Normalization of Kazakh Texts

Assina Abdussaitova
Suleyman Demirel University
Computer Science
Kazakhstan, Kaskelen
assina.abdussaitova@gmail.com

Alina Amangeldiyeva
Suleyman Demirel University
Computer Science
Kazakhstan, Kaskelen
ali099838@gmail.com

Abstract

Kazakh, like other agglutinative languages, has specific difficulties on both recognition of wrong words and generation the corrections for misspelt words. The main goal of this work is to develop a better algorithm for the normalization of Kazakh texts based on traditional and machine learning methods, as well as the new approach which is also considered in this paper. The procedure of election among methods of normalization has been conducted in a manner of comparative analysis. The results of the comparative analysis turned up successful and are shown in detail.

1 Introduction

The Kazakh is a Turkic language which belongs to the Kipchak branch of the Ural-Altaic language family. It is an agglutinative language and differs from other languages like English in the way lexical forms are generated. Since the roots of Kazakh words may make thousands (or even millions) of valid forms which never appear in the dictionary, it has a complex structure such as inflectional and derivational morphology. The topic of analysis of Kazakh was not considered deeply enough; therefore, only a few works were accomplished in building tools in this field. Being one of the oldest problems in Natural Language Processing (NLP) with arguably the highest demand for a practical solution, automatic normalization is one of the necessary steps in text-processing for any language. This paper presents an approach for normalization in agglutinative languages that is based on a combination of error-detection, error-classification and ill-formed word correction methods that take advantage of statistical and rule-based approaches. Note that these developments also consider emoticons (emoji), stylistic uniquenesses (hashtag mention), mixed case problems and more. The main goal is to select the suitable normalization algorithm for Kazakh texts by comparison-analysis of Levenshtein and Naive-Bayesian algorithms for the case of spelling correction. Since the morphology of the Kazakh consists of unique features, the creation of a reliable model for text transformation, including standard dialects, slangs and emotional spelling errors, will also be a part of the problem. Today, there is no such data sources that can provide with non-dictionary words, except national historical texts and belles-lettres, that is why the new survey has been conducted. The poll has been held among 18-55 aged interviewers. As the survey itself, it has been divided into three parts:

- General questions about most frequently used Kazakh words
- Questions about local-area dialects and slangs
- Questions about wrongly carved words and shortenings

During this survey, the most commonly used words, including ill-formed and spoken words, were gathered. Moreover, the dataset has been collected by parsing websites with the massive amount of Kazakh texts such that commentaries, blog posts, quotes, articles and stories. Therefore, the dataset is provided not only by Kazakh national historical texts and poems, but also parsed comments from Kazakh websites, news blogs, and data gathered by questionnaire. In general, the approximate size of the dataset was about 110 thousand words, including 6% of the survey results, 72% of the parsed data and 22% of the dictionary words from literature and historical texts.

After performing a preliminary study of the normalization tools and Kazakh grammar with morphology, some problems of a misspelling for agglutinative languages in general and Kazakh, in particular, have been pointed out. Through the whole paper, the information about normalization

Proceedings of the Student Research Workshop associated with RANLP-2019, pages 1–6,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/issn.2603-2821.2019_001

technique, used approaches, obtained results have been considered, and analyses were conducted. For languages with a reasonably straightforward morphology recognition may be reduced to a trivial dictionary lookup: If a given the word is absent from the dictionary, then most likely it has an error. The classification algorithm is divided into two tasks: Error-type recognition and error correction. This process is done through passing the list of selected mistakes: Mixed/upper cases, hashtag mention, emoji, vowel repetition, consonant repetition, vowels absence and non-Cyrillic letters usage.

The contribution can be summarized in two ways: (i) the normalization system has been created for Kazakh texts by improving already existed spelling correction algorithms (ii) based on the methodology used, a website with normalization tool was developed.

The paper is organized as follows. Section 2 reviews related work; after that, the normalization system's algorithm for Kazakh is fully covered in Section 3. Analysis and evaluation are discussed in Section 4. Finally, conclusion and future works are described in Sections 5 and 6.

