Elaborating Russian spelling-correction algorithms with custom n-gram models

Mikhail Minin, Olga Mitrofanova
Saint-Petersburg State University, Russian Federation

https://doi.org/10.36505/ExLing-2023/14/0018/000612

Abstract

The aim of the paper is to compare and improve the effectiveness of different approaches to the task of automatic spelling correction of Russian social media texts in terms of qualitative evaluation of results and algorithm complexity.

Keywords: Russian corpora, spelling correction, *n*-gram models

Problem statement

Typos are the simplest type of spelling errors. Apart from the obvious task of text correction when entering search queries in the information retrieval systems and exchanging messages in social media, an important area of application of automatic spelling correction algorithms is the preparation of text corpora for NLP tasks, such as building language models or improving the quality of speech recognition.

Currently, there are many approaches to solving the problem of automatic spelling correction, but most research has been done for English. The differences between English and Russian at all language representation levels lead to the necessity of developing a linguospecific approach and conducting independent experiments in the task of correcting typos in Russian texts. The data for spelling correction algorithms includes text corpora collected from social media which are used for training language models. Corpus quality affects its preprocessing, annotation and application in NLP tasks. An important aspect of developing an automatic spelling correction system is the amount of training data: large datasets are required for machine learning. Internet allows to collect large text corpora, the size of datasets determines the choice of NLP algorithms and their speed.

The aim of the paper is to evaluate different automatic spelling correction algorithms for Russian in terms of resulting quality and complexity.

Approaches to automatic spelling correction in Russian

All words to be corrected by the spelling correction algorithm can be divided into two classes: out-of-vocabulary words and vocabulary words with typos. In the first case, spelling correction implies the most probable replacements based

on spelling rules, Levenshtein distance and keyboard or phonetic similarity of characters. In the second case, when a word is still allowed by the rules, e.g., single-character words '*а*' ('*but*') and '*в*' ('*in*', '*into*'), the only way of correction is a context-dependent algorithm which may be based on the *n*-gram model calculating co-occurrence frequency of sequences including *n* elements.

In the task of correcting typos for single words, the main metric of word similarity is the edit distance or Levenshtein distance − the minimum number of single-character insertions, deletions and substitutions (Levenshtein 1965). In applying a context-dependent algorithm, spelling variants are compared as regards their semantic similarity to determine which word is closer to the context by its lexical meaning. The *n*-gram model is used to estimate semantic similarity of words.

Experimental setup

The dataset used in the study includes the corpus of LiveJournal blog texts consisting of 2008 sentences (https://www.dialog-21.ru/media/3838/test_sample_testset.txt] and a reference corpus with manual correction (https://www.dialog-21.ru/media/3835/corr_sample_testset.txt). The initial accuracy of the dataset – the ratio of the number of tokens without typos to the total number of tokens – is 90%.

For our experiments from the diversity of approaches we selected two algorithms, namely, Norvig's algorithm and Symspell algorithm.

Norvig's algorithm (http://norvig.com/spell-correct.html) belongs to the class of context-free error models using Levenshtein distance and a frequency dictionary. For each word, all possible words with Levenshtein distance below 2 are considered and checked for their presence in the frequency dictionary. Among the variants found, the word with the highest frequency is chosen.

As an alternative to Norvig's algorithm, we considered Symspell algorithm (https://github.com/wolfgarbe/SymSpell), which is also based on Levenshtein distance, but works with an extended dictionary. For each word, all variants of words obtained from the original word by removing one character with a reference to the index of the original word are also stored. Comparison is made between them and variants with similar deletions from the candidate word.

We proposed and tested the context-aware extension of the given approaches with a bigram language model. In addition to basic operations performed by Norvig's and Symspell algorithms, the bigram model chooses the closest correct pairs for each pair of adjacent words with possible typos.

Results

Results of experiments are defined in terms of classification errors (error matrix and the related parameters accuracy, precision, recall and F1-measure).

