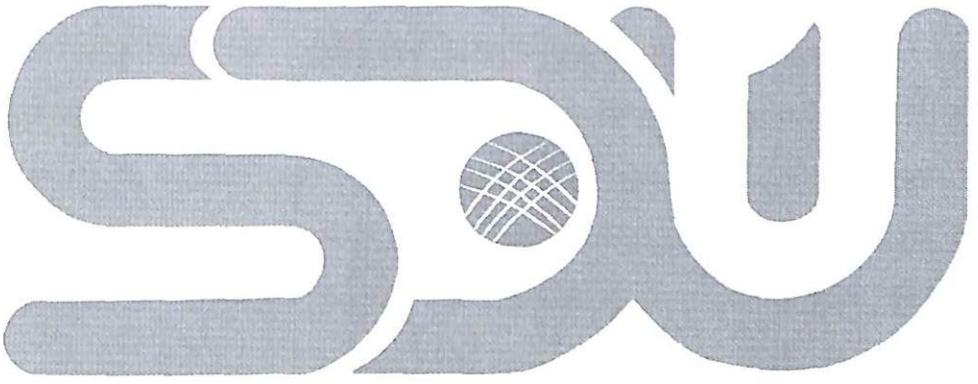
MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN
SULEYMAN DEMIREL UNIVERSITY

UDK 004.8



CHAPAYEV DAUREN

Aspect-oriented definition of emotional tonality of documents in the Kazakh language.

Specialty: "6M070400 – Computing systems and software"

Academic Degree: Master of Computer Sciences

"Admitted to defense":
Dean of Faculty
Assist. Prof., PhD Zhaparov M.
«09» 06 2018 г.

Head of Department — Assist. Prof.,PhD. Alimanova M.U.

Scientific advisor — Dr. tech.sc.professor Amirgaliyev Y. N.

Kaskelen, 2018

АҢДАТПА

Қазіргі уақытта, әрбір адам күнделікті өмірінде, оны мүлдем аңғармаса да, жасанды интеллектті қолданып жүр. Біз оны кез-келген жерде, интернеттегі іздеуден бастап, А нүктесінен В нүктесіне дейінгі жылдам бағытты табуға дейін қолданамыз. Соңғы 10 жыл ішінде жасанды интеллект саласындағы бағдарламалар жаңа тыныс алды. Көптеген зерттеулер XX ғасырда жүргізілгенмен, ол кезде үлкен ақпараттардың жинағы болмағандықтан оларды дұрыс жағдайда қолдана алмады. Бірақ қазір дүние жүзінде әлдеқандай деректер жиналғандықтан нақты тапсырманы орындайтын әлсіз жасанды интеллектті (weak AI) оқытып жасауға болады. Әлсіз ЖИ-тің жақсы мысалы - Apple Siri және Google Assistant. Оның мақсаты - нақты адамдармен әңгімелесуді қолдау (Question & Answering). Сонымен қатар, мәтіннің тоналдылығын анықтау, машиналық аудармасын табу және сөйлеуді тану сияқты тапсырмалар табиғи тілді өңдеудің басқа да міндеттері болып табылады.

Мәтіннің тоналдығын анықтау көптеген пайдаланушыларды түсіну үшін пайдаланылады, бірақ көңіл-күйді талдау үшін қосымшалар шексіз. Әлеуметтік желілер мен VOC (тұтынушының дауысы) тұтынушыларға кері байланыс, сауалнама жауаптары, бәсекелестер мониторингінде және т.б. заттарды қадағалау үшін пайдаланылады. Дегенмен, ол іскерлік аналитикада және мәтінге талдауды қажет ететін жағдайларда қолдану үшін де пайдалы. Өнімнің сапасын арттыру, клиенттерге қызмет көрсетуді жақсарту, дағдарысты басқару сияқты салаларда.

Бұл диссертациялық жұмыста қазақ тіліндегі мәтіннің эмоционалдық тоналдығын анықтау бағдарламасын құру жұмысы түсіндіріледі. Бағдарламаға мәтінді қазақ тілінде береміз және ол тиісінше теріс немесе жағымды түрде жауап беруі керек.