2 Related Work

There are many works performed on the general spelling correction problem. A lot of approaches were based on comparing a misspelt word with words in a lexicon and suggesting as possible corrections the ones with the minimal edit distance (Damerau, 1964; Levenshtein, 1966). Makazhanov and Makhambetov (Makazhanov et al., 2014) have researched spelling-correction by using the Levenshtein algorithm. According to them, there are two tasks for spelling-correction: Word recognition and error recognition. Hal and Baldwin (Han and Baldwin, 2011) also divided text normalization into two tasks: Ill-formed word detection and candidate word generation. A classical approach to spelling correction for agglutinative languages is to use FSAs (Alegria et al., 2008; Oflazer and Guzey, 1994; Pirinen et al., 2012). Oflazer and Guzey have presented a spelling correction algorithm for agglutinative languages by using finite state automata(FSA). In the proposed method, candidate words are generated using two-level transducers. To optimize the recognizer, the authors prune the paths that generate the substrings of the candidate words which do not pass some editing distance threshold. In a more recent work presented by Pirinen (Pirinen et al., 2012), the authors use two weighted FSAs one for language model and second for error model, where the authors reorder corrections by using POS n-gram probabilities for a given the word. Recently, another approach is often used (Church and Gale, 1991; Wood, 2013) that is based on applying a noisy channel model (Damerau, 1964), which consists of a source model and a channel model. These works differ in the way how authors weigh the edit operations and in context-awareness of the source models. Researchers Church and Gale (Church and Gale, 1991) utilize word trigram model, while Mays (Pirinen et al., 2012) do not consider the context. Later Brill and Moore (Brill and Moore, 2000) proposed an improved technique with more subtle error model, where instead of using single insertions, deletions, substitutions and transpositions, the authors model substitutions of up to 5- letter sequences that also depend on the position in the word. Hodge and Austin (Hodge and Austin, 2002) proposed an interesting method based on neural system AURA. They have employed two correlation matrix memories: one trained on patterns derived from handling typing errors by binary Hamming distance and n-grams shifting, and another trained on patterns derived from handling phonetic spelling errors. The list of suggested corrections is accomplished by choosing the maximum score obtained from the addition of the scores for Hamming distance and n-grams shifting with the score for phonetic modules. In 2018 Slamova and Mukhanova proposed the keyboard model of spelling correction for Kazakh which was based on replacement rules as a regular expression pattern (Slamova and Mukhanova, 2018).

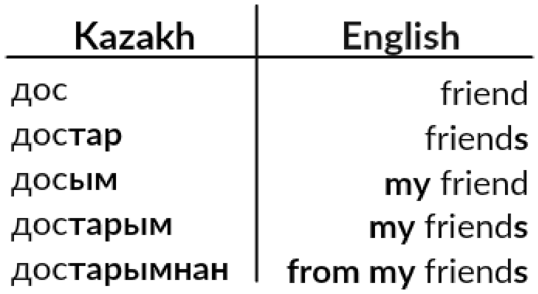
This paper differs in the way it does spelling corrections. The method for this was combined by mentioned above approaches: Levenshtein and Naive-Bayes. However, these algorithms were used not only by already suggested methods (word recognition, error recognition, ill-formed word detection, candidate word generation, FSTs) but also newly added classification algorithm's techniques to each approach that are further described in the section 3. Normalization algorithm itself is described by more error-types corrections. All in all, this paper focuses on three broad algorithms: Extended classical normalization, normalization based on the Levenshtein and Naive-Bayes algorithms. Each of them will be described in detail further and obtained during the paper results, which, obviously, are more accurate and stronger than in past methods, would be also described and compared.

