Linggle Knows: A Search Engine Tells How People Write

Jhih-Jie Chen[1], **Hao-Chun Peng**[2]
Mei-Cih Yeh[1], **Peng-Yu Chen**[1], **Jason S. Chang**[2]
[1]Department of Computer Science
[2]Institute of Information Systems and Applications
National Tsing Hua University
{jjc, henry.p, meicih, pengyu, jason}@nlplab.cc

Abstract

This paper presents *Linggle Knows*, an English grammar and linguistic search engine. *Linggle Knows* help people writing by displaying lexical and grammatical information extracted from a couple of large scale corpora, including Google Web 1T 5-gram, British National Corpus (BNC), New York Times Annotated Corpus (NYT), etc. It not only describes how a word is genuinely used, but also recommends various alternative collocations and word combinations. In addition, it gives real-world examples to better explain how a word is used in reality.

1 Introduction

It is estimated that roughly a billion people are learning and using English around the world (Graddol, 2003), most of which are second language (L2) learners. More specifically, further analysis reported that there are 375 million native speakers of English, and 750 million people use English as a second language (Crystal, 1997). Writing is probably the most difficult and profound among the four skills of language learning, even for a native speaker. For L2 Writer, much of the frustration in writing stems from the lack of vocabulary, misused preposition or verb, insufficient understanding of grammar, etc. Consequently, people have developed a variety of writing assisting tools to help writing.

Oxford Dictionaries contains extensive vocabularies along with explanations and examples. *NetSpeak* manipulates *Google Web 1T 5-gram* to provide a way of accessing n-gram information and is capable of filling the blank, reordering the text, choosing a better preposition, etc. Meanwhile, *Linggle* features better n-gram retrieval performance than *NetSpeak* and advances some ideas such as query with specific part of speech, and operator nesting (Boisson et al., 2013). *Grammarly* and *Ginger Software* check and correct grammatical errors, but only to the extent that is fairly narrow. *Write & Improve* gives corrective feedback sentence by sentence and assigns an overall grade for a submitted essay. On the other hand, *WriteAhead* proposed an interactive writing environment which suggests subsequent patterns or collocations while the user is writing away (Yen et al., 2015).

Each of the above tools indeed solves some writing problems in somewhat different ways. Yet there is no such an integrated system trying to solve all kinds of problems of writing considerately. As a result, we have to switch from one window to the other while trying to solve different kinds of writing problems. It is quite disturbing and sometimes upsetting, since it severely affect the efficiency and reduce the productivity of writing. Our objective is to develop a comprehensive tool which provides essential linguistic knowledge to help people obtain required information immediately and effortlessly.

We incorporate four mechanisms we considered the most important. N-gram search provides linguistic information in which you can fill the black or search for appropriate preposition. Pattern grammar enables giving instant writing suggestions while typing away. Rephrasing recommends correct or better use of words, while example sentence illustrate actual uses in real world.

In the following sections, we introduce the system design, interface, and underlying architecture. Next, we briefly describe each of the four subsystems. Finally, we exploit the great potential of the system and envision the future of writing.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 166–169, Osaka, Japan, December 11-17 2016.

