Spelling Correction for Morphologically Rich Language: a Case Study of Russian

Alexey Sorokin
Moscow State University / GSP-1, Leninskie Gory, 1
Faculty of Mathematics and Mechanics, 119991, Moscow, Russia
Moscow Institute of Physics and Technology / Institutskij per., 9,
Faculty of Innovations and High Technologies,
141701, Dolgoprudny, Russia
alexey.sorokin@list.ru

Abstract

We present an algorithm for automatic correction of spelling errors on the sentence level, which uses noisy channel model and feature-based reranking of hypotheses. Our system is designed for Russian and clearly outperforms the winner of SpellRuEval-2016 competition. We show that language model size has the greatest influence on spelling correction quality. We also experiment with different types of features and show that morphological and semantic information also improves the accuracy of spellchecking.

The task of automatic spelling correction has applications in different areas including correction of search queries, spellchecking in browsers and text editors etc. It attracted intensive attention in early era of modern NLP. Many researchers addressed both the problems of effective candidates generation (Kernighan et al., 1990; Brill and Moore, 2000) and their adequate ranking (Golding and Roth, 1999; Whitelaw et al., 2009). Recently, the focus has moved to close but separate areas of text normalization (Han et al., 2013) and grammar errors correction (Ng et al., 2014), though the task of spellchecking is far from being perfectly solved. Most of early works were conducted for English for which NLP tasks are usually easier than for other languages due to simplicity of its morphology and strict word order. Also there were studies for Arabic (papers of QALB-2014 Shared Task (Ng et al., 2014)) and Chinese (Wu et al., 2013), but for most languages the problem still is open. In context of Slavic languages, there were just a few works including Sorokin and Shavrina (2016) for Russian, Richter et al. (2012) for Czech and Hladek et al. (2013) for Slovak.

However, spelling correction becomes actual again due to intensive growth of social media. Indeed, corpora of Web texts including blogs, microblogs, forums etc. become the main sources for corpus studies. Most of these corpora are very large so they are collected and processed automatically with only limited manual correction. Hence, most texts in such corpora contain various types of spelling variation, from mere typos and orthographic errors to dialectal and sociolinguistic peculiarities. Moreover, orthographic errors are unavoidable since the more social media texts we have, the higher is the fraction of those, whose authors are not well-educated and therefore tend to make mistakes. That increases the percentage of out-of-vocabulary words in text, which affects the quality of any further NLP task from lemmatization to any kind of parsing or information extraction. Summarizing, it is desirable to detect and correct at least undoubtable misspellings in Web texts with high precision.

Unfortunately, there were very few studies dealing with spellchecking for real-world Web texts, e.g. LiveJournal or Facebook. Most authors investigated spelling correction in a rather restricted fashion. They focused on selecting a correct word from a small pre-defined confusion set (e.g., *adopt*/*adapt*), skipping a problem of detecting misprints or generating the set of possible corrections. Often researchers did not deal with real-world errors just randomly introducing typos in every word with some probability. Therefore, spelling correction has no "intelligent baseline" algorithm such as trigram HMM-models for morphological parsing or CBOW vectors for distributional similarity. One of the goals of our work is to propose such a baseline. The principal feature of our approach is that it works with entire sentences, not on the level of separate words.

A serious problem for research on spellcheck-

Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, pages 45–53,
Valencia, Spain, 4 April 2017. ©2017 Association for Computational Linguistics

ing is the lack of publicly available datasets for spelling correction in different languages. Fortunately, recently such a corpus was created for Russian during SpellRuEval-2016 competition (Sorokin et al., 2016). Russian is rather complex for NLP tasks because of its developed nominal and verb morphology and free word order. Therefore it is well-suited for extensive testing of spelling correction algorithms, although our results are applicable to any other language having similar properties.

We propose a reranking algorithm for automatic spelling correction and evaluate it on SpellRuEval-2016 dataset. The paper is organized as follows: Section 1 summarizes previous work on automatic spelling correction focusing on context-sensitive approaches, Section 2 contains our algorithm, Section 3 describes test data, Section 4 analyzes the performance of our system depending on different settings and we conclude in Section 5.

