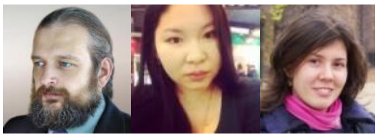
Stemming and word forms generation in automatic text processing systems in the Kazakh language

V Barakhnin[1, 2]*, A Bakiyeva[2], T Batura[2, 3]

[1]Institute of Computational Technologies of SB RAS, Lavrentiev av., 6, 630090, Novosibirsk, Russia
[2]Novosibirsk State University, Pirogov str.,1, 630090, Novosibirsk, Russia
[3]A.P.Ershov Institute of Informatics Systems of SB RAS, Lavrentiev av., 6, 630090, Novosibirsk, Russia

**Corresponding author's e-mail: bar@ict.nsc.ru*



Abstract

This article is dedicated to algorithms of stemming and word forms generation. The proposed algorithms are based on the principles of division the words into inflectional classes. As the Kazakh language is agglutinative, linking the word forms dictionary for the automation of morphological analysis is impractical. Using affix dictionary and sets of rules is much more effective. During the research the dictionary was developed. It includes about 2,000 verbal affixes and their combinations for the 17 inflectional classes and about 3,500 affixes and their combinations (variants of endings) for nouns and adjectives. Some combinations of affixes are repeated. This volume of the dictionary is sufficient to perform text analysis of any themes. The proposed algorithm can be applied at the stage of morphological analysis in the search engines, summarization systems and question-answer systems, as well as in the construction of thesauri and ontologies.

Keywords: stemming, generation, morphological analysis, affixes, inflectional classes

1 Introduction

Concerning with the expansion of the information space, there is an urgent necessity for automatic processing of texts in various languages, in particular, in Kazakh language.

The Kazakh language has a rich and complicated morphology. Like in other Turkic languages, any word consists of a stem, which affixes with different grammatical characteristics are attached. To the stem of a word several form-building affixes (sometimes called endings) can be attached, while each such affix fulfills a grammatical function inherent only in it, the order of affixes is strictly determined.

In the process of thematic indexing of document, a set of key terms is usually used to determine if it belongs to a particular domain, which denotes a concept from the given subject area, and the terms are found in different word forms. Consequently, when choosing the characteristics of documents, it is more correct to take into account not the word forms, but the stems of words, therefore, it is necessary to create a qualitative algorithm for stemming.

Thus, since the Kazakh language is agglutinative, it is more convenient to use the affix dictionary and sets of rules for both stemming and generation.

A distinctive feature of the proposed algorithms for stemming and generation of word forms of the Kazakh language is the use of the principle of words splitting into inflectional classes in accordance with the ideas of work [1]. For the implementation of these algorithms for all changeable parts of speech (noun, adjective, verb) our rule sets combining affixes have been described.

2 Inflective classes of nouns, adjectives and verbs of the Kazakh language

The basis for constructing algorithms for morphological analysis and synthesis is the division of all words into classes that determine the character of the change in the literal composition of word forms. These classes are conditionally called morphological. Changes in the form of words can have a different character. They can be related both to the change in the formative affixes of the word and its stems (which in the Kazakh language is extremely rare: for example, there are 18 exceptions for nouns, 352 for verbs).

Morphological classes of words are divided into two types [2]: the stem-changable classes that characterize the system of word changes, and the inflectional classes of words. The inflective classes of changeable words were distinguished on the basis of an analysis of their syntactic function and systems of case, personal and generic endings. The classes of uninflected words were distinguished only by the syntactic principle.

In this article, we will consider the inflective classes of the verbs of the Kazakh language in more detail, since the inflective classes of nouns and adjectives were described in [1]. Table 1 provides examples of the endings of inflectional classes for certain tense and person affixes. Following the rules of the grammar of the Kazakh language [3] for verbs, we established 17 inflective classes:

1. hard, ends to vowel (except **ю**);
2. soft, ends to vowel (except **ю**);
3. hard, ends to **б, г, ғ**;
4. soft, ends to **б, г, ғ**;
5. hard, ends to **з**;
6. soft, ends to **з**;
7. hard, ends to **р, л**;
8. soft, ends to **р, л**;
9. hard, ends to **м, н, ң**;
10. soft, ends to **м, н, ң**;
11. hard, ends to **ж, д**;
12. soft, ends to **ж, д**;

13. hard, ends to a deaf consonant;
14. soft, ends to a deaf consonant;
15. hard, ends to a **ю**;
16. soft, ends to a **ю**;
17. hard, ends to a **у**.