АННОТАЦИЯ

В наше время каждый человек в своей повседневной жизни сталкивается с искусственным интеллектом даже этого не замечая. Мы используем его буквально везде начиная с поиска в интернете и заканчивая проложением быстрого маршрута с точки А до точки Б. За последние 10 лет, разработки в сфере искусственного интеллекта приобрели новое дыхание. Так как многие исследования были сделаны еще в XX веке, но не были использованы в нужном деле так как не хватало большого объема данных. Но сейчас в мире собралось очень много данных, на которых можно обучать и создавать так называемый слабый искусственный интеллект (weak AI) который выполняет одну конкретную задачу. Хорошим примером слабого ИИ является Apple Siri и Google Assistant. Задача которых является поддержка беседы с реальными людьми (Question & Answering). Наряду с этим в обработке естественного языка есть и другие задачи как анализ тональности текста, машинный перевод и распознавание речи.

Анализ тональности текста используется во многом для понимания конечного пользователя, но приложения для анализа настроений бесконечны. Все больше и больше мы видим, что он используется в мониторинге социальных сетей и VOC (voice of the customer) для отслеживания отзывов клиентов, ответов опросов, конкурентов и т. д. Однако он также практичен для использования в бизнес-аналитике и ситуациях, в которых текст нуждается в анализе. В таких как разработка качества продукции, улучшение обслуживания клиентов, антикризисное управление.

В этой диссертации будет разъясняться работа приложения, которая будет определять эмоциональный тон текста на казахском языке. На вход будем давать текст на казахском языке, а программа соответственно должна выдавать ответ в виде негатив или же позитив.

ABSTRACT

In our time, every person in his everyday life is facing artificial intelligence even without noticing it. We use it literally everywhere, from searching the Internet and ending with the location of a fast route from point A to point B. Over the past 10 years, developments in the field of artificial intelligence have acquired a new breath. Since many studies were made as far back as the twentieth century, they were not used in the right case because there was a lack of a large amount of data. But now in the world there is a lot of data on which it is possible to train and create the so-called weak artificial intelligence (weak AI) which performs one specific task. A good example of a weak AI is Apple Siri and Google Assistant. The task of which is to support the conversation with real person (Question & Answering). Along with this, in the processing of natural language there are other tasks such as the sentiment analysis of the text, machine translation and speech recognition.

The sentiment analysis of the text is used in many ways to understand the end user, but applications for analyzing the mood are endless. More and more we see that it is used in monitoring social networks and VOC (voice of the customer) to track customer feedback, survey responses, competitors, etc. However, it is also practical for use in business analytics and situations, in which the text needs analysis. In such areas as product quality development, improvement of customer service, crisis management.

In this thesis, the work of the application, which will determine the emotional tone of the text in the Kazakh language, will be explained. As the input, we will give the text in Kazakh, and the program should accordingly give an answer in the form of negative or positive.

CONTENTS

Introduction

Sentimental analysis, or tonality analysis, is a young but rapidly evolving section of automatic word processing. In the mid-1990s. researchers began to show an interest in expressing the author's subjective attitude in the text [1], including in this notion the opinions, moods, the author's attitude, expressed in some way in the text [2].

With the development of the Internet, sentimental analysis draws the attention of researchers as one of the sections of the analysis of subjectivity whose task is the definition of the meaning of the "tonality" of the text, namely, the classification of the text as reflecting the positive, negative or neutral attitude of the author to objects, phenomena, persons mentioned in the text.

It is important to note that no clear theoretical criteria have yet been formulated on which a section of the text can be assigned to positive, negative or neutral classes, despite the successful attempts of some researchers to theoretically substantiate sentimental analysis [3]. Thus, the evaluation of the value key is established experimentally, with the help of markings by assessors, which then it is used as a "gold standard" for teaching and evaluating the results of the sentimental analysis. The availability of data marked in this way is critical for the development of this area, including because most of the research is focused on the learning methods of classification.