1 Previous Work

Here we give a brief review of literature on spellchecking especially dealing with context-sensitive error correction.

- Edit distance model was introduced by Levenshtein (1966) and Damerau (1964), Kukich (1992) showed that about 80% of errors lie on distance of 1 edit.

- Weighted variants of error distances were considered in Kernighan et al. (1990) and Brill and Moore (2000).

- Toutanova and Moore (2002) added a pronunciation model for spelling correction, phonetic features were also exploited by Schaback and Li (2007).

- Noisy channel model of error correction based on ngrams appears in Mays et al. (1991) and Brill and Moore (2000). Other context-sensitive approaches include Golding and Roth (1999) and Hirst and Budanitsky (2005).

- Different sources of information were integrated by means of the final classifier in Flor (2012), who mainly uses semantic features, and Schaback and Li (2007), utilizing syntactic, phonetic and semantic information. Feature-based approach was also pursued by Xiong et al. (2014).

Since our method is also based on reranking, we compare it with the works of the last group. First, we work with sentences and consider each word as a potential typo while Schaback and Li (2007) and Flor (2012) try to correct isolated words using context features. To be applied to real-world texts their algorithm must be preceeded by a preliminary error detection stage which is not necessary in our approach. This makes the model more robust since error detection is a nontrivial task for social media texts due to high number of slang, proper names (including colloquial) etc. By its architecture our model more resembles Xiong et al. (2014), however, the set of features used differs significantly reflecting the difference between Chinese and Russian. As far as we know, our model is one of the first HMM-based systems used for spelling correction of a morphologically rich language.

There are also very few works dealing with spelling correction of Russian texts: Panina et al. (2013) uses feature-based approach to correct search queries. Works for other Slavic languages include Richter et al. (2012) for Czech, who used a feature-based method to correct errors in words given their context, and Hladek et al. (2013) who performed unsupervised error correction for Slovak. The present work is a part of ongoing research started by Sorokin and Shavrina (2016). The algorithm the latter is also based on reranking, however, they did not use morphological and semantic features. Actually, the effectiveness of these features was under question and one of the objectives of the work was to test their applicability in case of morphologically rich languages. We answer to this question positively.

2 Algoritm Description

Our system performs context-sensitive spelling error correction. The workcycle is divided into three main steps: candidate generation, n-best list extraction and feature-based ranking of hypotheses. Candidates are generated for every word in sentence since in real-world applications it is not known which words are mistyped. Pairs of consecutive words are also processed to deal with space insertion. There are four types of candidates:

1. Words from the dictionary on Levenstein distance of 1 from the observed word.

2. Words having the same phonetic code by the METAPHONE-style algorithm of Sorokin and Shavrina (2016).

3. Dictionary words or word pairs obtained by space/hyphen insertion/deletion. We also write several rules for candidate generation encoding frequent error patterns, for example the informal writing of ***-цца*** instead of ***-ться*** or ***-тся*** in the infinitive suffix (****нравицца*** ↦ ***нравится***).

4. A manually written correction list including colloquial writings as ****ваще*** ↦ ***вообще***, ****оч*** ↦ ***очень***.

Not all candidate words have the same score. We calculate the frequencies of different errors on SpellRuEval development set and set the probabilities of different error types (Levenshtein correction, phonetic correction, space insertion/deletion etc.) proportional to their frequencies. This constitutes the basic error model $P(t|s)$ for transforming the hidden word s into observed word t.[1]

We construct hypotheses for the whole sentence choosing one word from each candidate set and extract n best candidate sentences using beam search. To score the sentences we used noisy channel model $p(\mathbf{s}|\mathbf{t}) = p(\mathbf{t}|\mathbf{s})p(\mathbf{s}) = \prod_i p(t_i|s_i)p(\mathbf{s})$, where $p(t_i|s_i)$ is the probability of transforming the i-th aligned group in the hidden correct sentence to i-th group in the observed sentence and $p(\mathbf{s})$ is a trigram language model probability. Actually, this is a hidden Markov model (HMM) with word bigrams being the states of HMM and candidate words being the output symbols.