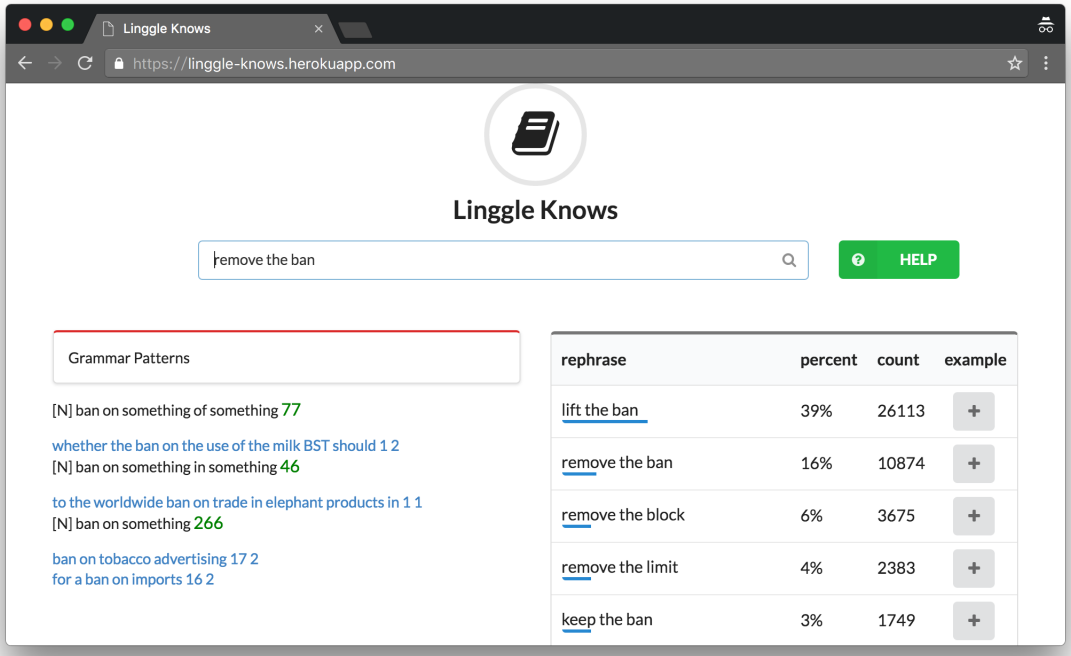

Figure 1: Example using *Linggle Knows* typing "remove the ban".

2 Linggle Knows

To establish user friendly, our design is to be simple and intuitive. Again, our objective is to develop a comprehensive tool helping people obtain information immediately and effortlessly. We build *Linggle Knows* as a web application, which can be accessed easily through browsers whenever using computer, laptop, or tablet. As shown in Figure 1, *Linggle Knows* is accessible at `https://linggle-knows.herokuapp.com`.

2.1 User Interface

Aiming at improving the productivity of writing, we design the interaction to be simple and intuitive. The interface purely consists of a search box and a result area. Instant feedback is received while entering words in the search box. The result is rendered in a clear and straightforward way so that users can find desired information right away. Moreover, n-gram search is very powerful but requires a certain understanding of linguistics, which is not innate in everyone. If one uses the query operator, n-gram information is displayed. Otherwise, our system shows writing suggestion based on grammar patterns and paraphrase recommendation. Interface characteristics described above reflects the design philosophy to be simple and intuitive.

2.2 System Architecture

To develop a robust and reliable system, we implement the four main components separately which can be accessed in a standard way. Each independent component can be accessed through RESTful API in JSON format, which is universal and can be easily utilized. The main system retrieves different information from the four independent subsystems through the universal interface mentioned above.

3 N-gram

We adopted the query functions described in (Boisson et al., 2013), whereas a different set of query operators is defined. The syntax of the patterns for n-grams is shown in Figure 2, and the explanation and examples are described in the following subsection.

Operators	Description	Example
*	match zero or more words	play * role
_	match any word	listen _ music
~	search for the similar words of TERM	~reliable person
?	search for TERM optionally	listen ?to music
/	either TERM1 or TERM2	receive/accept education
PoS.	search for word with specific part-of-speech tag. (v, n, adj, adv, prep, det, conj, pron)	v. det. report

Figure 2: Query operator instruction

3.1 Query Instruction

Wildcard enables the users to query zero, one or more arbitrary words up to five words in total. (i.e., "play * role" is intended to search for a maximum distance of three words.) Besides, the "?" operator before a word stands for a search of n-grams with or without the word. (i.e., one wanting to determine whether to use the word "to" between listen and music, then one can make the query "listen ?to music.") Yet another operation "/" is to search for information related to word choice. (i.e., "receive / accept * education" can be used to reveal that **receive education** is used much more often than **accept education**.) Finally, a set of PoS symbols is defined to support queries that need more precision than the symbol "_".

4 Pattern Grammar

Pattern grammar identifies the syntactic information of individual lexical terms (Hunston et al., 1996). As envisioned by(Hearst, 2015), writing software can be more effective if they can facilitate intensive interaction while writing in progress. (e.g., giving feedback for every word entered even with only partially written sentences or incomplete paragraphs). As described in (Yen et al., 2015), grammar patterns can be used to give instantaneous feedback while typing away, and such information can be extracted by generalizing the words nearby a term. In Figure 1, *Linggle Knows* provides writing suggestions by displaying extracted patterns along with examples.

