Suggesting Sentences for ESL using Kernel Embeddings

Kent Shioda and **Mamoru Komachi** and **Rue Ikeya** and **Daichi Mochihashi**
{shioda-kent@ed., komachi@}tmu.ac.jp
ikeya@nii.ac.jp, daichi@ism.ac.jp

Abstract

Sentence retrieval is an important NLP application for English as a Second Language (ESL) learners. ESL learners are familiar with web search engines, but generic web search results may not be adequate for composing documents in a specific domain. However, if we build our own search system specialized to a domain, it may be subject to the data sparseness problem. Recently proposed word2vec partially addresses the data sparseness problem, but fails to extract sentences relevant to queries owing to the modeling of the latent intent of the query. Thus, we propose a method of retrieving example sentences using kernel embeddings and N-gram windows. This method implicitly models latent intent of query and sentences, and alleviates the problem of noisy alignment. Our results show that our method achieved higher precision in sentence retrieval for ESL in the domain of a university press release corpus, as compared to a previous unsupervised method used for a semantic textual similarity task.

1 Introduction

Many English writing assistant tools are currently being studied and developed. However, even for advanced ESL learners, it is difficult to write sentences conforming to the styles and expressions in a specific domain. Therefore, it is beneficial for non-native speakers to search for sentences using keywords that the writer aims to use.

However, existing sentence retrieval systems fail to capture the latent intent of query, owing to the modeling of sentences. We address this problem by using a kernel embeddings framework. Kernel embeddings makes it possible to add expression to the query in sentence retrieval by using latent probability distribution. In addition, our method of taking N-gram windows boosts the precision of sentence retrieval by considering words that are highly related to the query.

The main contributions of this study are as follows:

- We propose a novel sentence similarity metric based on kernel embeddings and N-gram windows.
- We build a corpus of university press releases and annotated example sentences for ESL, given a query of two words.
- We show that our proposed method outperforms unsupervised baselines on our dataset.

2 Proposed Method

To address the problem of query intent, we propose a sentence retrieval method that considers the latent distribution of a sentence using kernel embeddings. Our proposed method calculates the similarity between the keywords and the target sentences in a high dimensional space defined by kernels using the latent distribution of the query. In addition, our system only requires several keywords as an input, and finds a relevant sentence based on N-grams in the sentence.

In the following subsections, we first describe how we adopt kernel embeddings for sentence retrieval, and then explain how to incorporate N-gram windows to improve the precision of sentence retrieval.

2.1 Kernel Embeddings

Yoshikawa et al. (2015) proposed a method to calculate the similarity between instances across different domains by embedding all the features of

Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications, pages 64–68,
Taipei, Taiwan, December 1, 2017 ©2017 AFNLP

different domains into a shared latent space. Their approach, which calculates the similarity between instances in shared latent space, employed the kernel embeddings framework of Smola et al. (2007). The kernel embeddings of distributions are used to embed any probability distribution $\mathbb{P}$ on space $\mathcal{X}$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ specified by kernel k, where the mapped distributions of $\mathbb{P}$ are represented as an element in the RKHS.

We extend their methods and apply them to a sentence retrieval task. Our method considers words comprising a query and a sentence as a set of words, and assumes that each word has a latent probability distribution mapped in a shared latent space. In other words, the query and sentence can be expressed as instances in an RKHS. Then, we calculate the similarity between the mapped sets in shared latent space using the kernel embeddings framework. In this paper, we use the word embeddings $\vec{w} \in \mathcal{X}$ trained by word2vec to represent a latent distribution.

