Dialogue Session Segmentation by Embedding-Enhanced TextTiling

Abstract

In human-computer conversation systems, the context of a user-issued utterance is particularly important because it provides useful background information of the conversation. However, it is unwise to track all previous utterances in the current session as not all of them are equally important. In this paper, we address the problem of session segmentation. We propose an embedding-enhanced TextTiling approach, inspired by the observation that conversation utterances are highly noisy, and that word embeddings provide a robust way of capturing semantics. Experimental results show that our approach achieves better performance than the TextTiling, MMD approaches.

Introduction

Human-computer dialog/conversation is one of the most challenging problems in artificial intelligence. Given a user-issued utterance (called a query in this paper), the computer needs to provide a reply to the query. In early years, researchers have developed various domain-oriented dialogue systems, which are typically based on rules or templates BIBREF4 , BIBREF5 , BIBREF6 . Recently, open-domain conversation systems have attracted more and more attention in both academia and industry (e.g., XiaoBing from Microsoft and DuMi from Baidu). Due to high diversity, we can hardly design rules or templates in the open domain. Researchers have proposed information retrieval methods BIBREF7 and modern generative neural networks BIBREF8 , BIBREF9 to either search for a reply from a large conversation corpus or generate a new sentence as the reply.

In open-domain conversations, context information (one or a few previous utterances) is particularly important to language understanding BIBREF1 , BIBREF9 , BIBREF10 , BIBREF11 . As dialogue sentences are usually casual and short, a single utterance (e.g., “Thank you.” in Figure FIGREF2) does not convey much meaning, but its previous utterance (“...writing an essay”) provides useful background information of the conversation. Using such context will certainly benefit the conversation system.

However, tracking all previous utterances as the context is unwise. First, commercial chat-bots usually place high demands on efficiency. In a retrieval-based system, for example, performing a standard process of candidate retrieval and re-ranking for each previous utterance may well exceed the time limit (which is very short, e.g., 500ms). Second, we observe that not all sentences in the current conversation session are equally important. The sentence “Want to take a walk?” is irrelevant to the current context, and should not be considered when the computer synthesizes the reply. Therefore, it raises the question of session segmentation in conversation systems.

Document segmentation for general-purpose corpora has been widely studied in NLP. For example, Hearst BIBREF12 proposes the TextTiling approach; she measures the similarity of neighboring sentences based on bag-of-words features, and performs segmentation by thresholding. However, such approaches are not tailored to the dialogue genre and may not be suitable for conversation session segmentation.

In this paper, we address the problem of session segmentation for open-domain conversations. We leverage the classic TextTiling approach, but enhance it with modern embedding-based similarity measures. Compared with traditional bag-of-words features, embeddings map discrete words to real-valued vectors, capturing underlying meanings in a continuous vector space; hence, it is more robust for noisy conversation corpora. Further, we propose a tailored method for word embedding learning. In traditional word embedding learning, the interaction between two words in a query and a reply is weaker than that within an utterance. We propose to combine a query and its corresponding reply as a “virtual sentence,” so that it provides a better way of modeling utterances between two agents.

Dialogue Systems and Context Modeling

Human-computer dialogue systems can be roughly divided into several categories. Template- and rule-based systems are mainly designed for certain domains BIBREF4 , BIBREF5 , BIBREF13 . Although manually engineered templates can also be applied in the open domain like BIBREF14 , but their generated sentences are subject to 7 predefined forms, and hence are highly restricted. Retrieval methods search for a candidate reply from a large conversation corpus given a user-issued utterance as a query BIBREF7 . Generative methods can synthesize new replies by statistical machine translation BIBREF15 , BIBREF16 or neural networks BIBREF8 .

The above studies do not consider context information in reply retrieval or generation. However, recent research shows that previous utterances in a conversation session are important because they capture rich background information. Sordoni et al. BIBREF11 summarize a single previous sentence as bag-of-words features, which are fed to a recurrent neural network for reply generation. Serban et al. BIBREF17 design an attention-based neural network over all previous conversation turns/rounds, but this could be inefficient if a session lasts long in real commercial applications. By contrast, our paper addresses the problem of session segmentation so as to retain near, relevant context utterances and to eliminate far, irrelevant ones.

