Deep Learning Method to Extract Implicit Keywords for Historical Essay Questions

Yuanzhi Ke†1 *Kotaro Sakamoto*†2†3 *Madoka Ishioroshi*†3 *Hideyuki Shibuki*†2
Masafumi Hagiwara†1 *Tatsunori Mori*†2 *Noriko Kando*†3†4

†1*Keio University* †2*Yokohama National University*
†3*National Institute of Technology* †4*SOKENDAI*
E-mail: {enshika8811.a6, hagiwara}@keio.jp, {sakamoto, shib, mori}@forest.eis.ynu.ac.jp,
{ishioroshi, noriko}@nii.ac.jp

1 Introduction

AI and deep learning technology have been successful for reading comprehension [1] and shown effectiveness for multi-choice questions [2, 3, 4]. However, it is still challenging to answer the essay questions that require the examinees to write an essay for the question. Especially, the historical essay questions in the entry tests of universities usually provide few materials and require the examinees to search the related knowledge in their brain and organize them into an essay. In such questions, stories are not provided together like those in the reading comprehension task. The machines need to recall some additional related keywords that are not in the question text and search an external knowledge base. Moreover, historical terms have much less frequency which makes them hard to learn. Thirdly, because of the large knowledge base, it is expensive to build an end-to-end neural network model like that for reading comprehension.

In our works, we pay attention on the extraction of the implicit keywords for the retrieval of the related knowledge. Implicit keywords here refer to the keywords that are the scoring points but do not appear in the question text. They are important to answer essay questions and also help the machine get related knowledge. It has the following challenging points:

1. The implicit keywords are not in the question text.
2. The relationship of the question and the answer is hard to be defined by rules. Rule-based systems are very limited for the task, which motivated us to explore machine learning-based methods.
3. The keywords are the historical events and entities, which are infrequent. It is challenging to train their embeddings.

For the issues above, we propose a bi-directional LSTM model with attention mechanism to extract and rank the implicit keywords, a data augmentation method and a learning controlling method to improve the training of the embeddings. To validate our method, we compared the performance of the proposed methods with the conventional methods in the task of implicit keyword extraction. Our achievements include,

1. Our method outperformed TF-IDF, BM25, and Elman network.
2. We proposed a novel method to augment the data and showed that it is effective to improve the performance.
3. We have shown that controlling the learning rate for different words by their frequencies is effective for a more reasonable ranking.

2 Baselines

2.1 TF-IDF and BM25

The hypothesis of the two methods is that the keywords have higher TF-IDF score [5] in the related texts. In the TF-IDF-based method, at first, the system searches the knowledge base for the related texts

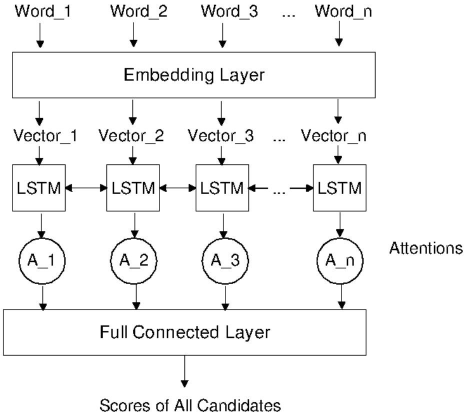


Fig. 1. Architecture of the neural network for ranking in our system.

by the TF-IDF score, then ranks the content words by the TF-IDF scores of the words. Finally, the system outputs the top-ranked words that are not included in the question text.

The BM25-based method is similar to the TF-IDF-based method, but the Okapi BM25 [6] algorithm is used instead of TF-IDF.

2.2 Elman Network

Elman network is a simple recurrent neural network (RNN) model. It has been reported to be effective to extract the implicit keywords [7]

3 Methodology

3.1 Bi-directional RNN of LSTM with Attention For Ranking

The architecture of the proposed neural network is shown in Fig. 1. Formally, denote the vectors of words in the input text as $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ...\mathbf{x}_t, ...\mathbf{x}_n$, where n is the length of the input. The output of each LSTM unit $\mathbf{y}_t$ is the follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f\left[\mathbf{y}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_f\right), \\
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i\left[\mathbf{y}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_i\right), \\
\tilde{\gamma}_t &= \tanh\left(\mathbf{W}_c\left[\mathbf{y}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_c\right), \\
\gamma_t &= \mathbf{f}_t * \gamma_{t-1} + \mathbf{i}_t * \tilde{\gamma}_t, \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o\left[\mathbf{y}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_o\right), \\
\mathbf{y}_t &= \mathbf{o}_t * \tanh\left(\gamma_t\right),
\end{aligned} \tag{1}$$

where, $\sigma(\cdot)$ and $*$ are the element-wise sigmoid function and multiplication operator.

