An Approach to Summarize Concordancers' Lists Visually to Support Language Learners in Understanding Word Usages

Yo Ehara
Shizuoka Institute of Science and Technology / 2200-2, Toyosawa, Fukuroi, Shizuoka, Japan
ehara.yo@sist.ac.jp

Abstract

Concordancers are interactive software that searches for the input word and displays the list of its usages in a corpus. They have been widely used by language learners and educators to analyze word usages. Because naively listing all usages of the word overwhelms users, determining how to summarize the list is important for usability. Previous studies summarized the list by using the surrounding word patterns and showed their frequency; however, such a naive method counts substantially the same usages, such as "the book" and "a book," separately; hence, such a method is not very informative to learners. Here, a novel approach for summarizing the list is proposed. According to the user's input word, the proposed system semantically visualizes each usage of the word using contextualized word embeddings interactively. It is shown that the system responds quickly with intuitive use cases.

1 Introduction

Concordancers are interactive software tools that search and display a usage list of the input words or word patterns within a corpus. The tools have been widely used in corpus linguistics and computer-aided language education to assist language learners and educators analyze word usages within a corpus (Hockey and Martin, 1987). In Natural Language Processing (NLP), studies have built sophisticated concordancers to support second language writing and translators in searching bilingual sentence-aligned corpus (Wu et al., 2004; Jian et al., 2004; Lux-Pogodalla et al., 2010). However, the information that conventional concordancers can provide for analyses of each usage is limited to the frequency of surrounding context patterns, parts of speech, and so on. The words that second language learners can search to learn their usages tend to be frequent.

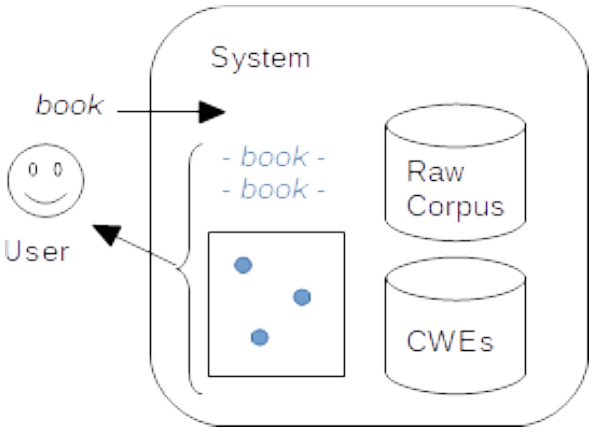

Figure 1: System layout. CWE means contextualized word embeddings.

Therefore, a more sophisticated method to summarize many word usages in a large corpus for concordancers is desirable. Recently, contextualized word embeddings such as (Devlin et al., 2019) were proposed in NLP to capture the context of each word usage in vectors and to model the semantic distances between the usages using contexts as a clue. Unlike previous studies (Liu et al., 2017; Smilkov et al., 2016) that visualized different words using word embeddings, in this paper, we introduce a novel system intuitively helpful for concordancer users to visualize different usages of a word of interest.

2 System Overview and Use Cases

Fig. 1 shows our system layout. Once a user provides a word to the system, it automatically searches the word in the corpus in a similar way to typical concordancers. Unlike concordancers, our system has a database that stores contextualized word embeddings for each *usage* or occurrence of each word in the corpus. We used half a million sentences from the British National Corpus (BNC

Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019), pages 29–31,
Tokyo, Japan, October 29 2019.

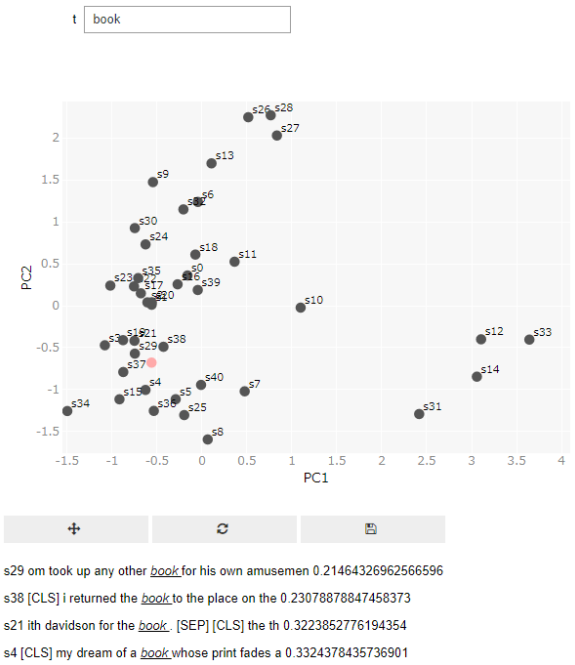

Figure 2: Use case of searching the word *book*.