Continuous development of social networks, various kinds of blogs on the Internet, as well as all kinds of resources, where people express their opinion in the form of text messages, growth to interest in tasks related to extraction and analysis opinions of users in automatic mode. To the most urgent problems, from number of text data analysis definition of emotional color text (the views of a person) of a certain polarity, directed toward the object of utterance. Application of methods, collect for the analysis of the tonality of textual information is realized not only as

research tools, but also in commercial activities. Active use of approaches and solutions used in the analysis of the key texts, notes in the following areas: development of recommendations systems; analytics public opinion; extract the user's attitude to the product. This area of research is a topical direction. You can add a number of programs at the international level: "TACL", "ICAICT", "Dialog", "AIST", representing problems in computer linguistics and the analysis of the tonality of the text, confirming this fact.

Automatic analysis of the tonality of texts, i.e. automatic identification of the author's relationship to the objects and situations discussed in the text, is one of the dynamically developing areas of automatic analysis of texts in natural language. Most of the proposed approaches to the analysis of tonality are tested, primarily in English-language texts, and for the English language, a variety of vocabulary resources and tools are created. Recently, a large number of studies in the field of tonality analysis and for other languages have been carried out [11].

For Kazakh language we don't have many researches for now in this area. I found two papers from ENU researchers where they build rule-based sentiment analyzer and get 83% accuracy for simple sentences [27] and used this analyzer for hotel reviews in the Kazakh language [26]. Also, I found paper where researchers from Innopolis University build deep learning model for bilingual sentiment classification for short texts [28]. First, they train classifier in one language then predict in another. From this they get 60% accuracy. As the pair languages they used English-Russian and Russian-Kazakh.

CHAPTER 1. Natural Language Processing

1.1. What is NLP?

Natural language processing is one of the directions of artificial intelligence and mathematical linguistics, which deals with the problems of computer analysis and the synthesis of natural languages.

Let us decipher some of the concepts that make up this definition.

A natural language is a complex system of rules stored in the mind of a person, in accordance with which the speech activity takes place, i.e. generation and understanding of texts.

Artificial intelligence is a field of computer science (informatics), specializing in modeling of intellectual and sensory abilities of a person with the help of computing devices.

Mathematical linguistics is a mathematical discipline, the subject of which is the development and study of concepts that form the basis of a formal apparatus for describing the structure of natural languages (that is, the metalanguage of linguistics).

Analysis as a form and method of activity of artificial intelligence is treated as an understanding of the language. By synthesis, in the sphere of artificial intelligence, the generation of a literate text is understood.

1.4 Main tasks

Understanding, recognition of natural language is the key task, since the recognition of the living language requires a multitude of knowledge of the linguistic system, the language system, their features and regularities.

Define the main, most pressing tasks of natural language processing. It:

Lemmatization - going back to our mushrooms, even an amateur chef knows that you shouldn't just chop off the stems with a knife. Instead, you should carefully

remove the stems, cutting around them with a paring knife or gently twisting them off. Lemmatization tries to take a similarly careful approach to removing inflections.

Lemmatization does not simply chop off inflections, but instead relies on a lexical knowledge base like WordNet to obtain the correct base forms of words.

For example, WordNet lemmatizes *geese* to *goose* and lemmatizes *meanness* and *meaning* to themselves. In these examples, it outperforms than the Porter stemmer.

But lemmatization has limits. For example, Porter stems both *happiness* and *happy* to *happi*, while WordNet lemmatizes the two words to themselves. The WordNet lemmatizer also requires specifying the word's part of speech—otherwise, it assumes the word is a noun. Finally, lemmatization cannot handle unknown words: for example, Porter stems both *iphone* and *iphones* to *iphon*, while WordNet lemmatized both words to themselves.

In general, lemmatization offers better precision than stemming, but at the expense of recall.

Stemming - when we stem a mushroom, we chop off its stem and keep the cap that most people think of as the edible portion. Similarly, when we stem a word, we chop off its inflections and keep what hopefully represents the main essence of the word. Technically, it depends on the type of mushroom, and we're throwing away the mushroom stems while keeping the word stems. Nonetheless, I hope the metaphor is useful.

