# Write-righter: An Academic Writing Assistant System

**Yuanchao Liu, Xin Wang, Ming Liu, Xiaolong Wang**

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
ycliu@hit.edu.cn, xwang@insun.hit.edu.cn, mliu@insun.hit.edu.cn, wangxl@insun.hit.edu.cn

**Abstract**

Writing academic articles in English is a challenging task for non-native speakers, as more effort has to be spent to enhance their language expressions. This paper presents an academic writing assistant system called Write-righter, which can provide real-time hint and recommendation by analyzing the input context. To achieve this goal, some novel strategies, e.g., semantic extension based sentence retrieval and LDA based sentence structure identification have been proposed. Write-righter is expected to help people express their ideas correctly by recommending top N most possible expressions.

## Introduction

It is a challenging work for non-native speakers to write articles in English, as more effort has to be spent to enhance their language besides from content development. Some writing assistant systems have been proposed correspondingly. FLOW (Chen et al., 2012) is an interactive writing assistant system which aims mainly at providing phrases suggestions. Translation-based English writing assistant system PENS (Liu et al., 2000) can provide translated words or phrases, thus can allow users to write in their native language occasionally. EAME (Yang et al., 2000) is an academic abstract writing assistant system, which uses a question-driven framework to create abstract drafts by asking users to select some patterns and fill blanks. Corpus linguistics plays an important role in writing assistant system (Sun et al., 2007; Dai et al., 2014).

This paper presents a corpus based academic writing assistant system called Write-righter. In the following, we will briefly introduce the system and its main strategies.

## The main framework of Write-righter

The system is a stand-alone editor[1] and the attendees can interact with it directly in the session. The main framework is shown in Fig. 1. While users are writing in the client, the server side can give corresponding real-time hint and recommendation for words, phrases, example sentences and similar words in separate areas. Currently, 5.4GB index generated from 40, 243 scientific papers, which is mainly crawled from *Sciencedirect*[2], is used in the system.

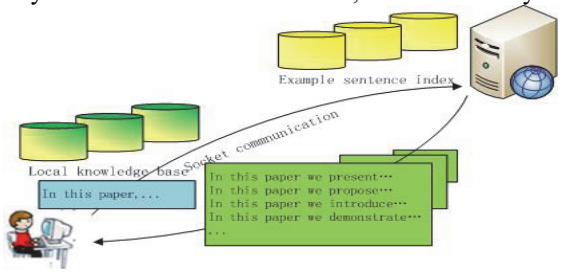


Fig. 1. The main framework of write-righter

## Semantic extension based sentence retrieval

Semantic extension is very useful for fully utilizing the existing corpus. For example, supposing the original sentence is "***Several*** *online writing assistance tools have been* ***developed*** *through efforts in the areas of natural language processing"*, the extensions can be like this: 1). "*several*"→"*various*"; 2). "*developed"→"designed"* and etc. In this way the recommended results may be richer by using extension query. As shown in Fig.2, we use the *Wordnet*[3] synonym to construct the lexicon for semantic extension. Only frequently used words in academic papers are extended, because they are so frequent that authors tend to express their meaning using different forms to improve the diversity.

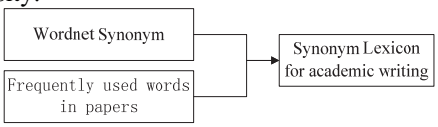


Fig. 2. The construction of synonym lexicon



[1]The demo video address: *http://www.iqiyi.com/w_19rt9lbzz5.html*
[2] *http://www.sciencedirect.com*
[3] *http://wordnet.princeton.edu/*

## LDA based pattern identification

Patterns here means the frequently used expressions and they are usually composed of non-topic words, e.g., the underlined part in sentence "***Various*** *online writing assistance tools* ***have been presented through efforts in the areas of*** *natural language processing*".