3 Algorithms for generation of word forms and stemming

A step-by-step description of the algorithm for generation of nouns is given in the article [1]. Similarly, word forms are generated for adjectives and verbs. The only difference is in the first step. The input is a noun or an adjective in the nominative case; the verb is in the form of an infinitive.

3.1 ALGORITHM FOR GENERATION OF VERBS

For verbs, there are the following types of endings (in parentheses, we denote each type of endings with the capital latin letter):

a. ending of negative (A);
b. tense ending (B);
c. pronoun ending (C);

The following combinations of endings are possible:

1. tense ending (B);
2. tense ending + pronoun ending (BC);
3. ending of negative + tense ending (AB);
4. ending of negative + tense ending + pronoun ending (ABC);

Once again, we note that the order of joining affixes strictly fixed and is conditioned by inflected class.

3.2 THE ALGORITHM FOR VERBS STEMMING

We state the algorithm for verbs stemming (the algorithm for stemming of nouns and adjectives is described in [4]). It is based on Porter's algorithm [5]. Depending on the conditions, a decision is made whether a word stem is obtained or the affix is cut off. All the necessary rules for transformations can be divided into groups according to inflectional classes. The algorithm for obtaining the stems consists of the following steps.

1. To the input arrives any word form (verb, noun, adjective).
2. Starting with the last letter of the word, the list of affixes is searched.
3. If this affix is found, then it is cut off. Otherwise, the remaining of the word is considered as the stem.

The order of the rules has the following steps:

Step 1 - ending of negative + tense ending + pronoun ending + plural ending (ABC). For example: «ма + ды + ңыздар».

Step 2 - - ending of negative + tense ending (AB). For example: «пе + ді».

Step 3 - ending of negative + tense ending + pronoun ending (ABC). For example: «па + ған + мын».

Step 4 - tense ending + pronoun ending (BC). For example: «ген + сің».

It should be noted that the proposed algorithms of stemming and generation are applicable to simple forms of verbs. More complex forms of verbs, consisting of 2-4 words, are planned to be considered in the future. Note that in scientific and technical texts complex verbs are practically not used.

4 Implementation and testing of proposed algorithms

Previously, a web application for generating of word forms for nouns and adjectives was developed and described in [1]. The implementation of the above algorithms for generating of verb forms and stemming of nouns, adjectives and verbs was added to this application. The generation module and the stemming module are implemented in Python using libraries: psycopg2, collections. The dictionaries are stored in the database PostgreSQL.

During the research, 14 inflectional classes were extracted for nouns and adjectives, and 17 for verbs. At the moment, a dictionary of exceptions is composed of 18 nouns, 352 verbs, in which the word forms are formed by changing the stem. The volume of the dictionaries is sufficient to perform text analysis of any themes.

We tested words belonging to different parts of speech and didn’t find any errors. This allows us to judge the correctness of the proposed algorithms.

5 Conclusions

Algorithms of stemming and generation of verbs are described in the article. This, together with the results of [1, 4], completely solves the problem of analysis and synthesis of word forms for scientific and technical texts. 17 inflected classes of verbs were described. The created affix dictionaries include more than 5,500 affixes and their combinations (taking into account duplicate combinations for different grammatical forms). We tested words belonging to different parts of speech and didn’t find any errors. This allows us to judge the correctness of the proposed algorithms.

The generation module and the stemming module are implemented in Python using libraries: psycopg2, collections. The dictionaries are stored in the database PostgreSQL.

Acknowledgments

Work is executed with partial support of the Presidential programme «Leading scientific schools of RF» (grant 7214.2016.9).