5 Paraphrases and Corrections

Paraphrasing is the action of restating meaning using different words. It has been shown that for the English Language Leaners (ELLs) the inability in paraphrasing may hinder the writing skills and the ability of expression (Ismail and Maasum, 2009) . To help ELLs, a promising approach is automatic paraphrase generation (APG). In this section, we introduce a new strategy for extracting synonyms from large scale monolingual corpus, and then automatic paraphrase generation with corrections.

5.1 Synonyms Extraction

We separate this stage into two steps. First, we extract potential synonyms from a larges scale monolingual corpus (e.g., BNC), by exploiting different kinds of surface patterns (i.e., "ADJ and ADJ", "ADJ or ADJ", "NOUN but NOUN", etc.), to the extent of adjective, verb, adverb, and noun.

However, these extracted potential synonyms may contain some noise. In order to filter out non-synonyms, we apply rank ratio (RR) statistics (Deane, 2005) and adjust the overlap coefficients of our strategy. Finally, we tune both the RR and overlap coefficient thresholds to refine extracted synonyms.

5.2 Automatic Paraphrases Generation

We exploit Web-scale n-grams and word embedding to automatically generate paraphrases. First, we use large scale monolingual corpora to train a word2vec model (Mikolov et al., 2013a; Mikolov et al., 2013b). Second, we store each word with its corresponding vector, as well as its synonyms mentioned above into a database.

At run-time, words in the given query are substituted by their synonyms to derive candidate paraphrases. However, substituting blindly may lead to awkward phrases and sentences. To resolve this

problem, we utilize Web-scale n-gram statistics via the linguistic search engine in section 3, to filter out improper candidate paraphrases. Next, we retrieve each word's vector from the database to construct the phrasal vector, then we compute the cosine similarity of the given query. Finally, we rerank the result based on cosine similarity and n-gram statistics.

6 Example Sentences

To help learners use the lexicon properly, examples are especially important. We collect a set of text data over 100 GB, and use *elastic search* (`github.com/elastic/elasticsearch`) to index sentences in several corpora, including NYT, VOA English, news crawl data from WMT16, etc.

6.1 Good Dictionary Example

Elasticsearch is not designed for indexing sentences and finding *good dictionary examples* (GDEX) as described in (Kilgarriff et al., 2008). Therefore, we apply the GDEX method to rank and select appropriate and representative sentences retrieved from Elasticsearch. The GDEX method considers sentence length, word frequency, the presence of pronouns, and most importantly collocations.

7 Conclusion and Future Work

Linggle Know shows the great potential of incorporating different approaches to help writing. Not only did they solve different kinds of writing problems, but also they complement and reinforce each other to be a complete and effective solution. Despite the extensive and multifaceted feedback and suggestion, writing is not all about syntactically or lexically well-written. It involves contents, structure, the certain understanding of the background, and many other factors to compose a rich, organized and sophisticated text. (e.g., conventional structure and idioms in academic writing). There is still a long way to go to accomplish the ultimate goal. We envision the future of writing to be a joyful experience with the help of instantaneous suggestion and constructive feedback.

References

Joanne Boisson, Ting-Hui Kao, Jian-Cheng Wu, Tzu-Hsi Yen, and Jason S Chang. 2013. Linggle: a web-scale linguistic search engine for words in context. In *ACL (Conference System Demonstrations)*.

David Crystal. 1997. English as a global language. *Cambridge University Press*.

Paul Deane. 2005. A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 605–613. Association for Computational Linguistics.

David Graddol. 2003. The decline of the native speaker. *Translation today: trends and perspectives. Clevedon: Multilingual Matters*.

Marti A Hearst. 2015. Can natural language processing become natural language coaching? In *Proceedings of the 53rd Annual Meeting of the Association for Computation Linguistics. ACL*, pages 1245–1252.

Susan Hunston, Gill Francis, and E Manning. 1996. Collins cobuild grammar patterns 1: verbs.

Syafini Ismail and N. R. Maasum. 2009. The effects of cooperative learning in enhancing writing performance. In *Proceedings of the language and culture:creating and fostering global communities*. SOLLS.INTEC 09 International Conference.

Adam Kilgarriff, Milos Husák, Katy McAdam, Michael Rundell, and Pavel Rychlỳ. 2008. Gdex: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX International Congress*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Tzu-Hsi Yen, Jian-Cheng Wu, Joanne Boisson, Jim Chang, and Jason Chang. 2015. Writeahead: Mining grammar patterns in corpora for assisted writing. *ACL-IJCNLP 2015*.