Since our error model does not take into account weights of different edits and other helpful linguistic clues, we rerank the hypotheses using features. Our feature set includes the following features:

- Length of the sentence, scores of original error and language models.

- Weighted edit distance between source and correction. The model was learned on the development set of (Sorokin et al., 2016) using the algorithm of Brill and Moore (2000).

- The total number and the number of corrections for out-of-vocabulary, long, short and capitalized words.

- The number of words that can be transformed into two dictionary words by space insertion and actual number of such corrections.

- The number of possible word pairs that can form a single word by space deletion or hyphen insertion and actual number of such corrections (hyphen errors are very common in informal writing).

- Morphological and semantic features (see extensive description in Section 4).

We also tried to implement more fine-grained features for hyphen and space insertion/deletion. For example, we counted the occurrences of the word ***по*** in the sentence and the number of words having ***по*** as its prefix as well as the number of hyphen insertions in such words/word pairs to reflect the common error pattern ***по-русски*** "in Russian" ↦ ***по русски*** or ***порусски***. However, most of such features appeared noisy in our experiments and were excluded from the final feature set. In total, our model includes 31 basic features, 9 morphological features, 6 semantic features and 1 morphosemantic feature – the unigram model score for the lemmatized sentence.

For every candidate sentence we obtain a feature vector with up to 47 dimensions. These vectors are ranked using a linear model returning the vector $\mathbf{u}_i$ with the highest scalar product $\langle \mathbf{w}, \mathbf{u}_i \rangle$. The weight vector $\mathbf{w}$ is learned using the method of Joachims (2006): in training phase we generate candidate sentences for each sentence of the training set; if $\mathbf{u}_0$ is the vector of the correct hypothesis and $\mathbf{u}_1, \ldots, \mathbf{u}_m$ of others, then the vectors $\mathbf{u}_0 - \mathbf{u}_1, \ldots, \mathbf{u}_0 - \mathbf{u}_m$ are assigned to the positive class and the opposite vectors to negative. Afterwards the weights can be learned by any linear classifier. We also experimented with the perceptron method of learning but the results were significantly worse.

3 Test Data

We used the development and test set of SpellRuEval contest (Sorokin et al., 2016). Development set consisted of 2001 and testing set of 2009 sentences respectively, taken from Livejournal segment of GICR corpus (Piperski et al., 2013). We refer the reader to the contest organizers paper for the full description of the dataset and just give a few examples:

[1] As usual in noisy channel models, the order of transformation is inversed in the error model.

1. Typos:
   *Программа **преложила** посмотреть, что получилось
   Программа предложила посмотреть, что получилось
   The program offered to see what happened

2. Colloquial writing:
   *а в результате в сумке кроме трусов и носков у меня больше **ниче** не лежало
   а в результате в сумке кроме трусов и носков у меня больше ничего не лежало
   As a result, there was nothing except underpants and socks in my bag

3. Space errors:
   *вот я и снова с вами к **сожелению не на долга**
   вот я и снова с вами к сожалению ненадолго
   I am again with you, but unfortunately, not for a long time

4. Hyphen errors:
   *фильм **помоему** очень реальный про настоящие чувства
   фильм по-моему очень реальный про настоящие чувства
   The film is very cool, I think, about real senses.

5. Real-word errors:
   ***пастель (pastel) очень уютная и мягкая но в то же время какая-то плотная***
   **постель очень уютная и мягкая но в то же время какая-то плотная**
   The bed is very soft and cosy but somehow dense

6. The combinations of different errors.

7. Correct sentences (799 of 2007).

Development set was used to train the reranker and to test hand-written rules of candidate generation. We built a trigram language model with Kneser-Ney smoothing using KenLM toolkit (Heafield, 2011). It was trained on the subset of GICR corpus containing 25mln words. The subset used for model training had no intersections with development and test sets. We also selected a 5mln word subset of this corpus to obtain cooccurrence counts and to investigate the dependence of performance quality from language model size.