References

[1] Barakhnin V, Lukpanova L, Solovyov A 2014 An algorithm for constructing word forms using inflected classes for systems of morphological analysis of the Kazakh language *Bulletin of NGU, Series: Information technology* **12**(2) 25-31 (*In Russian*)

[2] Belonogov G, Zelenkov Y 1985 Algorithm for automatic analysis of Russian word *Questions of information theory and practice* **53** 62-93 (*In Russian*)

[3] Valiayeva T *The grammar of the Kazakh language* http://kaz-tili.kz (*In Russian*)

[4] Fedotov A, Tusupov D, Sambetbayeva M, Yerimbetova A, Bakiyeva A, Idrisova I 2015 The model determine the normal form of the word for the Kazakh language *Bulletin of NGU, Series: Information technology* **13**(1) 107-16 (*In Russian*)

[5] Porter M F 1980 An algorithm for suffix stripping Program **14**(3) 130–7

[6] Bakiyeva A *Program generation of word forms of the Kazakh language* http://poem.ict.nsc.ru/~bakieva_aigerim/kazGen/ *(in Russian*)

Automatic summarization

A Orynbayeva

Faculty of Information Technology, Kazakh-British Technical University, Tole Bi Str. 59, 00500 Almaty, Kazakhstan

**Corresponding author's e-mail: altynai_56@mail.ru*

Abstract

The volume of accessible data on the Web is expanding very quickly. The necessity of frameworks that able to work with those amount of information is becoming ever more desirable. Therefore, dealing with them, it can be beneficial to work with one field of natural language processing called summarization. Automatic summarization plays crucial role in compacting expansive amounts of text into effective summary. This study means to investigate some of the most relevant approaches both in the areas of single-document and multiple-document summarization, giving particular underline to some methods and extractive strategies.

Keywords: natural language processing, automatic summarization, single-document, multiple-document

1 Introduction

Natural language texts are the most common form of knowledge representation, which are easily perceived and interpreted by a human. However, the volumes of these texts have increased significantly and it is not easy to search, process and analyse their contents manually.

As a result, nowadays, there are many technological application which focuses on the analysing and understanding human languages and can be considered by big field named Natural Language Processing (NLP). For instance, entity linking and information extraction, sentiment analysis and opinion mining and also text summarization.

2 Goals

The goals of this study are to study a big area called "Natural Language Processing" and to provide a comprehensive overview of field within NLP named automatic summarization. Moreover, aims of this study to build web-based platform and use known tools and libraries for the reading of articles and documents, whether to compare similarity between articles and their shared keywords, to identify main problem there and to resume it. The idea is to construct the automatic summary from the information by analysis. The service will be built on Djangos framework and will be written with python and their useful library NLTK (Natural language toolkit).

3 Background

Summarization is the technique to reduce a text of document with lines of code based on machine learning or algorithms in order to create a summary which will retain crucial points of the original information. Although research on summarization started approximately 55 years ago, there is still a long trip to research in this field. Over time, attention has drifted from summarizing scientific articles, news, mail messages, blogs and medical sources.

These systems are designed to take information or url of interested website as input and to produce a concise summary of the most related points as output.

There are two methods to automatic summarization: extractive and abstractive ways. Extractive approaches select a subset of existing words or sentences from the original text to form the informative summary, whereas abstraction is important and active research area due to their complexity to research. Furthermore, abstractive method construct semantic representation and generate summary which is closer to human languages. The difficulty differential between these two approaches is greatly increased when the task is handed over to computers. Even with the current state of the art in artificial intelligence, computers are still not nearly advanced enough to support the ability to "reorganize, modify and merge information expressed in different sentences in the input." [4]

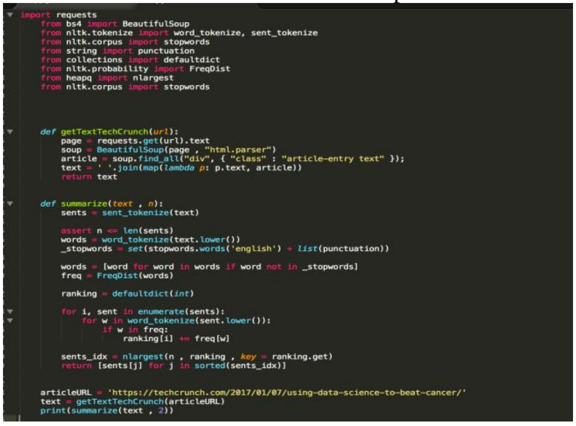
Methods of automatic summarization also divide based on the number of sources of information: single-document or multi-document.