The attention mechanism is to compute a logistic regression for each output of the LSTM units as its coefficient of importance. Denote a_t as the attention of $\mathbf{y}_t$,

$$a_t = \sigma\left(W_a\left[\mathbf{y}_t\right] + b_a\right). \tag{2}$$

The final output prediction for each class is,

$$\mathbf{z} = \sum_t^n \left(a_t \cdot \mathbf{y}_t\right), \tag{3}$$

$$\mathbf{Pr}\left(\hat{y} = i|s\right) = \frac{exp\left(\mathbf{W}_p^i \mathbf{z} + \mathbf{b}_p^i\right)}{\sum_{i' \in \mathcal{E}} exp\left(\mathbf{W}_p^{i'} \mathbf{z} + \mathbf{b}_p^{i'}\right).} \tag{4}$$

Here i is one of the keywords in the keyword candidate set I.

3.2 Learning Rate Controlling

The frequencies of the words in the answers are inevitably unbalanced. The corresponding parameters of the more frequent words are updated more times. As the proposed model is deep, it is easy to be overfitted by the unbalanced data. To balance the training progress for different words, we control the learning rate for each word according to its frequency in the dataset. Formally, denote α as the global learning rate, $TF(i_k)$ as the frequency of keyword candidate i_k, the learning rate α_k for keyword candidate i_k is defined as the follows,

$$\alpha_k = \frac{\alpha}{TF(i_k)}. \tag{5}$$

Hence, the parameters for the frequent words are trained more slower while those for the rare ones are trained more faster.

3.3 Data Augmentation

In Japanese, the subjects are often omitted. In the glossary included in our corpus to train the word embeddings, when the subject is always shown as the title, the descriptions of most documents do not contain the subject. It makes the machine difficult to make use of them to build relationship between the subject and the description. To help the machine to recognize the relationship between them, we made a script to concatenate the subject word and each sentence in the description for each document in the glossary.

4 Experiments

4.1 Purpose

To validate the effectiveness of our proposed method, we compared its performance for the implicit keyword extraction task with the baselines. We investigated the ranking scores of the candidate keywords ranked by different methods, and compare them with the gold standards. Besides, in order to validate the data augmentation method alone, we also applied it to Elman network and compared it with the vanilla Elman network.

4.2 Setup

The knowledge resource we used was 3 history textbooks, 1 glossary (7,038 terms included), and 6 exercise books (1,205 questions includes). The textbooks include “世界史 A”, “新選世界史 B” and “世界史 B” published by Tokyo Shoseki. The glossary is “世界史 B 用語集改訂版” published by Yamakawa Shuppansha. The exercise books include “詳説世界史論述問題集” published by Yamakawa Shuppansha, “実力をつける世界史１００題改訂第３版” and “段階式世界史論述のトレーニング” published by Z-kai, “判る解ける書ける世界史論述” published by Kawai Publishing, “大学入試世界史 B 論述問題が面白いほど解ける本” published by KADOKAWA/Chukei, and “世界史論述練習帳 new” published by Parade. The text book and the glossary were employed to pretrain the word embeddings, with data augmentation described in Section 3.3. The question and answers pairs in the exercise books were used to train the ranking models. For the test set, we used the first question on world history of the admission test of Tokyo University from 2000 to 2011 (totally 12 questions).

4.3 Metric

Because we want to investigate whether the methods are able to rank the correct keywords as high as possible, we use Mean Average Precision at K (MAP@K) [8] as the metric. To observe whether the keywords are gathered at the top, and for ease of comparison between the proposed methods and the reported performance of the baselines [7],

Table 1. The Performance Measured by MAP@K. “DA” refers to data augmentation. “LC” refers to learning rate controlling.

Method	@5	@10
Baseline(TF-IDF) [7]	0.522	0.433
Baseline(BM25) [7]	0.493	0.425
Baseline(Elman net) [7]	0.594	0.572
Proposed(Elman net+DA)	0.712	0.643
Proposed(BiLSTM+DA)	**0.732**	0.651
Proposed(BiLSTM+DA+LC)	0.729	**0.675**

Table 2. The average hits with/without learning rate controlling. “DA” refers to data augmentation. “LC” refers to learning rate controlling.