Results of SpellRuEval competition (https://www.dialog-21.ru/en/evaluation/2016/spelling_correction/) were used as a baseline. Norvig's algorithm implemented in pyspellchecker library (https://pypi.org/project/pyspellchecker/) was tested with two Russian frequency dictionaries consisting of 160K and 1 mln word forms correspondingly (https://opencorpora.org/?page=downloads). As the dictionary size increases, recall improves significantly, which corresponds to the intuition that more correct words are now included in the dictionary and are not distorted by the algorithm. However, the accuracy (81%) remains the same, hence, for this algorithm this value is close to the maximum achievable value.

Symspell algorithm implemented in SymSpellPy library (https://github.com/wolfgarbe/SymSpell) was tested with the same dictionaries. As Symspell has a rich potential in working with *n*-gram language models, it gains in operation speed as compared with Norvig's algorithm. Further, Symspell algorithm was tested with the language model including 900K bigrams. The original unigram dictionary was kept and used to identify and correct words written with hyphens. Thus, due to the extension Symspell uses a combination of the error model and the language model. Results outperform SpellRuEval in accuracy (best 71% vs. our result 91%) and approach in F1-measure (best 76%, 2nd result 65% vs. our result 71%).

Among the considered methods of automatic spelling correction, the combined method based on Levenshtein distance for unigram and bigram models showed the best result (F1-measure 71%, accuracy 91%).

Error analysis

We marked up a corpus with erroneous corrections and a reference corpus using pymorphy2 morphological analyser (https://pymorphy2.readthedocs.io/en/stable/; Korobov 2015). Then we calculated the distribution of tokens by parts of speech in both corpora and extracted items for which the spellchecker produced wrong results.

Comparison of erroneous corrections in different morphological classes provides evidence that the maximum proportion of mistakes is found in out-of-vocabulary items (up to 30%). In other words, the algorithm handles vocabulary items satisfactorily and makes mistakes on rare or occasional items. It is correct to compare the given lists of ratios, as for Symspell with the bigram model we obtained the same accuracy value as for the uncorrected text, i.e. the number of typos in both cases is approximately equal.

Conclusion

In course of experiments we obtained evidence that it is necessary to use dictionaries of sufficient size to test words for typos when using the context-

free algorithm. When the dictionary size increases to 1 mln word forms, the number of false corrections decreases.

A more accurate result is obtained when considering Levenshtein distance that does not exceed 1. It improves correction accuracy and gets it in line with the result obtained by (Panina et al. 2013) which shows that most of the sequencies with errors (83.6%) in search queries contain only one error.

Data structuring affects the time complexity of an algorithm. It is possible to increase the speed of dictionary search by a million times (https://github.com/wolfgarbe/SymSpell), as for large Levenshtein distances, e.g., in case of *n*-grams search.

The quality of spelling correction has a significant impact on the corpus tokenization procedure. For information extraction, it is important to correctly identify tokens and parts of speech of those words (nouns, verbs, adjectives) that are included in the output of keyphrase and named entity extraction algorithms. According to error analysis results, the proportion of errors in these classes decreases after processing by the algorithm. After correction, the largest proportion of typos is in the class of out-of-vocabulary items.

Prospects for improvement are as follows: enhancement of spelling correction accuracy, implementation of Levenshtein distance with a weight corresponding to phonetic coding, as well as the trigram language model, can be additionally used in further experiments.

Acknowledgements

Research is performed with support of RSF grant № 21-78-10148 «Modeling the meaning of a word in individual linguistic consciousness based on distributive semantics».

References

Korobov, M. 2015. Morphological Analyzer and Generator for Russian and Ukrainian Languages. In: International Conference on Analysis of Images, Social Networks and Texts AIST 2015, CCIS, volume 542, 320–332.

Levenshtein, V.I. 1965. Binary codes with correction of dropouts, insertions and substitutions of characters. In: Reports of the USSR Academy of Sciences 163:4, 845-848.

Panina, M.F., Baytin, A.V., Galinskaya, I.E. 2013. Context-independent autocorrection of query spelling errors. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue" (Bekasovo, May 29 – June 2, 2013), vol. 1, 12(19), vol. 1, 556-567.