The trigram model for morphological tags was trained on the subset of Golden Standard of GICR corpus,[2] the size of the training data was 10000 sentences. Instead of the full tags we used POS labels and selected grammemes: gender, number and case for nouns; gender, number, case, shortness and comparison degree for adjectives; mood for verbs and case for prepositions. Participles were considered as adjectives and pronouns as nouns or adjectives depending on their syntactic role. We used ABBYY Compreno dictionary containing about 3,7 mln word forms.[3]

We used logistic regression (though linear SVM showed almost the same results) for the final reranking, the implementation was taken from scikit-learn package (Pedregosa et al., 2011).

4 Results and Discussion

4.1 Comparison of Different Models

As our first experiment we compare 4 sets of features: WORD-LEVEL, including 31 features specified in Section 2; MORPHO, which also includes the morphological model score; SEM, extending WORD-LEVEL with semantic features and MORPHOSEM using both morphological and semantic information. For all 4 settings we run two experiments with different language models (trained on 5mln and on 25 mln words respectively). The morphological score is the negative log-probability of the sequence of morphological tags assigned to the words in proposed correction. We selected the most probable sequence considering all tags in the dictionary with equal probability. For the out-of-vocabulary words the tags and their probailities were guessed using simple suffix classifier.

Semantic scores were calculated from cooccurrence statistics. We calculated them as follows: first, all the lemmas of nouns, adjectives, verbs and adverbs appearing at least 100 times in our training data were selected. Then for every pair of such lemmas we calculated the number of times its members appear in the same sentence and kept all the pairs occurring at least 20 times. The set of pairs was pruned further: we kept w_2 as the potential pair of w_1 only if its probability to appear in the sentences containing w_1 is at least 3 times higher than its unconditional probability. From these statistics we extracted the following features

[2] http://www.webcorpora.ru/news/282

[3] http://www.abbyy.ru/isearch/compreno/, the dictionary itself is not open.

(w_2 is said to be a matching pair for w_1 if their pair is listed in the set of cooccurrence counts, lemma l_1 is frequent if it has at least one matching pair).

1. The number of words in the sentence whose lemma has a matching pair with some other word in the sentence.
2. Average number of matching lemmas for frequent lemmas in the sentence.
3. Maximal and average probabilities $p(l_2|l_1)$ for the lemma l_2 in the sentence to appear together with l_1 averaged over all l_1 in the sentence.
4. The number of frequent lemmas and whether the sentence contains at least one frequent lemma.

We compare our algorithm against the one of Sorokin and Shavrina (2016) – the top ranking system of SpellRuEval competition (BASELINE method). The results of our experiments are given in Table 1. Each row contains two subrows for smaller and larger language models. The following metrics are reported. They were calculated using the evaluation script of SpellRuEval-2016, for details refer to Sorokin et al. (2016).

1. Precision (the proportion of properly corrected tokens among all such tokens).
2. Recall (the fraction of misspelled tokens which were properly corrected).
3. F1-measure (the harmonic mean of precision and recall).
4. Accuracy (the percentage of correct output sentences).
5. The mean reciprocal rank (MRR) of correct output sentences and the number of times they appear in list of hypotheses (Coverage). Only the top 5 variants are taken into account.

Let T, F, W, M denote the number of exact corrections, the number of detected typos where the correction was wrong, the number of "false alarms", when a correctly spelled word was considered as typo and a number of missed typos, respectively. In this notation precision equals $\frac{T}{T+F+W}$ and recall is $\frac{T}{T+F+M}$. Therefore making an incorrect correction is worse than making no correction since both these operations decrease recall, but the former also affects precision. Hence we think that the percentage of correctly predicted sentences is more adequate as performance measure. It is also the objective maximized by the learning algorithm.

We give a detailed analysis of results in the next section. The preliminary conclusions are the following:

1. The size of the language model is the most significant factor affecting the algorithm performance.
2. Using the score of morphological model leads to significant improvement, reducing error rate by 8% in terms of F1-measure (84.24% instead of 82.87)and by 5.9% in terms of sentence accuracy (78.34% instead of 76.99%).[4]
3. Using semantic features further improves performance.
4. The impact of complex features is more significant in case of smaller language model. It is expected: the less data you have, the more complex algorithm you need to achieve the same level of performance.