3 Normalization Algorithms Methodology

Primarily the term normalization means not only spelling correction but also emotional letter repetition, specific characters or symbols use (such that ’@’, ’’), emojis and so on. The full list of error-types is shown below (word in Kazakh, English transcription, English translation):

- Mixed-case/Upper-case (’АспАН’, [’aspan’], ’sky’)
- Emoticons (’:-)’)
- Vowel repetition (’ааааспаан’, [’aspan’], ’sky’)
- Consonant repetition (’солллай’, [’solay’], ’so’)
- Absence of vowels (’жқс’, [’zhaqsy’], ’good’)
- Non-verbal symbols and characters (’@example’)

During the research, three main approaches have been used: Normalization based on the Levenshtein, Levenshtein with classification rules and Naive-Bayes algorithms. The main task was to compare three algorithms and select the better one, which is more appropriate for the particularities of the Kazakh language. The texts have been tokenized into words by using finite state transducer (Kessikbayeva and Cicekli, 2014) Implementation of FST has been applied by Foma programming language. Since the phrase generation for the Kazakh language differs from other languages, the syllabification based on FST was also used, which divides a word into the root and adjacent endings (Figure 1).



Kazakh	English
дос	friend
достар	friends
досым	my friend
достарым	my friends
достарымнан	from my friends

Figure 1: Features of word formation in the Kazakh in comparison with English

3.1 Normalization Based on the Levenshtein Algorithm

Since the Levenshtein algorithm is mainly presented as the spelling corrector, based on the minimal distance calculation, some problems like vowel repetition are usually wrongly corrected. Therefore it was decided to add three more steps before implementing it to improve the original model:

1. Classification
2. Preliminary correction of the error
3. Pretest of spelling

The first step is to recognize whether the word has an error or not and, if it does, to classify the type of error (one of six common types of failures listed in Section 3). It should be noted, that usually, there can be more than one error in a word, for example, the word ”OooOH” which means ”ten” and has two types of errors:

- Mixed case problem
- Emotional vowel repetitive

Therefore, this step returns the list of types of detected errors. After the classification step, the preliminary error-correction is triggered. This step replaces the corrected word according to the list of detected errors. The third step involves an initial check of the spelling of the word before passing the Levenshtein algorithm. If after the third step, the word is still ill-formed, the fourth step is triggered, which is the Levenshtein algorithm. Levenshtein algorithm is based on the distance between two strings source and target (Wood, 2013). The main idea is to measure the difference between two sequences. Mathematical interpretation of the Levenshtein distance is implemented as a matrix, where M(i,j) is the function that calculates the minimum value between the executed operations (Damerau, 1964).

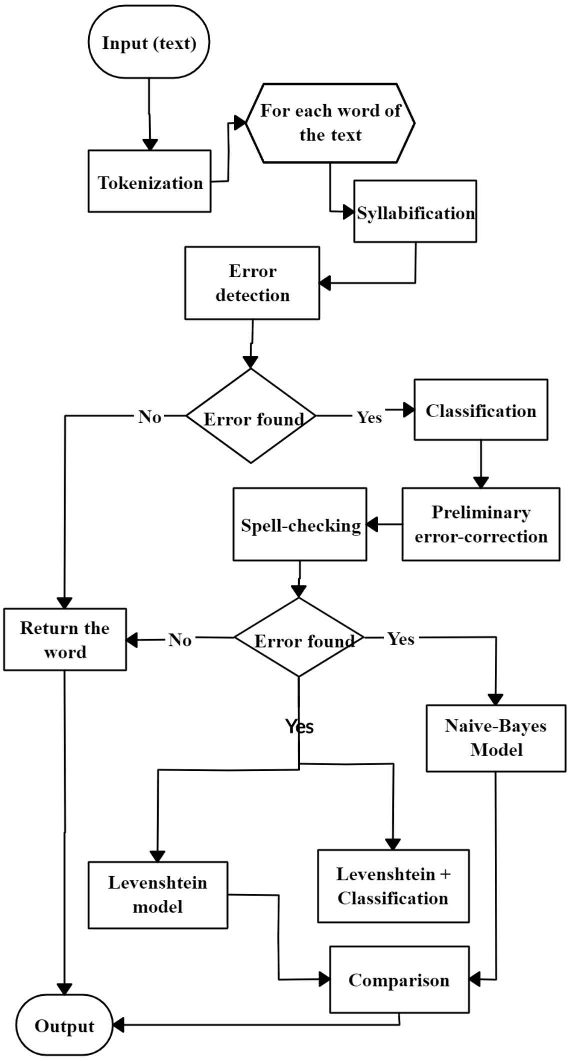


Figure 2: System architecture of the text normalization for Kazakh

When the distances for all targets are calculated, the next step is to choose the appropriate one. Formally, the shortest distance is selected as the best option.

