is indeed helpful for speaker change detection, and that our tailored model can make better use of context than other neural networks.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[2] S. Bowman, G. Angeli, C. Potts, and C. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. 632–642.
[3] Y. Chen, M. Sun, A. I Rudnicky, and A. Gershman. 2016. Unsupervised user intent modeling by feature-enriched matrix factorization. In *ICASSP*. 6150–6154.
[4] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel. 2000. Strategies for automatic segmentation of audio data. In *ICASSP*. 1423–1426.
[5] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL*. 110–119.
[6] J. Li, T. Luong, and D. Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *ACL-IJCNLP*. 1106–1115.
[7] R. Li, T. Schultz, and Q. Jin. 2009. Improving speaker segmentation via speaker identification and text segmentation.. In *INTERSPEECH*. 904–907.
[8] R. Lowe, N. Pow, L. Serban, and J. Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL*. 285–294.
[9] L. Lu and H. Zhang. 2005. Unsupervised speaker segmentation and tracking in real-time audio content analysis. *Multimedia Systems* 10, 4 (2005).
[10] Anguera M., S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. 2012. Speaker diarization: A review of recent research. *TASLP* 20, 2 (2012), 356–370.
[11] T. Rocktäschel, E. Grefenstette, M. Hermann, T. Kočiskỳ, and P. Blunsom. 2016. Reasoning about entailment with neural attention. In *ICLR*.
[12] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.
[13] S. Shen and H. Lee. 2016. Neural Attention Models for Sequence Classification: Analysis and Application to Key Term Extraction and Dialogue Act Detection. In *INTERSPEECH*. 2716–2720.
[14] R. Sinha, S. Tranter, M. Gales, and P. Woodland. 2005. The Cambridge University March 2005 speaker diarisation system. In *INTERSPEECH*. 2437–2440.
[15] Y. Song, L. Mou, R. Yan, L. Yi, Z. Zhu, X. Hu, and M. Zhang. 2016. Dialogue session segmentation by embedding-enhanced TextTiling. In *INTERSPEECH*. 2706–2710.
[16] J. Tiedemann. 2009. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *RANLP*. 237–248.
[17] O. Vinyals and Q. Le. 2015. A neural conversational model. In *ICML Workshop*.
[18] P. Wang, L. Ji, J. Yan, L. Jin, and W. Ma. 2016. Learning to extract conditional knowledge for question answering using dialogue. In *CIKM*. 277–286.
[19] R. Yan, Y. Song, X. Zhou, and H. Wu. 2016. Shall I be your chat companion? Towards an online human-computer conversation system. In *CIKM*. 649–658.