The words embeddings $\vec{q_i}$ and $\vec{s_j}$ contained in query q and sentence s are the instances $\mu_{\mathbb{P}}$ on RKHS $\mathcal{H}_k$ determined by kernel k. Note that, in this paper, we treat word vectors as independent and identically distributed (i.i.d.). We show an instance of a query in the following RKHS. Instances of sentences are determined in the same manner.

$$\mu_{\mathbb{P}_q} = \frac{1}{|q|} \sum_{l=1}^{|q|} k(\cdot, \vec{q_l}) \in \mathcal{H}_k \tag{1}$$

Here, we describe a method of measuring similarity between instances mapped on the RKHS. Assuming two sets of i.i.d. samples $X = \{\mathrm{x}_l\}_{l=1}^{n}$ and $Y = \{\mathrm{y}_{l'}\}_{l'=1}^{n'}$ existing in the same space, they are expressed as $\mu_{\mathbb{P}_X}$, $\mu_{\mathbb{P}_Y}$ by kernel embeddings representation. Moreover, the distance between the two distributions $D(X,Y)$ is calculated as follows:

$$D(X,Y) \quad = ||\mu_{\mathbb{P}_X} - \mu_{\mathbb{P}_Y}||^2_{\mathcal{H}_k} \tag{2}$$

Therefore, the similarity sim_{ke} of the query and sentence is calculated by the inner-product $\langle \mu_{\mathbb{P}_q}, \mu_{\mathbb{P}_s} \rangle_{\mathcal{H}_k}$ in the RKHS as follows:

$$\begin{aligned} sim_{ke}(q,s) &= \langle \mu_{\mathbb{P}_q}, \mu_{\mathbb{P}_s} \rangle_{\mathcal{H}_k} \\ &= \frac{1}{|q||s|} \sum_{i=1}^{|q|} \sum_{j=1}^{|s|} k(\vec{q_i}, \vec{s_j}) \end{aligned} \tag{3}$$

Algorithm 1 Calculate Sentence Similarity

```
Input: sentence, query, N
Output: similarity
max_SIM ← 0
for each $N$-gram ($N$= 1, 2, ..., $N$) ∈ sentence do
    SIM ← $sim_{ke}$(query, N-gram)
    if SIM > max_SIM then
        max_SIM ← SIM
    end if
end for
return max_SIM
```

2.2 N-gram Window

The kernel embeddings method is good at improving the recall of keyword-based sentence retrieval. However, owing to the canonical inner-product, it may decrease precision for a long sentence where keywords appear far apart. We propose a simple N-gram based method to overcome this challenge.

The algorithm is shown as Algorithm 1. First, our method delimits sentences by word N-grams. Second, we calculate the similarity between the query and each N-gram in the sentence. Finally, the highest similarity between the query and all N-grams is considered as the sentence similarity.

3 Experiment

3.1 Settings

As the latent vector, we used published word embeddings[1] learned by word2vec on part of a Google News dataset. To tokenize sentences, we used the Stanford Core NLP tokenizer (Ver. 3.6.0)[2]. Tokenized words were changed to lowercase to calculate similarity. We used the following cosine similarity and RBF kernel for k in Equation (3).

$$k_{cos}(q_i, s_j) = \frac{\langle q_i, s_j \rangle}{|q_i||s_j|} \tag{4}$$

$$\begin{aligned} k_{\mathrm{RBF}}(q_i, s_j) &= \exp\left(-\frac{||q_i - s_j||^2}{2\sigma^2}\right) \\ &= \exp\left(-\gamma||q_i - s_j||^2\right) \end{aligned} \tag{5}$$

[1] https://code.google.com/archive/p/word2vec/

[2] http://nlp.stanford.edu/software/stanford-corenlp-full-2015-12-09.zip

Table 1: Example of pairs of query.

education innovative, identify research,
provide advice, plan annual,
recipient award, goal ensure,
partnership support, field industry,
improve success, lead experience

Preliminary experiments were performed using the hyperparameter γ of the RBF kernel within the range of $\gamma \in \{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. We set the hyperparameter γ as $\gamma = 10^1$ based on the results.

3.2 Data

In this study, we experimented on a domain of academic press release articles. We constructed a sentence retrieval dataset for ESL in the following manner.

First, we created a corpus extracted from web pages containing ".edu" at the end of the domain name. We crawled the ".html" files up to three levels within the ".edu" domain, and used the text surrounded by p tags. The resulting corpus contained 579,867 sentences. We crafted 30 queries consisting of two words using a professional annotator, and extracted sentences from the corpus by exact matching of each query. The annotator evaluated whether the sentence was relevant to the query. The results were used as evaluation data.