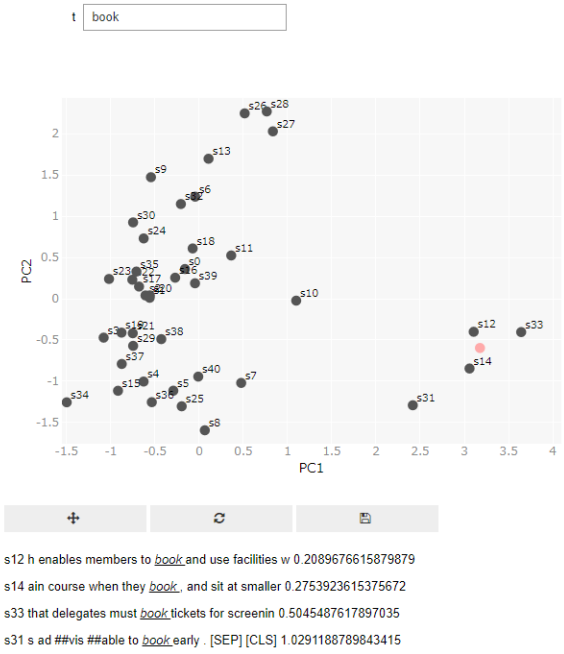

Figure 3: Another use case of searching the word *book*.

Consortium, 2007) as the raw corpus. We built the database by applying the **bert-base-uncased** model of the PyTorch Pretrained the BERT project [1](Devlin et al., 2019) to the corpus. We used the last layer, which was more distant from the surface input, as the embeddings. The size of the database is roughly 200MB per thousand sentences. Our system visualizes these searched contextualized word embedding vectors. We visualize the contextualized word embedding vectors for the provided word by projecting these vectors into a two-dimensional space. To visualize, we used principal component analysis (PCA) because its fast calculation is beneficial for short system response time and better interactivity. The number of points in the visualization is also set to a maximum of 100 so that users can easily understand it.

Fig. 2 shows a use case of searching *book*. [2]. Users can directly type the word in the textbox shown at the top of Fig. 2. Below is the visualization of the usages found and their list. Each dark-colored point links to each usage. The red lightly-colored point is the *probe point*: the usages are listed in the nearest order of the probe point. No usage is linked to the probe point. Users can freely and interactively drag and move the probe point to change the list of usages below the visualization. Each line of the list shows the surrounding words of the usage, followed by the distance between the vectors of the usage and probe point in the two-dimensional visualization. In Fig. 2, the probe point is on the left part of the visualized figure. In the first several lines of the list, the system successfully shows the usages of the word *book* about reading. In contrast, Fig. 3 shows the case, in which the users drag the probe point from the left to the right of the visualization. The first several lines of the list or the usages nearest the probe point show the usages of the word *book* about reservation. A careful reading of the usage list below shows that the words surrounding the word *book* vary. Thus, merely focusing on the surrounding words, such as "to" before *book*, cannot distinguish the usages of *book* about reservation from the usages of *book* about reading.

3 Demo Outline

We are expecting language learners to be users. We are planning to make our software openly available under an open-source license after we evaluate our system in more detail [3]. As for the interoperability of the software, the software is

[1] https://github.com/huggingface/pytorch-pretrained-BERT

[2] Fig. 2 and Fig. 3 shows use cases on a 10,000-sentence exerpt of the BNC corpus to avoid having too many hits hinder the reading of the paper.

[3] When we are prepared to make our software public, we plan to announce the details under https://yoehara.com/.

built on the Jupyter notebook [4] using ipywidgets [5]; hence it is accessible online via browsers without the need to install it to each learner's terminal computer.

4 Conclusions

We proposed a novel concordancer that can search the usages of a word and visualize the usages using contextualized word embeddings. Through use cases, we illustrated that a learner can understand different types of usage of *book*, which could not be captured only by surface information of the surrounding words. As future work, we will evaluate our system on more practical use cases with many language learners, especially from the perspective of support systems for second language vocabulary learning and reading (Ehara et al., 2012, 2013, 2014).

5 Acknowledgments

This work was supported by JST, ACT-I Grant Number JPMJPR18U8, Japan. We used the AI Bridging Cloud Infrastructure (ABCI) by the National Institute of Advanced Industrial Science and Technology (AIST) for computational resources. We thank anonymous reviewers for their insightful and constructive comments.

References

The BNC Consortium. 2007. *The British National Corpus, version 3 (BNC XML Edition).*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing word sampling for vocabulary prediction as graph-based active learning. In *Proc. of EMNLP*, pages 1374–1384.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining words in the minds of second language learners: learner-specific word difficulty. In *Proc. of COLING*.

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2013. Personalized reading support for second-language web documents. *ACM Transactions on Intelligent Systems and Technology*, 4(2).

[4] https://jupyter.org/

[5] https://ipywidgets.readthedocs.io/en/latest/

Susan Hockey and Jeremy Martin. 1987. The Oxford Concordance Program Version 2. *Digital Scholarship in the Humanities*, 2(2):125–131.

Jia-Yan Jian, Yu-Chia Chang, and Jason S. Chang. 2004. TANGO: Bilingual collocational concordancer. In *Proc. of ACL demo.*, pages 166–169.

Shusen Liu, Peer-Timo Bremer, Jayaraman J Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2017. Visual exploration of semantic relationships in neural word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):553–562.

Véronika Lux-Pogodalla, Dominique Besagni, and Karën Fort. 2010. FastKwic, an "intelligent" concordancer using FASTR. In *Proc. of LREC*.

Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B Viégas, and Martin Wattenberg. 2016. Embedding projector: Interactive visualization and interpretation of embeddings. In *NIPS Workshop on Interpretable Machine Learning in Complex Systems*.

Jian-Cheng Wu, Thomas C. Chuang, Wen-Chi Shei, and Jason S. Chang. 2004. Subsentential translation memory for computer assisted writing and translation. In *Proc. of ACL demo.*, pages 106–109.