Experiments with Automatic Spelling Correction of Russian Google ngrams

Marina Mitrofanova
National Research University
Higher School of Economics
School of Linguistics
Moscow
mamitrofanova@edu.hse.ru

Abstract

In contrast to the modern orthography older Russian writing systems has more alphabet letters which are often confused by Google's OCR and words that contain them don't appear in the search results among Google ngrams as if they never existed or were not in use. The paper discusses the problem of reconstructing these words with the help of spelling correction techniques in the case of out-of-context unigrams. In condition of the lack of the sufficient size of the dictionary we are using a language model to obtain the probabilities of character sequences being a part of a language in order to ignore non-dictionary words that actually do exist in a language but did not appear in our dictionary. This helps us to achieve higher precision score in OCR error corrections.

1 Introduction

Being one of the most important problems of computational linguistics automatic spelling correction is addressed to solve a variety of practical tasks from orthography correction in text editors to query completion and text anticipation in messengers (Jurafsky and Martin, 2018). This article introduces another problem for automatic spellchecking addressing the task of correcting the mistakes of the OCR by reconstructing the original words from the scanned and digitized Google books. The mistakes observed in our case are spelling errors in Russian Google ngrams that are connected to incorrect recognition of Russian pre-reform orthography which was in use before the orthographic reform of the 1918 year. The goal of the project is to contribute to the field of Culturomics and to reconstruct Russian Google ngrams with pre-reform orthography so that they could be searchable and representative for further statistical analysis.

We believe that it is possible to reduce the algorithms false positive results with the help of the language model by setting the threshold of the sequence probabilities given by the model. Words with probabilities of belonging to a language above this threshold will not be corrected.

2 Research

Spelling correction is considered from several perspectives. There is non-word spelling correction and real-word spelling correction. Non-words are errors resulting in non- or out-of-vocabulary words. The usual approach is to generate candidates from the dictionary (the larger the better) with a certain distance (usually Damerau-Levenshtein (Damerau, 1964) distance which measures the minimum number of

single character insertions, substitutions and deletions or sometimes transpositions) from the error and rank them according to this distance metric and candidate probabilities in context of the error which are obtained by building confusion matrices with the help of the manually corrected data collected in advance. (Jurafsky and Martin, 2018). Real-word errors are errors that result in actual words of the language we apply a spellchecker to. In this case we assume that any word in a sentence can be an error so we generate candidates for each word and rank them according to their likelihood given by our noisy channel model and our language model (Jurafsky and Martin, 2018).

Then there's correction of a word in a sentence and out-of-context spelling correction. In the first instance it is always reasonable to use a language model to calculate the likelihood of getting a particular sequence of words (Brill and Moore, 2000), (Sorokin and Shavrina, 2016). In the second instance it is harder to make a decision. The best candidate can be chosen just by relying on the noisy channel model and the word frequencies (Jurafsky and Martin, 2018). That is exactly the case in our project. The next section describes the task and the resources.

The task of correcting the character sequences can be also formulated as string-to-string transduction where the goal is to map an input string to an output string. With this, such technique as finite-state transducer (FST) can be applicable to the task or more sophisticated machine learning methods like character-level attention (Wu et al., 2018; Watson et al., 2018). As some characters in Google ngrams are constantly misrecognized by others (see Figure 1) with enough manually-edited examples the problem of word reconstruction could be solved with the neural models.

(Shillingford and Jones, 2018) introduces a slightly similar problem of reconstructing the language orthography where the task is to recover missing characters in old Hawaiian writing. It is solved with the help of FST combined with a language model. As we do not have a large training set of string mappings we are limiting ourselves to traditional approaches to automatic spelling correction.

3 Project Description

The main goal of the project is to contribute to the field of Culturomics that is to reconstruct Russian Google ngrams with pre-reform orthography and to publish open source results so that they could be searchable and representative for further statistical culturomics analysis.

3.1 Old and Modern Orthography

We will briefly introduce the changes in Russian orthography to give a picture of the cause and consequences of the problem discussed.

Pre-reform Russian orthography was in use before the Russian orthographic reform of 1917-1918 took place and introduced changes into the Russian language. These changes included the elimination of several letters from the alphabet and the spelling of ubiquitous affixes. These letters (see Figure 2) are not recognized right by Google ngrams OCR system (they tend to be recognized as symbols that are most similar to them as seen in Figure 1 which causes the loss of a huge amount of data in the corpus of the period before the mentioned date and consequently prevents a reliable culturomics analysis using this data.