The best-known and most popular stemming approach for English is the Porter stemming algorithm, also known as the Porter stemmer. It is a collection of rules (or, if you prefer, heuristics) designed to reflect how English handles inflections. For example, the Porter stemmer chops both *apple* and *apples* down to *appl*, and it stems *berry* and *berries* to *berri*.

If we apply a stemmer to queries and indexed documents, we can increase

recall by matching words against their other inflected forms. It is critical that we apply the same stemmer to both queries and documents.

You can find an implementation of the Porter stemmer in any major natural language processing library, such as NLTK and the Stanford NLP suite. You can find stemmers for other languages (or create your own) in Snowball.

Just as using a knife to chop a mushroom stem may leave a bit of the stem or cut into the cap, stemming algorithms sometimes remove too little or too much. For example, Porter stems both *meanness* and *meaning* to *mean*, creating a false equivalence. On the other hand, Porter stems *goose* to *goos* and *geese* to *gees*, when those two words should be equivalent.

In general, stemming algorithms err on the side of being too aggressive, sacrificing precision in order to increase recall.

Information extraction is the task of automatically retrieving (building) structured data from unstructured or semi-structured machine-readable documents. In modern information technologies, the role of such a procedure as the extraction of information is increasing - due to a rapid increase in the number of unstructured (without metadata) information, in particular, on the Internet. This information can be made more structured by converting to a relational form or by adding XML markup. When monitoring news feeds with the help of intelligent agents, you just need methods of extracting information and converting it into a form that will be more convenient to work with later.

Information search is the process of searching unstructured documentary information that satisfies information needs, and the science of this search. Information search is a process of identifying in a set of documents (texts) all those that are devoted to the subject (subject), satisfy a predetermined search condition (request) or contain the necessary (information-relevant) facts, information, data.

Lexical semantics is a part of semantics that deals with values (dividing them into denotation and connotation) of individual lexical elements of words, morphemes and lexemes, thus differing from the semantics of sentences.

The basis of lexical semantics is:

- classification and analysis of words;
- description of differences and common features in lexical semantic structures between different languages;
- the way in which the meaning of individual lexical elements refers, through syntax, to the meaning of the whole sentence.

Semantic units and vocabulary of a language are called lexical units. If one lexical unit is made up of two or more words (a phrase), then one speaks of a phraseological unit.

The central concepts in lexical semantics are lexical relations and how much the meaning of a single word is determined by the meaning of the sentence as a whole, which is called a semantic network in this case. Also pay attention to the relationship values of different words. The central concepts are synonymy, antonymy, hyponymy, as well as significant and auxiliary words. An important role is played also by homonyms and paronyms, but they are connected both with the external form (writing) of words, and with their meaning. Semasiology is a section of linguistics that studies lexical semantics.

Sentiment Analysis or **Opinion mining** - a class of methods of content analysis in computer linguistics, designed to automatically identify in texts emotionally colored vocabulary and emotional assessment of authors (opinions) in relation to objects, which are discussed in the text. Tonality is the emotional attitude of the author of a statement to a certain object (an object of the real world, an event, a process or their properties / attributes) expressed in the text. The emotional component, expressed at the level of the token or communicative fragment, is called

lexical tonality (or lexical sentiment). The tone of the whole text can be defined as a function (in the simplest case, the sum) of the lexical tonalities of its constituent units (sentences) and the rules for their combination.

Machine translation is the process of translating texts (written, and ideally oral) from one natural language to another using a special computer program. The same is called the direction of scientific research related to the construction of such systems.

Question-answering system is an information system capable of taking questions and responding to them in a natural language, in other words, it is a system with a natural-language interface. Good examples of QA is Apple Siri and Google Assistant.

Speech recognition is the process of converting a speech signal into digital information (for example, text data). The inverse problem is the synthesis of speech.