We use mallet LDA (McCallum, et al, 2002) to get the topic distribution. The probability that one word $w$ is topic word $P_T(w)$ can be calculated in formula (1) and (2).

$$P_T(w) = p_w^M / \sum_{i=0}^{|T|-1} p_w(i)) \quad (1)$$

$$P_w^M = \max(p_w(i)), \quad i = 0,...,|T|-1 \quad (2)$$

where $p_w(i)$ means the probability of term $w$ in topic $i$. $|T|$ means topic count. Bigger $P_T(w)$ usually means that $w$ is more likely to be a topic word. Correspondingly the probability that $w$ is non-topic word (structure word) $P_S(w)$ can be calculated as follows.

$$P_S(w) = 1 - P_T(w) \quad (3)$$

If $P_S(w) > P_S^\tau(w)$, then $w$ is more likely to be the structure word. Here $P_S^\tau(w)$ is the threshold, and we set $P_S^\tau(w)$ to 0.65 and use 10 topics by experience.

## The dynamic phrase hints

The dynamic hint of phrases is based on user's current input, the whole context information, academic N-gram phrase library extracted from the *Sciencedirect* corpus. Suppose the local context is $C_1$ (usually the recently inputted two words), the whole context is $C_2$ (all texts in composing area), the probability of each candidate phrase is $P_p(C_1, C_2)$, and then top N phrases with biggest probability will be ranked and selected to appear in the phrase hint area.

$$phrase_{1,2,...N} = \arg\max_{1,2,3....N}(P_p(C_1, C_2)) \quad (4)$$

Generally, for each phrase in the phrase hint area, the first consecutive words must be same as $C_1$. The calculation of phrase probability depends on two kinds of information: 1). the phrase frequency $Fre_p$ in the corpus; 2). the similarity with the users' input topic.

In detail, if the number of words in input area is bigger than threshold (e.g., 30 words), the whole context information $C_2$ will be considered and incorporated. $P_p(C_1, C_2)$ can be calculated by the following formulas.

$$P_p(C_1, C_2) = \alpha * Sim1 + \beta * Sim2, \quad (5)$$

$$0 \le a \le 1.0, 0 \le \beta \le 1.0, \alpha + \beta = 1.0$$

$$Sim1 = Fre_p / Max(\{Fre_i\}), \quad i = 0,...,|\{p\}| \quad (6)$$

$$Sim2 = Simlarity(C_2, p) \quad (7)$$

Where $Sim2$ denotes the cosine similarity between the candidate phrase $p$ and the users' whole input. We set $\alpha = 0.6, \beta = 0.4$ in this paper. The steps are shown as follows: 1). Get the local context $C_1$ and the whole context $C_2$; 2). Find phrases from the phrase library which begins with $C_1$; 3). Compute the comprehensive probability of each phrase; 4). Sort these phrases using probability $P_p(C_1, C_2)$ in descending order; 5). Output top N phrases. There are 34, 040 phrases in the library and we set N=20 in this paper.

## Conclusions and future work

By mining from large-scale published papers, Write-righter can provide recommendations by trying to "guess" users' writing intention. For future, we will incorporate more user preferences into the suggestion and make corresponding extensions. Besides, it may be interesting to provide revision suggestions directly by leveraging current input sentence and good-quality expression patterns.

## Acknowledgements

This work is funded by National Natural Science Foundation of China (61572151).

## References

Mei-Hua Chen, Shih-Ting Huang, Hung-Ting Hsieh, Ting-Hui Kao, Jason S. Chang. FLOW: A First-Language-Oriented Writing Assistant System. ACL 2012, pages 157–162

Liu, Ting, Mingh Zhou, Jianfeng Gao, Endong Xun, and Changning Huan. PENS: A Machine-Aided English Writing System for Chinese Users. ACL 2000. pp 529-536.

Yong-Lin Yang. EAME: A question-driven abstract generating system. Modern foreign languages, 2003: 296-305 (In Chinese)

Yu-Chih Sun. Learner Perceptions of a Concordance Tool for Academic Writing. Computer Assisted Language Learning. Vol. 20, No. 4, October 2007, pp. 323 – 343

CHEN H, HE B. Automated Essay Scoring by Maximizing Human-Machine Agreement. EMNLP 2014. pages 1741-1752.

McCallum, Andrew Kachites. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

Xianjun Dai, Yuanchao Liu, Xiaolong Wang. WINGS: Writing with Intelligent Guidance and Suggestions. ACL 2014, Baltimore, Maryland USA, June 23-24, pages 25–30.