4 Surveying the Field of Automatic Summarization

A summary must prioritize the most important themes, sentences. The well-known fundamental method determined by Luhn to identify significance of a sentence. [6] A word's significance is equal to its probability of occurring, which is defined by:

$$\text{significance}(w) = p(w) = c(w)/N, \tag{1}$$

where p(w) is probability of a word, w, occurring and c(w) is number of times a word, w, occurs in the input document and N is total number of words in the input.



```
import requests
    from bs4 import BeautifulSoup
    from nltk.tokenize import word_tokenize, sent_tokenize
    from nltk.corpus import stopwords
    from string import punctuation
    from collections import defaultdict
    from nltk.probability import FreqDist
    from heapq import nlargest
    from nltk.corpus import stopwords



    def getTextTechCrunch(url):
        page = requests.get(url).text
        soup = BeautifulSoup(page , "html.parser")
        article = soup.find_all("div", { "class" : "article-entry text" });
        text = ' '.join(map(lambda p: p.text, article))
        return text


    def summarize(text , n):
        sents = sent_tokenize(text)

        assert n <= len(sents)
        words = word_tokenize(text.lower())
        _stopwords = set(stopwords.words('english') + list(punctuation))

        words = [word for word in words if word not in _stopwords]
        freq = FreqDist(words)

        ranking = defaultdict(int)

        for i, sent in enumerate(sents):
            for w in word_tokenize(sent.lower()):
                if w in freq:
                    ranking[i] += freq[w]

        sents_idx = nlargest(n , ranking , key = ranking.get)
        return [sents[j] for j in sorted(sents_idx)]


    articleURL = 'https://techcrunch.com/2017/01/07/using-data-science-to-beat-cancer/'
    text = getTextTechCrunch(articleURL)
    print(summarize(text , 2))
```

It was one of the approaches to summary extraction. Summary extraction can be frequency based approach, feature based approach and machine learning based approach. Two techniques that use frequency based approach are word probability and term frequency-inverse document frequency.

One of the other way to determine the importance of a sentence is based on feature approach which reflects the relevance of that sentence that can be shown from sentence position, presence of title word and keywords.

In later works from journal od Computer Science, there was used Particle Swarm Optimisation (PSO) algorithm, genetic algorithm, differential evolution algorithm and fuzzy logic in order to enhance finding important sentence by combining term frequency weight with position and node weight [9].

Machine Learning (ML) approach need to have a set of training document (dataset) and their corresponding summary extracts. There are some well-known methods such as Naive Bayes Classifier and Markov Hidden Model, also Neural Network.

In using Naive Bayes method, there are given a sentences where the probability being chosen to be included in the summary is:

$$\mathrm{P}(\mathrm{s}\in S \mid F_1, F_2, \dots, F_n) = \frac{\prod_{i=1}^{n} \mathrm{P}\left(F_i \middle| s \in S\right) * P(s \in S)}{\prod_{i=1}^{n} P(F_i)}$$

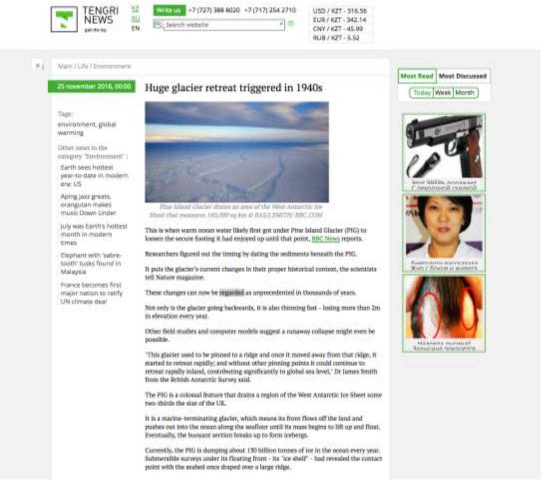
where F1, F2, …, Fn are the sentence features (assuming the features are independent of each other) for the classification and S is the summary to be generated.

Each sentence is then scored according to Equation 2 and ranked for summary selection [7].