Method	@5	@10
Proposed(BiLSTM+DA)	2.083	3.667
Proposed(BiLSTM+DA+LC)	**2.333**	**4.08**

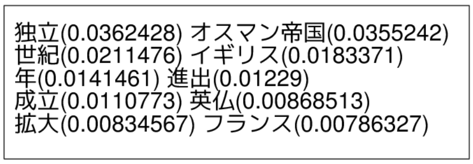
5 Results and Discussion

5.1 Results

The results are shown in Table 1. The proposed method achieved the best performance. The Elman network with data augmentation outperformed the baseline Elman network. Learning rate controlling improved the MAP@10 of the RNN of LSTM, but failed to improve the MAP@5 of it. It is an interesting result, hence we additionally compared the average hits of the keywords with and without learning rate controlling as shown in Table 2. We find that even though there is not significant difference between MAP@5 with or without learning rate controlling, the learning rate controlling brings more hits of correct keywords at top 5 and at top 10.

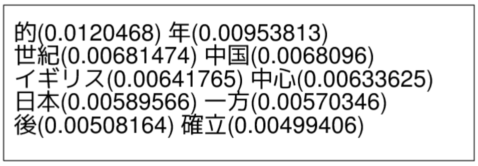
5.2 Discussion

The results show that the proposed method is more effective than the conventional ones. The improvement of Elman network brought by data augmentation in the results indicates that data augmentation is very powerful to improve the performance. The improvement of the average hits by learning rate controlling shows that it is effective to make the system to highly rank more correct keywords.



```
独立(0.0362428) オスマン帝国(0.0355242)
世紀(0.0211476) イギリス(0.0183371)
年(0.0141461) 進出(0.01229)
成立(0.0110773) 英仏(0.00868513)
拡大(0.00834567) フランス(0.00786327)
```

Fig. 2. An output by the proposed method.



```
的(0.0120468) 年(0.00953813)
世紀(0.00681474) 中国(0.0068096)
イギリス(0.00641765) 中心(0.00633625)
日本(0.00589566) 一方(0.00570346)
後(0.00508164) 確立(0.00499406)
```

Fig. 3. An output by Elman net.

As a qualitative validation of the bi-direction RNN of LSTM model, we also compared the outputs by the proposed model with the others. Fig. 2 and Fig. 3 show examples of the outputs by the proposed model and the Elman network respectively. We can see that the output by the proposed method is full of meaningful content words of historical entities and events, while Elman network outputs some frequent but not critical words such as "的", "一方" and "後". Moreover, we can see that the scores of the critical keywords such as "オスマン帝国" and "英仏" are higher. The findings indicate that the bi-directional RNN of LSTM can achieve better rankings.

6 Conclusions

For learning and ranking the implicit keywords for historical essay questions, we proposed a bi-directional RNN of LSTM, a data augmentation method and a learning rate controlling trick. The experimental results indicate that the bi-directional RNN of LSTM can offer better rankings, the data augmentation method obviously improves the performance and the learning rate controlling makes the bi-directional RNN model extract more critical keywords.

We believe that there is still room for improvement. For example, transfer learning is expected to help the scenes where the data is limited. We are going to explore such learning technology for this task in the future. Besides, the tunning of the parameters of the stochastic gradient descending optimizer was difficult for the objective of the proposed model. We also would like to explore other suitable objective functions and optimizers for the task.

Acknowledgement

The authors would like to thank IR-Advanced Linguistic Technologies, Inc. for providing us the gold standards, and the students in Yokohama National University for helping us computerize the datasets.

References

[1] Karl Moritz Hermann, Tom Koisk, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems, (2015).

[2] Hideyuki Shibuki, Kotaro Sakamoto, Yoshionobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. Overview of the NTCIR-11 QA-Lab Task. NTCIR-11, (2014).

[3] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshionobu Kano, Tatsunori Mori, and Noriko Kando. Overview of the NTCIR-12 QA Lab-2 Task. NTCIR-12, (2015).

[4] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. Overview of the NTCIR-13 QA Lab-3 Task. NTCIR-13, (2017).

[5] Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development, 1(4), 309-317, (1957).

[6] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. Proceedings of the Third Text REtrieval Conference (TREC 1994), (1994).

[7] 阪本浩太郎, 陸宇傑, 福原優太, 渋木英潔, 石下円香, 森辰則, 神門典子. 大学入試世界史論述問題における非指定重要語句生成に関する検討. 言語処理学会第 23 回年次大会, (2017).

[8] Nick Pentreath, Manpreet Singh Ghotra, and Rajdeep Dua. Machine Learning with Spark. Packt Publishing Ltd., 109-113, (2015).