Speech recognition systems are classified:

- by the size of the dictionary (a limited set of words, a large dictionary);
- depending on the speaker (speaker-independent and speaker-independent systems);
- by type of speech (joint or separate speech);
- by appointment (dictation systems, command systems);
- on the algorithm used (neural networks, hidden Markov models, dynamic programming);
- by type of structural unit (phrases, words, phonemes, dyphones, allophones);
- by the principle of selection of structural units (pattern recognition, selection of lexical elements).

For systems of automatic speech recognition, noise immunity is provided, first of all, by using two mechanisms:

- the use of several, parallel, working ways of separating the same elements of the speech signal on the basis of the analysis of the acoustic signal;
- arallel independent use of segmented (phonemic) and holistic perception of words in the flow of speech.

1.5 Corpus

In linguistics, the body (in this sense, the plural - corpora, not the corpus) - is a collection of texts, processed according to certain rules, used as a basis for the study of the language. They are used for statistical analysis and verification of statistical hypotheses, confirmation of linguistic rules in this language.

To solve many linguistic problems, so-called text corpus is used - specially selected and structured collections of texts. The most informative are the marked corps, that is, in which parts of the text are attributed linguistic information - for example, each word is assigned to one or another part of the speech.

Creating a marked body is a very time-consuming process, requiring time and effort for many people. For this reason, the most often marked corps are created by teams of researchers at state institutions, and there are not very many such buildings. Once created, the corps can be used by many researchers to solve various problems. Ways to use the hull can be very diverse, including those that were not thought of by its creators. In order for the corpus to bring maximum benefit to the scientific community, it is necessary that it be accessible not only for viewing through the interface provided by its developers, but also for downloading entirely to the user's computer.

Corpus is used as a data source in statistical linguistics for the detection of unigrams, bigrams and n-grams. These data help analyze the structure of the language and find the most used words, etc.

Basic properties of the corpus

Among the many definitions of the hull, its main properties can be distinguished:

- electronic - in the modern sense of the body should be in electronic form
- representative - should well "represent" an object that models
- marked - the main difference between the body and the collection of texts
- pragmatically oriented - must be created for a specific task

Classification of corpora

Classify the corpora can be on various grounds: the purpose of creating a hull, the type of language data, "literary", genre, dynamism, the type of markup, the volume of texts and so on. By the criterion of parallelism, for example, cases can be divided into monolingual, bilingual and multilingual. Multilingual and bilingual are divided into two types:

- parallel - a lot of texts and their translations into one or several languages
- comparable (pseudo-parallel) - original texts in two or more languages

Corpora marking

The markup consists in attributing to texts and their components special tags: linguistic and external (extra linguistic). The following linguistic types of marking are distinguished: morphological, semantic, syntactic, anaphoric, prosodic, discourse, etc. Further structural levels of analysis are applied to some buildings. In particular, some small cases can be completely syntactically marked. Such cases are usually called deeply annotated or syntactical, and the syntactic structure itself is a tree of dependencies. Manual marking (annotation) of texts is an expensive and time-consuming task. At the moment, open source software provides various software for marking cases. Conditionally they can be divided into stand-alone

(stand-alone) and web-based (web-based). At the same time, the developers' focus in recent years has shifted to web applications. These systems have several advantages:

- possibility of simultaneous marking of one document by several people
- do not require the installation of additional software, except for the browser
- flexible delineation of access rights
- display current progress of the markup process
- possibility of modification of the marked housing

Internet as a corpus

Modern technologies allow creating "web corpus", that is, cases obtained by processing Internet sources:

> Web corpus is a special kind of linguistic corpus that is created by gradually loading texts from the Internet using automated procedures that on the fly determine the language and encoding of individual web pages, delete templates, navigation elements, links and advertising (so-called boilerplate) , they transform into text, filtering, normalization and deduplication of received documents, which can then be processed with traditional instruments of corpus linguistics (tokenization, myth syntactic and syntactic annotation) and introduce it into the search corpus system. The creation of a web corpus is not only much cheaper, but first of all its size can be even an order of magnitude larger than traditional cases [4].