4.2 Further Results and Discussion

Our results are rather convincing in order to prove that morphological and semantic features are useful for better spelling correction. However, they are still far from being perfect, therefore we should ask about further improvements that can be achieved on this way. At first, let us illustrate how morphological model helps to select a correct hypothesis. Consider the sentence ***к *сожаления, придётся постараться*** which should be corrected to ***к сожалению, придётся постараться*** ("it's a pity, (I) have to make an effort"). Lexeme ***сожаление*** ("pity") is erroneously written in its Sg+Gen form ***сожаления***, not Sg+Dat ***сожалению***. However, the preposition ***к*** requires a dative after it. On the level of morphological tags we have an erroneous sequence Prep+Dat Noun+Neut+Sg+Gen and a correct sequence Prep+Dat Noun+Neut+Sg+Dat. Since a dative preposition never has a genitive immediately to the right, the former sequence has much lower probability and is penalized by the ranker.

[4] For the larger language model.

Model	Precision	Recall	F1	Accuracy	MRR	Cov
BASELINE	81.98	69.25	75.07	70.32	NA	NA
WORD-LEVEL	88.62	73.17	80.15	74.35	81.09	90.54
	89.89	76.86	82.87	76.99	83.95	93.23
MORPHO	89.10	74.73	81.29	75.85	82.23	91.09
	89.35	79.69	84.24	78.34	84.81	93.28
SEM	88.48	73.77	80.46	74.65	81.30	90.34
	89.94	77.21	83.09	77.14	84.09	93.28
MORPHOSEM	88.86	75.34	81.54	76.20	82.44	91.19
	89.89	79.54	84.40	78.44	84.88	93.33

Table 1: Comparison of different feature sets using Sorokin et al. (2016) dataset.

Certailnly, it has lower probability by language model already, but this is not sufficient to make a correction since it is a dictionary word which is corrected. Indeed, most of the dictionary words in the sentence are spelled correctly which means that the number of corrections in dictionary words should be a negative feature. Therefore additional evidence is required to overcome this negative gain. Also morphological model is less sparser than lexical therefore it leaves less probability to unseen events which means the cost of unlikely sequence is much higher.

However, not all incorrect sequences of morphological tags can be rejected by trigram model only, especially in case of restricted set of tags, like we have. For example, in Russian each preposition restricts possible cases of its dependent noun. Most prepositions select only one case, for example, ***из*** "from" allows only genitive after it; other prepositions like ***за*** "besides" can govern accusative and instrumental cases, but rules out other 4 main cases. Nouns and adjectives in noun groups agree in case, number and gender; a verb agrees with its subject (usually noun or pronoun) in number and in gender (in past tense). All these dependencies are unbounded which means that an arbitrary number of words can separate two elements of the same phrase. However, the emerging constraints may be used to determine that, for example, a verb in particular position cannot be finite and hence reject or penalize a corresponding hypothesis of the spellchecker. That observation seems promising since confusion of 3rd person and infinitive forms of a verb is a common orthographic mistake (***мне нравится кофе*** "I like coffee" ↦ ****мне нравиться кофе***, where ***нравиться*** is the infinitive form).

Therefore we added 4 groups of features, 2 features in each groups, which contain the following counts:

1. The total number of prepositions and the number of prepositions which do not have a noun to the right which agrees with them.
2. The total number of adjectives and the number of adjectives which do not have a noun to the right which agrees with them.
3. The total number of infinitives and the number of infinitives which do not have a head (a predicative or a transitive verb).
4. The total number of indicative verbs and the number of verbs that do not a have a subject which agrees with them.