3.2 Normalization Based on the Naive-Bayes Algorithm

The next algorithm is the Naive-Bayes, which is based on classifiers applying Bayes theorem with strong (naive) independence assumptions between the features. Bayes theorem describes the probability of an event based on prior knowledge of conditions that might be related to the event. Mathematically, the Bayes theorem is presented by the following expression:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Here $P(A|B)$ - is a conditional probability showing how often an event A occurs given that B occurs. $P(B|A)$ - is a conditional probability showing how often an event B occurs given that A happens, this is an error model which denotes a likelihood of B being transformed into A. $P(A)$ - is a source model that indicates how likely A is on its own. $P(B)$ - is characteristic for all suggestions denominator that shows how possible B is on its own.

The goal of this paper is to find the probability of correctness of a given the word. Since there is a list of candidate corrections, it could also be used. Suppose that the correction of A given the original word B is to be found:

$$P(A \mid B)$$

The correction of A should be found which has the greatest value of $P(A|B)$. By substituting Bayes theorem, this is equivalent to:

$$max_a \frac{P(B \mid A)\, P(A)}{P(B)}$$

The correction of A should be found, which has the greatest value of $P(A|B)$. By substituting Bayes theorem, this is equivalent to:

$$max_a \frac{P(B \mid A)\, P(A)}{P(B)}$$

Since $P(B)$ is the same for all kinds of correction, $P(B)$ can be eliminated, the simplified equation looks like:

$$max_a P(B \mid A) P(A)$$

The $P(A)$ is a probability that the proposed correction stands on its own. In this experiment, P(c) will be determined by word ranks in the dictionary. For example, the word "көк" (kok, blue) has a greater probability than "көктем"(koktem, spring) based on words' usage statistics.

$P(B|A)$ is a probability that B would be typed when the user meant A. Said, this is the probability of how likely the user would type B by mistake when A was intended.

The word with maximum probability from all possible words in the dictionary has been chosen. Of course, the word that is having edit distance greater than 1 has a probability of 0. In 3.1 and 3.2, only $P(A)$ was used to checking.

There are many factors of $P(B|A)$ that needs to be taken into account, but since some factors are

Model	***Leven-shtein***	***Leven-shtein + rules***	***Naive-Bayes***
Avg of properly corrected words	5,1	5,9	0,35
Avg of wrongly corrected words	6,15	4,75	0,2
Avg of unnecessary corrected words	6,1	1,4	0,05
Precision	93,58	96,85	99,19
Recall	46,36	56,42	81,33
f1 score	***62***	***71,3***	***89,38***

Table 1: Testing results of 3 models for normalization

not entirely independent (increasing probability of x may decrease the probability of y), the simple analysis through it was made.

4 Analysis

Since three approaches have been selected, there are three stages of testing: Levenshtein-based algorithm, Levenshtein plus classification rules based algorithm and Naive-Bayes based algorithm. Twenty different experiments for each model with a variety of cases have been conducted. Each test consisted of sentences with 20-25 words, 11 of which were ill-formed at the average, at which point the average precision and recall have been calculated. The average records for each testing section are shown in figure 3, where the f1 score is the accuracy of the considered models. The first block of the table describes the results of Levenshtein spelling-correction algorithm-based model. F1 score for this model turned out only 62% with 46,36% of recall value. The next model (Levenshtein + classification rules algorithm) showed up quite higher than the first - 71,3% of the f1 score and 56,42% of recall gives almost 10% breakaway from original Levenshtein algorithm.