3.2 Culturomics

Culturomics is the application of large data collection to the quantitative analysis of human culture which enables to investigate cultural trends quantitatively (Michel et al., 2011). A greater interest in this field has been aroused since the publication of the Google Books Ngram corpus in 2009. This made it easier for many researchers to use statistical methods, as in Google Books Ngram Viewer, for analyzing frequencies of groups of corpus ngrams and connecting significant statistical outliers to real-world events (Stergiou and Tsikliras, 2014).

3.3 Data Sources

3.3.1 Ngram Corpus

The Google Books Ngram Corpus has been available at http://books.google.com/ngrams since 2010. It consists of words and phrases

real characters	errors
ѳ	е-о
і	1-г
ні-ій-ін	ш
ѣ	ь-т
ҙ	з-8

Figure 1: The common mistake patterns of Google ngrams OCR system

Pre-reform spelling	Modern spelling
I i (i desiaterichnoe)	И и
Ѣ ѣ (iat')	Е е
Ѳ ѳ (fita)	Ф ф
IV/Ѵ (izhitsa)	И и

Figure 2: Correspondences between obsolete letters and their modern counterparts

(ngrams) and their usage frequency over time with syntactic annotations for eight languages and has enabled the quantitative analysis of linguistic and cultural trends as reflected in millions of books written over the past five centuries(Lin et al., 2012). The corpus has emerged from Google's effort to digitize books. Most books were drawn from over 40 university libraries around the world. Each page was scanned with custom equipment and the text was digitized by means of optical character recognition (OCR) (Michel et al., 2011).

Only Russian Google unigrams were taken into consideration for this project and only words with the found year before 1918 as the main purpose of the project is to correct errors connected with pre-reform orthography recognition.

3.3.2 Training Data

The model was trained on a dictionary built from the texts written in pre-reform Russian orthography since we don't need a context for candidate generation only the examples of language units. A corpus of texts for building the dictionary of pre-reform Russian language was collected from wikisource.org, arhivarij.narod.ru and russportal.ru which amounted to the size of 34650505 tokens, where the unique words amount to 371468. We also had a dictionary with ordinary orthography from dikmax.name and ruscorpora.ru with 2844497 unique word forms. The dictionary was stored as a prefix tree to allow effective word search with Levenshtein distance more than one from the error word.

4 Our System

In this section we describe our approach[1] to correcting Google unigrams.

4.1 Levenstein and Noisy Channel Model

As was mentioned above only words before 1918 were taken into consideration. Only non-dictionary words were analysed to choose the optimal distance metric and to build a Noisy Channel model. The words with less than 5 characters were ignored as they are often too short to make a decision about the right correction and they generate more candidates.

The development set consisted of 1000 examples while the test set amounted to 500 examples. The analysis of the development set showed the following statistics:

[1]Code available at https://github.com/kak-to-tak/Google_rusngram_spellcheck

raw ngram	year	match count	volume count	idx	ngram	new idx	is bastard	new ngram	desicion
Астроном1я_NOUN	1779	1	1	a	Астроном1я	a	1	Астрономія	corrected
Адриано	1830	1	1	a	Адриано	a	0	Адриано	correct
Аиз1гаНап	1905	1	1	a	Аиз1гаНап	a	1	Australian	translit
АРОДНОЕ	1885	2	2	a	АРОДНОЕ	n	1	НАРОДНОЕ	corrected

Table 1: An example sample of the Google unigrams checked manually

$$P(x|w) = \begin{cases} \frac{\text{del}[x_{i-1}, w_i]}{\text{count}[x_{i-1} w_i]}, \text{if deletion} \\ \frac{\text{ins}[x_{i-1}, w_i]}{\text{count}[w_{i-1}]}, \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, \text{if transposition} \end{cases}$$

Figure 3: Calculation of error probabilities with confusion matrices (Jurafsky and Martin, 2018)

58.04 % of errors have edit distance 1

21.68 % of errors have edit distance 2

9.09 % of errors have edit distance 3

11 % of errors have edit distance more than 3 (mostly transliteration)

Therefore we decided to generate candidates with Levenstein distance of 1 and 2 from the error word. Only three variants of errors were considered: deletion, insertion and substitution since transposition was highly unlikely to appear as an OCR mistake. Three confusion matrices were built for probabilities calculation of each possible error using the formulas shown above in Figure 3.