We hoped that these features would be helpful to improve our system performance further, but this was not the case. Encoding additional information deteriorated the quality, possibly due to overfitting. However, we observed that careful encoding of these features is impossible due to high morphological complexity of Russian. For example, nouns usually follow their attributes, but may also precede them (***лицо, красное от мороза*** "the face, red from frost"), subject is often only subsumed but omitted in the surface form or there is no subject at all like in impersonal sentences (***холодает*** get_colder+Pres+Sing+3 "it is getting colder"). Adverbs are often homonymical to grammatically correct prepositional phrases (***вправду*** "indeed" and ***в*** "in" ***правду*** "truth+Sg+Dat"), which forces the algorithm to oversegment them in order to increase the number of prepositions that agree with their nouns, etc. Summarizing, designing more complex morphological features requires additional research, probably in the framework of constraint grammars.

That is a nesessary step since among 559 sentences of the test set which were not properly corrected about 30 had an error in the verb form.

Even using only one morphological feature is not straightforward. Our reported results stand for the case when WORD-LEVEL model was trained first and the obtained score was used as a feature on the second step of the classification together with morphological model score. Otherwise error reduction is about twice less. The same happens with semantic features: trying to determine their weights together with word-level features, we obtain no gain at all. It implies that new features should be added hierarchically. In our best model semantics are added after learning the weight of morphology model.

During error analysis we have found that about one third of algorithm errors can be attributed as "semantical" which means that incorrect sentence cannot be rejected by morphological or statistical features since both variants are rare and belong to the same grammatical category. Often these are so-called "real-word errors", where the erroneous word is also in the dictionary. However, it is not trivial to extract a formal semantic score that favors one variant and refutes the other. Consider, for example, the mistyped sentence ***География его выступлений *достегает Китая и Индии*** "The geography of his performances *lashes China and India". Here the word ****достегает*** "(it) lashes" must be replaced by ***достигает*** "(it) reaches". A correction in the dictionary word is penalized, therefore there must be a valuable gain in language or semantic model score to compensate this penalty. But the verb ***достигать*** "to reach" does not cooccur frequently with other lexemes in the sentence like ***география*** "geography" and ***выступление*** "performance". The score of the language model is substantially higher for the correct variant, but it is not sufficient to compensate the correction in dictionary word. In this particular case additional preprocessing phase could be helpful since we might not have an exact phrase "***достигает Китая***" "reaches China" in our corpus, but certainly have other constructions of the form "***достигает*** Name_Of_Country". However, we do not have a ready implementation of this approach, but using class-based or factored language model together with some semantic classification seems a promising idea for further investigation.

Actually, morphological and semantic features are the instruments to remedy the weaknesses of n-gram language model, which is not powerful enough to discriminate between probable and unprobable sentences. Using more adequate language models might make fine-tuning of features unnesessary. A promising candidate to replace ngram models are neural language models (Mikolov et al., 2010) since they solve exactly the problem of choosing the optimal word in given context which is the main problem of spellchecking. We leave this question for future research.

4.3 Generalization of Results

Since lack of publicly available datasets is one of obstacles in spellchecking research, it is reasonable to ask to what extent our results depend on the size of the dataset and the source language. Table 2 shows the dependence between the size of development set used to tune the reranker weights and the quality of correction. We observed that even for the development set of 200 sentences (which is possible to collect and annotate manually) results are acceptable, though performance accuracy increases when we use more data. All results are averaged for 10 independent runs. Note that the gain from using more complex features increases with the size of development data which means that their weights are not tuned properly on smaller datasets.

Another question is whether our approach can be adapted to other languages. The architecture of the model is language-independent. Moreover, linguistically motivated features we design also are not specific to any language since they use only cooccurrence counts. Candidate search and some of word-level features encode language-specific information, but they reflect more the nature of Russian spelling errors in Russian, not the Russian word structure. Actually, a linguist can add any word-level feature; for example, instead of hyphen errors we may look for diacritic errors if the language uses diacritics, such as Czech. Our reranking model can also incorporate arbitrary sentence-level features reflecting morphological or lexical constraints. It makes our architecture perspective to design spellcheckers for other languages, not only for Russian.