A similar (but different) research problem is topic tracking in conversations, e.g., BIBREF18 , BIBREF19 , BIBREF20 , BIBREF21 . In these approaches, the goal is typically a classification problem with a few pre-defined conversation states/topics, and hence it can hardly be generalized to general-purpose session segmentation.

Text Segmentation

An early and classic work on text segmentation is TextTiling, proposed in BIBREF12 . The idea is to measure the similarity between two successive sentences with smoothing techniques; then segmentation is accomplished by thresholding of the depth of a “valley.” In the original form of TextTiling, the cosine of term frequency features is used as the similarity measure. Joty et al. BIBREF22 apply divisive clustering instead of thresholding for segmentation. Malioutov et al. BIBREF23 formalize segmentation as a graph-partitioning problem and propose a minimum cut model based on tf INLINEFORM0 idf features to segment lectures. Ye et al. BIBREF24 minimize between-segment similarity while maximizing within-segment similarity. However, the above complicated approaches are known as global methods: when we perform segmentation between two successive sentences, future context information is needed. Therefore, they are inapplicable to real-time chat-bots, where conversation utterances can be viewed as streaming data.

In our study, we prefer the simple yet effective TextTiling approach for open-domain dialogue session segmentation, but enhance it with modern advances of word embeddings, which are robust in capturing semantics of words. We propose a tailored algorithm for word embedding learning by combining a query and context as a “virtual document”; we also propose several heuristics for similarity measuring.

TextTiling

We apply a TextTiling-like algorithm for session segmentation. The original TextTiling is proposed by Hearst BIBREF12 . The main idea is to measure the similarity of each adjacent sentence pair; then “valleys” of similarities are detected for segmentation.

Concretely, the “depth of the valley” is defined by the similarity differences between the peak point in each side and the current position. We may obtain some statistics of depth scores like the mean INLINEFORM0 and standard deviation INLINEFORM1 , and perform segmentation by a cutoff threshold.

where INLINEFORM0 is a hyperparameter adjusting the number of segmentation boundaries; INLINEFORM1 and INLINEFORM2 are the average and standard deviation of depth scores, respectively.

In the scenario of human-computer conversations, we compute the depth solely by the similarity difference between its left peak (previous context) and the current position. This is because we cannot obtain future utterances during online conversation.

Although bag-of-words features work well in the original TextTiling algorithm for general text segmentation, it is not suitable for dialogue segmentation. As argued by Hearst BIBREF12 , text overlap (repetition) between neighboring sentences is a strong hint of semantic coherence, which can be well captured by term frequency or tf INLINEFORM0 idf variants. However, in human-computer conversations, sentences are usually short, noisy, highly diversified, and probably incomplete, which requires a more robust way of similarity measuring. Therefore, we enhance TextTiling with modern word embedding techniques, as will be discussed in the next part.

Learning Word Embeddings

Word embeddings are distributed, real-valued vector representations of discrete words BIBREF25 , BIBREF26 . Compared with one-hot representation, word embeddings are low-dimensional and dense, measuring word meanings in a continuous vector space. Studies show that the offset of two words' embeddings represents a certain relation, e.g., “man” INLINEFORM0 “woman” INLINEFORM1 “king” INLINEFORM2 “queen” BIBREF25 . Hence, it is suitable to use word embeddings to model short and noisy conversation utterances.

To train the embeddings, we adopt the word2vec approach. The idea is to map a word INLINEFORM0 and its context INLINEFORM1 to vectors (INLINEFORM2 and INLINEFORM3). Then we estimate the probability of a word by DISPLAYFORM0

The goal of word embedding learning is to maximize the average probability of all words (suppose we have INLINEFORM0 running words): DISPLAYFORM0

We used hierarchical softmax to approximate the probability.

To model the context, we further adopt the continuous bag-of-words (CBOW) method. The context is defined by the sum of neighboring words' (input) vectors in a fixed-size window (INLINEFORM0 to INLINEFORM1) within a sentence: DISPLAYFORM0

Notice that the context vector INLINEFORM0 in Equation (EQREF12) and the output vector INLINEFORM1 in Equation (EQREF9) are different as suggested in BIBREF25 , BIBREF26 , but the details are beyond the scope of our paper.