Finally, the third model, which is based on Machine Learning Naive-Bayes algorithm gave the highest results compared with others. Its f1 score reached 89,38% which is 27,38% more than the original and 18,08% models. According to the precision values, Naive-Bayesian model is also the leader - 99,19%. One of the reasons for that lies on the large dataset, which was gathered by parsing websites and conducting the surveys. To compare, testing results of the model proposed by Slamova and Mukhanova has accuracy 85.4% (Slamova and Mukhanova, 2018).

5 Conclusion

In this paper, the normalization tool for Kazakh texts based on a Machine Learning algorithm has been developed. According to the results, this tool outperforms other analogs with spelling corrector based on Levenshtein-distance. Finally, the high overall accuracy in generating correct suggestions was received. The difference between normalization and spelling corrector lies in new added conditions and cases that expand the possibilities of normalization and increase the probability of words correctness. For instance, no research and tool consider the list of mistakes in the Kazakh language, which was suggested in this paper. The advantage of the proposed new method is that it can be iteratively improved by adding new rules/transitions to the normalization and new entries to the root lexicon. Moreover, the Bayesian approach, which is the core of this method, can also be used for morphological segmentation.

6 Future Works

Some areas need to be considered deeper in the future. In particular, this is the complete data for Kazakh dictionary taken from common knowledge of people (related to mother language and geographical area), list of frequently occurring slang, specific words are still in progress and should be enlarged, since for Kazakh language there are no big data sources of training data as opposed to resources in English. Moreover, after the process of gathering data, it will be necessary to analyze and structure it. Also, many aspects are needed to be taken into account to improve the effectiveness of normalization, such as:

1. Number of common and obscure words in the dictionary
2. Type of keyboard and its distance between two specific characters
3. Edit-distance (greater than 1 or 0, even though edit-distance of 1 has covered at least 80 per cent of correctness probability)
4. Word structures (the Kazakh language has a big number of endings with different roots)

Another further research question will be about the combination of Levenshtein and Naive-Bayesian algorithms. The future work will be directed towards answering this question, as well as incorporating context sensitivity into the method used and improvements that could be applied based on this research work.

Acknowledgments

Special gratitude is expressed to those who contributed to this work through suggestions, criticism and review. Special thanks to anonymous reviewers, as well as Gulshat Kessikbayeva, a professor at Suleyman Demirel University, and Nazerke Sultanova, a teacher.

References

I. Alegria, K. Ceberio, N. Ezeiza, A. Soroa, and G. Hernandez. 2008. Spelling correction : from two- level morphology to open source. in: Lrec, european language resources association. 4:1051–1054.

E. Brill and R. Moore. 2000. Proceedings of the 38th annual meeting of the association for computational linguistics. Association for Computational Linguistics.

K. Church and W. Gale. 1991. *Probability scoring for spelling correction. Statistics and Computing*.

F. Damerau. 1964. A technique for computer detection and correction of spelling errors. pages 171–176.

B. Han and T. Baldwin. 2011. Lexical normalisation of short text messages. pages 368–378.

V. Hodge and J. Austin. 2002. A comparison of a novel neural spell checker and standard spell checking algorithms. pages 2571–2580.

Gulshat Kessikbayeva and Ilyas Cicekli. 2014. Rule based morphological analyzer of kazakh language. *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*.

V. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. 2:707–710.

A. Makazhanov, O. Makhambetov, I. Sabyrgaliyev, and Z. Yessenbayev. 2014. Spelling correction for kazakh. computational linguistics and intelligent text processing. pages 533–541.

K. Oflazer and C. Guzey. 1994. Spelling correction in agglutinative languages. 2:194–195.

T. Pirinen, M. Silfverberg, and K. Linden. 2012. Improving finite-state spell-checker suggestions with part of speech n-grams.

Gaukhar Slamova and Meruyert Mukhanova. 2018. Text normalization and spelling correction in kazakh language. In *AIST*.

Z. Wood. 2013. Profiling spatial collectives. research and development in intelligent systems. 2:102–103.