Second, we picked ten queries with at least ten relevant sentences. As irrelevant sentences, we used 90 sentences that were deemed to be irrelevant by the annotator. When a query had less than 90 irrelevant sentences, we randomly sampled sentences from the evaluation data to increase this to 90 sentences. Note that all the relevant sentences used in this experiment contained two words. The average sentence length in the test data was 30 words. Table 1 lists the ten queries we used in testing.

3.3 Evaluation

We evaluated the result using Precision@k (hereafter, p@k), and compared our model with the following two baselines.

Average similarity. A simple baseline was calculated using the average similarity of the vectors of query and sentence. We used word2vec's word embeddings as the word vector. As the query vector, we averaged the word vectors of the query. Similarly, as the sentence vector, we averaged the word vectors of the words in the sentence. We compared these vectors using cosine similarity and RBF kernel.

Alignment-based similarity. As another baseline, we used one of the unsupervised sentence similarity measures proposed by Song and Roth (2015). These methods achieved state-of-the-art performance for a short text similarity (STS) task. We used their method to calculate the inter-sentence similarity (maximum alignment) based on the alignment in the distributed representation expressed by the following equation.

$$sim_{max}(q, s) = \frac{1}{|q|} \sum_{i=1}^{|q|} \max_j k(q_i, s_j) \tag{6}$$

This method calculates the maximum value of similarity between each keyword q_i of the query q and each word s_j included in the sentence s. Then, the similarity between the query and the sentence is calculated as the maximum value divided by the number of keywords $|q|$. We experimented with both cosine similarity and RBF kernel for k in Equation (6). Note that we did not symmetrize Equation (6).

3.4 Result

We show the results of the experiment in Figures 1 and 2. We calculated p@k from 1-gram to 40-grams and plotted the results of N-grams with an increment of ten, in addition to 1-gram to 5-gram. "Sentence" in figures refers to similarity based on all words in the sentence.

Figure 1 shows that when cosine similarity was used for the kernel, it was better not to use kernel embeddings. Further, alignment-based similarity is the most effective method, with the exception of the highest ranking. In this case, alignment-based similarity was better than almost all of the N-grams.

Figure 2 demonstrates that the accuracy of the RBF kernel increases with the incorporation of an N-gram window. In addition, the best result in the top-5 ranking was obtained by using RBF kernels and longer N-grams. These results indicate that the most effective RBF kernels have window sizes of 20-gram.

However, we observed that sizes of 1-gram to 3-gram produced a negative result for the RBF kernel. In the next section, we discuss why lower order N-grams led to negative results.

Table 2: Examples of a retrieved relevant sentence using a 19-gram, and an irrelevant one using cosine.

kernel	label	input query: partnership support
RBF	✓	The advisers work in *partnership* with the college staff and other university offices to provide information and *support* for all students and to offer programs on community issues as well as small-scale social activities.
Cosine	×	The Robert Mehrabian CIC is a *partnership* between Carnegie Mellon, the Carnegie Museums, and local economic development organizations and is funded with $8 million in Commonwealth of Pennsylvania tax *support*.

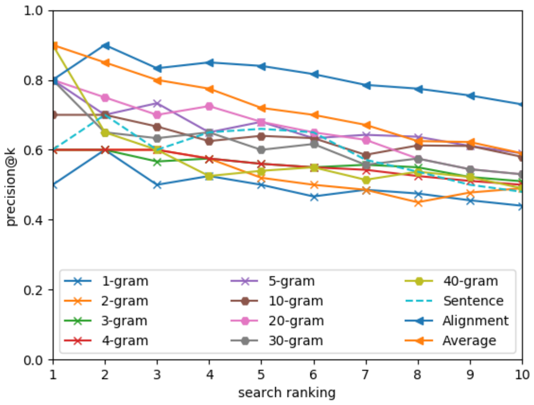

Figure 1: p@k of cosine similarity.