Virtual Sentences

In a conversation corpus, successive sentences have a stronger interaction than general texts. For example, in Figure FIGREF2 , the words thank and welcome are strongly correlated, but they hardly appear in the a sentence and thus a same window. Therefore, traditional within-sentence CBOW may not capture the interaction between a query and its corresponding reply.

In this paper, we propose the concept of virtual sentences to learn word embeddings for conversation data. We concatenate a query INLINEFORM0 and its reply INLINEFORM1 as a virtual sentence INLINEFORM2 . We also use all words (other than the current one) in the virtual sentence as context (Figure 2). Formally, the context INLINEFORM3 of the word INLINEFORM4 is given by DISPLAYFORM0

In this way, related words across two successive utterances from different agents can have interaction during word embedding learning. As will be shown in Subsection SECREF22 , virtual sentences yield a higher performance for dialogue segmentation.

Measuring Similarity

In this part, we introduce several heuristics of similarity measuring based on word embeddings. Notice that, we do not leverage supervised learning (e.g., full neural networks for sentence paring BIBREF27 , BIBREF28) to measure similarity, because it is costly to obtain labeled data of high quality.

The simplest approach, perhaps, is to sum over all word embeddings in an utterance as sentence-level features INLINEFORM0 . This heuristic is essentially the sum pooling method widely used in neural networks BIBREF29 , BIBREF30 , BIBREF27 . The cosine measure is used as the similarity score between two utterances INLINEFORM1 and INLINEFORM2 . Let INLINEFORM3 and INLINEFORM4 be their sentence vectors; then we have DISPLAYFORM0

where INLINEFORM0 is the INLINEFORM1 -norm of a vector.

To enhance the interaction between two successive sentences, we propose a more complicated heuristic as follows. Let INLINEFORM0 and INLINEFORM1 be a word in INLINEFORM2 and INLINEFORM3 , respectively. (Embeddings are denoted as bold alphabets.) Suppose further that INLINEFORM4 and INLINEFORM5 are the numbers of words in INLINEFORM6 and INLINEFORM7 . The similarity is given by DISPLAYFORM0

For each word INLINEFORM0 in INLINEFORM1 , our intuition is to find the most related word in INLINEFORM2 , given by the INLINEFORM3 part; their relatedness is also defined by the cosine measure. Then the sentence-level similarity is obtained by the average similarity score of words in INLINEFORM4 . This method is denoted as heuristic-max.

Alternatively, we may substitute the INLINEFORM0 operator in Equation (EQREF16) with INLINEFORM1 , resulting in the heuristic-avg variant, which is equivalent to the average of word-by-word cosine similarity. However, as shown in Subsection SECREF22 , intensive similarity averaging has a “blurring” effect and will lead to significant performance degradation. This also shows that our proposed heuristic-max does capture useful interaction between two successive utterances in a dialogue.

Experiments

In this section, we evaluate our embedding-enhanced TextTiling method as well as the effect of session segmentation. In Subsection SECREF17 , we describe the datasets used in our experiments. Subsection SECREF22 presents the segmentation accuracy of our method and baselines. In Subsection SECREF27 , we show that, with our session segmentation, we can improve the performance of a retrieval-based conversation system.

Dataset

To evaluate the session segmentation method, we used a real-world chatting corpus from DuMi, a state-of-the-practice open-domain conversation system in Chinese. We sampled 200 sessions as our experimental corpus. Session segmentation was manually annotated before experiments, serving as the ground truth. The 200 sessions were randomly split by 1:1 for validation and testing. Notice that, our method does not require labeled training samples; massive data with labels of high quality are quite expensive to obtain.

We also leveraged an unlabeled massive dataset of conversation utterances to train our word embeddings with “virtual sentences.” The dataset was crawled from the Douban forum, containing 3 million utterances and approximately 150,000 unique words (Chinese terms).

Segmentation Performance

We compared our full method (TextTiling with heuristic-max based on embeddings trained by virtual sentences) with several baselines:

Random. We randomly segmented conversation sessions. In this baseline, we were equipped with the prior probability of segmentation.