Dev. set size	Model	Precision	Recall	F1	Accuracy
200	WORD-LEVEL	88.17	74.88	80.85	74.88
	MORPHO	88.19	76.06	81.66	75.70
	MORPHOSEM	87.30	76.35	81.44	75.44
500	WORD-LEVEL	89.15	75.49	81.73	75.65
	MORPHO	89.29	76.92	82.62	76.61
	MORPHOSEM	88.76	77.34	82.63	76.61
2008	WORD-LEVEL	89.89	76.86	82.87	76.99
	MORPHO	89.35	79.69	84.24	78.34
	MORPHOSEM	89.89	79.54	84.40	78.44

Table 2: Dependence of results on development set size.

5 Conclusions and Future Work

We develop a language-independent model for spelling correction and apply it to Russian language. Our algorithm outperforms the previous best system. Its another merit is flexibility that allows to incorporate arbitrary word-level and sentence-level features. Experimenting with features of different type, we observe that the main factor for spelling corrector performance is the quality of language model. However, morphological and semantic information is also helpful.

The direction of future work is three-fold: the first step is to augment traditional language models with neural ones and check whether this allows to deal better with long-distance dependencies which might be helpful in choosing the correct candidate. The second step is to apply our model to other languages with complex morphology and check whether the same features are beneficial as in case of Russian. The third one is to reimplement our model using finite-state tools since its main components (candidate search and their ranking) are actually finite-state operations.

Acknowledgements

The author is grateful to Andrey Sorokin and Ekaterina Yankovskaya for their careful help in preparing the paper. I also thank the anonymous BSNLP reviewers whose comments further improved the paper. The work was partially supported by the grant NSh-9091.2016.1 for leading scientific groups.

References

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

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.

Michael Flor. 2012. Four types of context for automatic spelling correction. *TAL*, 53(3):61–99.

Andrew R. Golding and Dan Roth. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine learning*, 34(1-3):107–130.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Graeme Hirst and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111.

Daniel Hladek, Jan Stas, and Jozef Juhar. 2013. Unsupervised spelling correction for slovak. *Advances in Electrical and Electronic Engineering*, 11(5):392.

Thorsten Joachims. 2006. Structured output prediction with support vector machines. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 1–7. Springer.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale. 1990. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 205–210. Association for Computational Linguistics.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):377–439.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707.

Eric Mays, Fred J. Damerau, and Robert L. Mercer. 1991. Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *CoNLL Shared Task*, pages 1–14.

Marina Panina, Alexey Baitin, and Irina Galinskaya. 2013. Context-independent autocorrection of query spelling errors.[avtomaticheskoe ispravlenie opechatok v poiskovykh zaprosakh bez ucheta konteksta]. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference“Dialogue”*, number 12, pages 556–568.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alexander Piperski, Vladimir Belikov, Nikolay Kopylov, Vladimir Selegey, and Serge Sharoff. 2013. Big and diverse is beautiful: a large corpus of russian to study linguistic variation. In *Proc. 8th Web as Corpus Workshop (WAC-8)*, pages 24–29.

Michal Richter, Pavel Straňák, and Alexandr Rosen. 2012. Korektor-a system for contextual spellchecking and diacritics completion. In *COLING (Posters)*, pages 1019–1028.

Johannes Schaback and Fang Li. 2007. Multi-level feature extraction for spelling correction. In *IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text Data*, pages 79–86.

Alexey Sorokin and Tatiana Shavrina. 2016. Automatic spelling correction for russian social media texts. In *Proceedings of the Annual International Conference “Dialogue”*, number 15.

Alexey Sorokin, Alexey Baytin, Irina Galinskaya, and Tatiana Shavrina. 2016. Spellrueval: the first competition on automatic spelling correction for russian. In *Proceedings of the Annual International Conference “Dialogue”*, number 15.

Kristina Toutanova and Robert C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 144–151. Association for Computational Linguistics.

Casey Whitelaw, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 890–899. Association for Computational Linguistics.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing*, pages 35–42. Citeseer.

Jinhua Xiong, Qiao Zhao, Jianpeng Hou, Qianbo Wang, Yuanzhuo Wang, and Xueqi Cheng. 2014. Extended HMM and ranking models for chinese spelling correction. In *Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP 2014)*, pages 133–138.