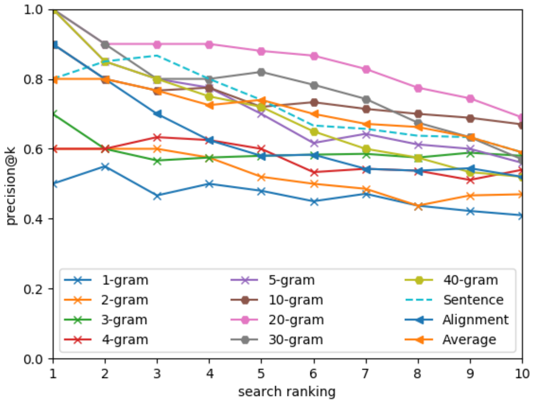

Figure 2: p@k of RBF kernel.

3.5 Discussion

We show the error analysis on the 19-gram RBF kernel, which exhibited the best score in the experimental results. Table 2 presents examples of one of the top-ten results of sentence retrieval. The topmost results of the proposed method comprise sentences relevant to the query. In contrast, the cosine baseline cannot consider latent intent; thus, it may output irrelevant sentences such as those without related keywords.

First, we measured the distance between words that match the query in the sentence of the test data exactly. The results show that the average distance between keywords was 11.8 words. In addition, for 72% of the gold sentences data, the keywords were found in the same clause. From these facts, we determined that useful sentences in this task were those in which keywords existed in the same clause, but were not located in close proximity. We regard this as one of the reasons why middle-sized N-grams were more effective.

Second, we compared alignment-based similarity with kernel embeddings. The former considers only the maximum similarity of words in the query and sentences. In contrast, kernel embeddings comprehensively considers all the words in the sentence. Furthermore, by combining this with the N-gram window approach, it is possible to focus on the surroundings of words with high similarity to the query. For these reasons, the kernel embeddings with N-gram window method outperforms alignment-based similarity.

4 Related Work

In recent years, many writing assistance systems have been developed. One of them is ESCORT, which is an English search system (Matsubara et al., 2008) for writing scholarly papers and survey reports; it aims to demonstrate examples of word usage. The input to this system is a sentence that will be parsed, and then the system will output sentences with the same syntactic structure. However, it assumes that there is a syntactic structure between keywords, which is not a valid assumption in our task. Further, latent intent of query is not modeled in their system.

In contrast, Chen et al. (2012) propose an English writing assistance system for ESL learners. The system, called FLOW, supplements the English vocabulary of non-English native speakers. If ESL learners cannot write in English owing to a lack of vocabulary, they can continue to write words in their first language within the sentence.

This system complements latent intent of query using their first language, whereas our approach improves sentence modeling using kernel embeddings.

In addition, Hayashibe et al. (2012) developed a tool to support English composition as the author writes. Like Chen et al. (2012), it accepts Romanized Japanese input in addition to English, to take the writer's first language into account. The tool can suggest a phrase considering context from the information already entered into the query. In contrast, we ask users to input only two words as a query. In addition, their example search system adopts an exact match approach, which may negatively impact the recall of the search system.

5 Conclusion

In this research, we proposed a new sentence retrieval method using a kernel embeddings framework to aid English composition. Our kernel embeddings method, using an RBF kernel and N-gram window, showed better results than two baseline methods (using cosine similarity and an alignment-based similarity). In future work, we aim to verify the effectiveness of our method for two or more queries.

References

Mei-Hua Chen, Shih-Ting Huang, Hung-Ting Hsieh, Ting-Hui Kao, and Jason S Chang. 2012. Flow: a first-language-oriented writing assistant system. In *Proceedings of the ACL 2012 System Demonstrations*, pages 157–162.

Yuta Hayashibe, Masato Hagiwara, and Satoshi Sekine. 2012. phloat : Integrated writing environment for ESL learners. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 57–72.

Shigeki Matsubara, Yoshihide Kato, and Seiji Egawa. 2008. Escort: example sentence retrieval system as support tool for English writing. *Journal of Information Processing and Management*, 51(4):251–259.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. 2007. A Hilbert space embedding for distributions. In *Proceedings of International Conference on Algorithmic Learning Theory*, pages 13–31.

Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280.

Yuya Yoshikawa, Tomoharu Iwata, Hiroshi Sawada, and Takeshi Yamada. 2015. Cross-domain matching for bag-of-words data via kernel embeddings of latent distributions. In *Advances in Neural Information Processing Systems 28*, pages 1405–1413.