MMD. We applied the MinMax-Dotplotting (MMD) approach proposed by Ye et al. BIBREF24 . We ran the executable program provided by the authors.

TextTiling w/ tf INLINEFORM0 idf features. We implemented TextTiling ourselves according to BIBREF12 .

We tuned the hyperparameter INLINEFORM0 in Equation ()on the validation set to make the number of segmentation close to that of manual annotation, and reported precision, recall, and the F-score on the test set in Table TABREF18 . As seen, our approach significantly outperforms baselines by a large margin in terms of both precision and recall. Besides, we can see that MMD obtains low performance, which is mainly because the approach cannot be easily adapted to other datasets like short sentences of conversation utterances. In summary, we achieve an INLINEFORM1 -score higher than baseline methods by more than 20%, showing the effectiveness of enhancing TextTiling with modern word embeddings.

We further conducted in-depth analysis of different strategies of training word-embeddings and matching heuristics in Table TABREF21 . For word embeddings, we trained them on the 3M-sentence dataset with three strategies: (1) virtual-sentence context proposed in our paper; (2) within-sentence context, where all words (except the current one) within a sentence (either a query or reply) are regarded as the context; (3) window-based context, which is the original form of BIBREF25 : the context is the words in a window (previous 2 words and future 2 words in the sentence). We observe that our virtual-sentence strategy consistently outperforms the other two in all three matching heuristics. The results suggest that combining a query and a reply does provide more information in learning dialogue-specific word embeddings.

Regarding matching heuristics, we find that in the second and third strategies of training word embeddings, the complicated heuristic-max method yields higher INLINEFORM0 -scores than simple sum pooling by 2–3%. However, for the virtual-sentence strategy, heuristic-max is slightly worse than the sum pooling. (The degradation is only 0.1% and not significant.) This is probably because both heuristic-max and virtual sentences emphasize the rich interaction between a query and its corresponding reply; combining them does not result in further gain.

We also notice that heuristic-avg is worse than other similarity measures. As this method is mathematically equivalent to the average of word-by-word similarity, it may have an undesirable blurring effect.

To sum up, our experiments show that both the proposed embedding learning approach and the similarity heuristic are effective for session segmentation. The embedding-enhanced TextTiling approach largely outperforms baselines.

We conducted an external experiment to show the effect of session segmentation in dialogue systems. We integrated the segmentation mechanism into a state-of-the-practice retrieval-based system and evaluated the results by manual annotation, similar to our previous work BIBREF27 , BIBREF31 , BIBREF32 .

Concretely, we compared our session segmentation with fixed-length context, used in BIBREF11 . That is to say, the competing method always regards two previous utterances as context. We hired three workers to annotate the results with three integer scores (0–2 points, indicating bad, borderline, and good replies, respectively.) We sampled 30 queries from the test set of 100 sessions. For each query, we retrieved 10 candidates and computed p@1 and nDCG scores BIBREF33 (averaged over three annotators). Provided with previous utterances as context, each worker had up to 1000 sentences to read during annotation.

Table TABREF26 presents the results of the dialogue system with session segmentation. As demonstrated, our method outperforms the simple fixed-context approach in terms of both metrics. We computed the inner-annotator agreement: std INLINEFORM0 0.309; 3-discrete-class Fleiss' kappa score INLINEFORM1 0.411, indicating moderate agreement BIBREF34 .

Case Study. We present a case study on our website: https://sites.google.com/site/sessionsegmentation/. From the case study, we see that the proposed approach is able to segment the dialogue session appropriately, so as to better utilize background information from a conversation session.

In this paper, we addressed the problem of session segmentation for open-domain dialogue systems. We proposed an embedding-enhanced TextTiling approach, where we trained embeddings with the novel notion of virtual sentences; we also proposed several heuristics for similarity measure. Experimental results show that both our embedding learning and similarity measuring are effective in session segmentation, and that with our approach, we can improve the performance of a retrieval-based dialogue system.

We thank anonymous reviewers for useful comments and Jingbo Zhu for sharing the MMD executable program. This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Nos. 61272343 and 61472006), the Doctoral Program of Higher Education of China (Grant No. 20130001110032), and the National Basic Research Program (973 Program No. 2014CB340405).