Neural network have the advantages to learn summary sentence attributes. The network learn best features and patterns from training to determine the most important information.

5 Simple experiment

In order to better understand how work with NLTK library which is the leading platform for building Python programs to work with human language data [8] and their features, there was done a simple script to summarise the article from Tengrinews (https://en.tengrinews.kz/environment/Huge-glacier-retreat-triggered-in-1940s-263491/) which is kazakh news portal by url getting using BeautifulSoap based on frequency approach to generate summary. As a result, there have been summarised article from Tengrinews into several sentences which more relevant to the summary identifying important sentences. The text was as following: “"This glacier used to be pinned to a ridge and once it moved away from that ridge, it started to retreat rapidly; and without other pinning points it could continue to retreat rapidly inland, contributing significantly to global sea level," Dr James Smith from the British Antarctic Survey said. Currently, the PIG is dumping about 130 billion tonnes of ice in the ocean every year. Submersible surveys under its floating front - its "ice shelf" - had revealed the contact point with the seabed once draped over a large ridge.”



25 november 2016, 00:00

Huge glacier retreat triggered in 1940s

Tags:
environment, global warming

Other news in the category "Environment" :

Earth sees hottest year-to-date in modern era: US

Aping jazz greats, orangutan makes music Down Under

July was Earth's hottest month in modern times

Elephant with 'sabre-tooth' tusks found in Malaysia

France becomes first major nation to ratify UN climate deal

Pine Island Glacier drains an area of the West Antarctic Ice Sheet that measures 160,000 sq km © BAS/J.SMITH/ BBC.COM

This is when warm ocean water likely first got under Pine Island Glacier (PIG) to loosen the secure footing it had enjoyed up until that point, BBC News reports.

Researchers figured out the timing by dating the sediments beneath the PIG.

It puts the glacier's current changes in their proper historical context, the scientists tell Nature magazine.

These changes can now be regarded as unprecedented in thousands of years.

Not only is the glacier going backwards, it is also thinning fast - losing more than 2m in elevation every year.

Other field studies and computer models suggest a runaway collapse might even be possible.

"This glacier used to be pinned to a ridge and once it moved away from that ridge, it started to retreat rapidly; and without other pinning points it could continue to retreat rapidly inland, contributing significantly to global sea level," Dr James Smith from the British Antarctic Survey said.

The PIG is a colossal feature that drains a region of the West Antarctic Ice Sheet some two-thirds the size of the UK.

It is a marine-terminating glacier, which means its front flows off the land and pushes out into the ocean along the seafloor until its mass begins to lift up and float. Eventually, the buoyant section breaks up to form icebergs.

Currently, the PIG is dumping about 130 billion tonnes of ice in the ocean every year. Submersible surveys under its floating front - its "ice shelf" - had revealed the contact point with the seabed once draped over a large ridge.

6 Conclusions

In this paper, the research focuses on summary evaluation and the implementation of tools for NLP tasks and especially for automatic summarization. The fundamental concepts and methods related to automatic text summarization have been discussed. Moreover, there was a simple experiment to better understanding working od special python library in order to summarise article from tengrinews with extractive way. It seems that future trend in automatic summarization is not only to focus on the summary information content, however, efforts should also be put into the readability approach of the generated summary and similarity to human language understanding.

References

[1] Jurafsky D, Martin J H 2009 *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics* 2nd edition. Prentice-Hall

[2] Manning C D, Schütze H 1999 *Foundations of Statistical Natural Language Processing* MIT Press

[3] Marcu D 2000 *The Theory and Practice of Discourse Parsing and Summarization* ISBN 0-262-13372-5

[4] Nenkova A, McKeown K 2011 Automatic summarization *Foundations and Trends in Information Retrieval* **5** 103–233

[5] Luhn H P 1958 The automatic creation of literature abstracts *IBM J. Res. Dev.* **2**(2) 159–65 Available: http://dx.doi.org/10.1147/rd.22.0159

[6] Edmundson H P 1969 New methods in automatic extracting *J. ACM* **16**: 264-85 DOI: 10.1145/321510.321519

[7] http://www.nltk.org/

[8] *Journal of Computer Science Science publications* Available: http://thescipub.com/journals/jcs