4.2 Language Model

In addition a language model was built to obtain extra information about a language the units of which were to be reconstructed. The model was trained on the dictionary of pre-reform Russian language. The dictionary-based model was chosen because it is smaller than the corpus-based language model and learns faster. As for the words frequencies, there was no need for a model to learn them as we could simply get them by counting the corpus tokens. We used the language model on the non-dictionary ngrams during the candidates generation to get the values which would indicate how likely it is that the word belonged to the language.

These probabilities of character sequences were calculated with the following way:

$$P(word) = \prod_{i=1}^{|word|} \frac{p(char_i \mid word[:i])}{|word|} \quad (1)$$

where $|word|$ is the length of a word and i is the index of a character.

We set upper and lower boundaries for the probabilities with such intuition that the sequences with high probability given by the language model were more likely to exist in a language and did not need to be corrected. Whereas words which emerged in Google ngrams as the attempt of an OCR to recognize non-Russian words as Russian, such as the following exmples:

Аидиз! August

Аиз1гаНап	Australian
аЬи1а	fabula
Ъау	bay
Ьагаге	Lazare
Ьайу	Lady
Ьапке	banke

are most unlikely to belong to the language. So if the probability of a word being a language unit is very low we are also not trying to correct it.

The boundaries are defined with the help of an analougue of a grid-search algorithm where we choose the values that give the best score, in our case precision.

4.3 Ensemble

We looked at the false negative errors of the algorithm and classified the them into four categories:

1) digit is inside the word;

2) digit is at the end of the word;

3) the word ends with 'ш' or 'щ';

4) the word's second to last letter is either 'ш' or 'щ' with the following 'и','ю','я' or 'е' representing the errors for the common Russian endings like 'ціи', 'цію', 'ція';

We than hardcoded the rules to correct the words which satisfy these conditions to solve the recall problem. The next section describes the obtained rates.

5 Results

We present results of three mentioned experiments:

1) Levenshtein distance with a Noisy Channel Model,

2) Applying the language model,

2) Applying both the language model and the rules.

The Levenshtein with a Noisy Channel Model is taken as a baseline. The results are evaluated on precision, recall and F1-score however our goal was to maximize precision with less attention to recall since we did not intend to replace the wrong words with other wrong words just to cover the recall. The results are presented in tables 2, 3, 4 below.

Precision	Recall	F1
0.24	0.91	0.38

Table 2: Experiment 1. Levenstein + Noisy Channel Model.

Precision	Recall	F1
0.8	0.03	0.06

Table 3: Experiment 2. Levenstein + Noisy Channel Model+Language Model

Precision	Recall	F1
0.81	0.11	0.2

Table 4: Experiment 3. Levenstein + Noisy Channel Model+Language Model+Rules

6 Conclusion and Future work

The paper described the experiment with automatic spelling correction of out-of-context words with low-resourced dictionary. Our application of a combination of a dictionary approach to correcting orthography with a language model drastically improves the precision along with significant losses in recall, thefore we added the rules for catching the most obvious-how-to-correct patterns. The best precision obtained is 81%.

In future work we are planning to experiment with higher-order n-gram and to make use of pentagrams first (as the context much simplifies the candidate choice) and descend by the ngrams counts towards unigrams after we have the corrected set of phrases. Also adding more data should significantly improve the results.

References

E. Brill and R. C. Moore. 2000. An improved error model for noisy channel spelling correction. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293.

F. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM 7(3):659-664*.

D. Jurafsky and J.H. Martin. 2018. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice-Hall.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Will Brockman Jon Orwant, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 169–174.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aide. 2011. Quantitative analysis of culture using millions of digitized book. *Science*, 331:176–182.

B. Shillingford and O.P. Jones. 2018. Recovering missing characters in old hawaiian writing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4929–4934.

A. Sorokin and T. O. Shavrina. 2016. Automatic spelling correction for russian social media texts. In *Proceedings of the International Conference "Dialogue 2016" Computational Linguistics and Technologies*.

KI Stergiou and AC Tsikliras. 2014. Global university reputation and rankings: insights from culturomics. *Ethics Sci Environ Polit 13:193-202*.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 837–843.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4425–4438.