Automatic collocation suggestion for student academic texts

Baranchikova A., Dmitrieva A., Fedorova M., Klimov A.
National Research University Higher School of Economics, Faculty of Humanities

Contacts: black-letter@yandex.ru

Introduction

This research is a part of the CAT&kittens project, which main goal is to develop a representative Russian Corpus of Academic Texts (CAT) outfitted with a built-in data processing tool, which allows for evaluation of texts written by novice writers of Academic Russian, both native and non-native, along a set of criteria in relation to the CAT corpus. Our current research is mainly concerned with collocational analysis and automatic collocation suggestion.

CAT was collected by extracting recently published texts sourced from textbooks, academic journals, and high-quality master's theses from available sources. All texts in CAT are divided into six disciplinary fields, amounting to appr. 2 million tokens in the corpus in general. CAT is supplied with metalinguistic information, as well as morphological and syntactic annotation, carried out with the help of Ru-Syntax (Mediankin et al. 2016). For collocational analysis, we have extracted domain-specific collocations for each sub-corpus and counted PMI, likelihood ratio and t-score for each of them.

In this paper we describe our approach to identifying miscollocations and automatic generation of possible replacements.

First approach: neural network language models

First method, which we tried for both finding miscollocations and suggesting alternatives from the reference list, was using neural network language models. We tested this method on the history sub-domain of the social studies and history sub-corpus. We used word lemmas as tokens and trained a recurrent neural network with an embedding layer, using one-hot encoding for the words from our texts. The model was supposed to store the probabilities of any of the known words to appear after the particular n input words (we tried bigram and trigram models). Each word n-gram of the input text was compared to the model, and if, according to the model, its probability was too low (different baselines were used), an alternative, more probable next word was suggested based on model's predictions.

However, even with lower sequence probability baselines, we found the model to consider too many input text n-grams incorrect, and a lot of suggested alternatives were not context appropriate. At this point, we decided to change our approach and consider finding miscollocations and suggesting the right alternative as two separate tasks. We also chose to use the linguistics domain as data because of us being more familiar with this topic, and work only with noun-verb, verb-noun, noun-noun and verb-verb collocations.

Word2Vec model and finding miscollocations

For this project we needed word2vec bigram model to use for both finding miscollocates and proposing their substitutes. This model was taught on 3040 linguistic texts (ca. 6.9 mln tokens) crawled from Cyberleninka, biggest Russian online resource containing millions of scientific papers. We made basic preprocessing including lower casing, deletion of non-cyrillic and non-alphanumeric characters. Dots were left in the texts, as well as other sentence terminators like exclamation or question marks were converted to dots to build our bigram model sentence-wise. As prepositions could be a substantial part of a collocation, we also did not delete stopwords from our texts.

In most of the cases, the task of finding miscollocations involves two steps: 1) checking each bigram against a dictionary of correct collocations; 2) if the bigram was not found in the dictionary, finding closest synonyms for each word and checking collocations with those synonyms. However, our reference collocation list is not big enough for this approach, so we tried to develop an additional approach to determining whether the collocation is correct. We had an idea that the task of choosing the most appropriate substitutions can be considered as a classification task where the classes are right and wrong collocations. We used collocations from the corpus as right examples and randomly generated pairs of words from the same corpus as wrong. It was suggested that Word2Vec similarity between words in the pair can be used as a feature, but it could not: the similarities had no correlation with the classes.

Correct Collocation Suggestion

Once a collocation is determined to be incorrect, a correct version needs to be chosen from a corpus of domain-specific collocations. Suggestion candidates are all collocations that have one of the words in common with the input and same tagset. In order to choose the best suggestions, features similar to those in (Liu et al., 2009) were used:

- Pointwise Mutual Information (PMI) for each candidate collocation. PMI was pre-calculated on all sub-corpus texts as in Manning and Schütze:

$$I(w_1, w_2) = \log_2 \frac{P(w_2 \mid w_1)}{P(w_1) * P(w_2)}$$

- Semantic similarity between the collocates (computed using Word2Vec model described above);
- The percentage of shared collocates in collocation cluster. Collocation clusters are sets of collocations that carry similar meaning and shared collocates (Liu et al., 2009). For example, if we get a N_0-V_0 collocation as an input and want to find substitutes for N_0 that V_0 forms correct collocations with, we take (N_1...N_i) possible substitutions for N_0 that collocate with V_0, and (V_1...V_j) substitutions for V_0. Then, for each N from (N_1...N_i) we obtain a set of verbs (V_2...V_k), and rank the substitution candidates (N_1...N_i) according to the percentage of similar words in (V_2...V_k) and (V_1...V_j).

Those features are then integrated into a probabilistic model, and the conditional probability of each case is then calculated as follows (assuming feature independence), where S_c is the situation where c is a correct substitute and F_c,m means the feature value between misused words and candidates:

$$P(S_c \mid F_{c,m}) = \frac{P(F_{c,m} \mid S_c)\,P(S_c)}{P(F_{c,m})} \approx \frac{\prod_{f \in F_{c,m}} P(f \mid S_c) P(S_c)}{\prod_{f \in F_{c,m}} P(f)}$$

As suggested in (Liu et al., 2009), we suggest 10 results with highest conditional probability as possible substitutions.

In general, 10 best suggestions are quite reasonable. For example, a stylistically inappropriate miscollocation "автор думает" gets the following suggestions: "автор выделяет", "автор предлагает", "автор выражает", "автор придерживается", "автор публикует", "автор апеллирует", "автор наталкивается", "автор посвящает", "автор сосредоточивает", "автор стремился".

Discussion

This research is a work in progress. We plan to further develop the miscollocation detection algorithm, probably using learner corpora as datasets. Upon completion of this tool, we will evaluate the results with the help of human assessors with advanced knowledge of Academic Russian language.

Acknowledgements

We would like to thank our academic advisors, Svetlana Toldova, Natalia Zevakhina, Olesya Kisselev, Mikhail Kopotev, and our machine learning teacher, Ekaterina Chernyak, for their supervision and advice on this project.

References

1. Mediankin N., & Droganova K. (2016). Building NLP Pipeline for Russian with a Handful of Linguistic Knowledge. In: Proceedings of the Workshop on Computational Linguistics and Language Science, pp. 48-56.
2. Rodríguez Fernández S. et al. Collocation and collocation error processing in the context of second language learning : дис. – Universitat Pompeu Fabra, 2018.
3. Liu A. L. E., Wible D., Tsao N. L. Automated suggestions for miscollocations //Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications. – Association for Computational Linguistics, 2009. – С. 47-50.
4. Futagi Y. et al. A computational approach to detecting collocation errors in the writing of non-native speakers of English //Computer Assisted Language Learning. – 2008. – Т. 21. – №. 4. – С. 353-367.