- Vladimir Benko ARANEA - FAMILY OF BILLION WEB-CORPORA

Kazakh language corpora

Kazakh language is one of Turkic languages, so it has inflectional and derivational agglutinative morphology, which causes that these languages have a lot

corps (liturgical texts, modern (XIX-XX century) and earlier periods), historical (including Old Russian, art. Aurora, birchbark letters), syntactic, accentological, multimedia and teaching subcorps.

The volume of the main building on April 7, 2018 was 283 million words, and the total volume of buildings exceeds 600 million words [7].

Texts are provided with meta-markup (by creation date, author, genre, etc.); word forms in texts are provided with automatic morphological and semantic marking; parallel texts are aligned; texts of the poetic corps are also equipped with special metric markings.

1.5% of the texts are provided with morphological and semantic markings with manual removal of homonymy.

Currently, free and free is only search by body. The hull site and search on it are supported by the company "Yandex", whose employees took part also in the development of the software of the hull. Access to the entire building is not possible due to copyright law. To gain access to 1/6 of the marked part of the subcorpus, you must register and accept the license agreement.

Unfortunately, to date, very little work has been done to create a publicly available Kazakh language corpus, but we found some good corpora. The first attempts to build a large-scale corpus of Kazakh language were made by Kazakh Language Corpus [8] and Almaty Corpus of Kazakh [9] projects.

Kazakh Language Corpus (KLC) [8] which is provided by Nazarbayev University research team containing over 135 million words and more than 400 000 documents classified into five major genres (domains): literary, publicistic, official, scientific and informal. Second, Almaty Corpus of Kazakh language (NCKL) [9] which is created by efforts of the department of general linguistics and foreign philology of faculty of philology, literary study and world languages of Al-Farabi Kazakh National University under the leadership of the chief of the department G.B.

Madiyeva with the participation of the staff of the faculty of philology of Higher School of Economics National Research University (Moscow). At the moment the size of the corpus is more than 40 million word tokens. Finally, The Kazakh Web Corpus [10] created by Sketch Engine. This corpus is collected from the web using SpiderLing crawler. The corpus was prepared according to standards described in the document A Corpus Factory for Many Languages (Kilgarriff et al. at LREC 2010). Data were downloaded in January 2012 with the total size 139 million words. Texts were cleaned and deduplicated.

All these corpora are probably good, but there is one minus, we can not use them in our studies since they are private. Because of the fact that we do not have a widely available open source corpus, many studies are being ended at this stage.

Chapter 2. Sentiment Analysis

2.1 What is Sentiment Analysis?

Analysis of the tonality of the text (Sentiment analysis, English Opinion mining) - a class of methods of content analysis in computer linguistics, designed to automatically identify in texts emotionally colored vocabulary and emotional assessment of authors (opinions) in relation to objects, which are discussed in the text.

Tonality is the emotional attitude of the author of a statement to a certain object (an object of the real world, an event, a process or their properties / attributes) expressed in the text. The emotional component, expressed at the level of the token or communicative fragment, is called lexical tonality (or lexical sentiment). The tone of the whole text can be defined as a function (in the simplest case, the sum) of the lexical tonalities of its constituent units (sentences) and the rules for their combination.

Sentiment Analysis explores the problem of studying texts, such as messages and reviews, uploaded by users on microblogging platforms, forums and electronic enterprises, regarding the opinions they have about the product, service, event, person or idea. The most common use of mood analysis is the classification of text for a class. Depending on the data set and the reason, the mood classification can be binary (positive or negative) or multiclass (3 or more classes). In addition, among researchers and stakeholders, you can find similar or completely different opinions about the relationship between the detection of emotions and the analysis of feelings depending on their perspective. However, regardless of the result or approach, they all use the same methods.

2.2 Types of classification

In modern systems of automatic determination of emotional evaluation of the text one-dimensional emotional space is most often used: positive or negative (good or bad). However, successful cases of using multidimensional spaces are known.

The main task in the analysis of tonality is the classification of the polarity of this document, that is, determining whether the expressed opinion in the document or proposal is positive, negative or neutral. More developed, "out of polarity," the classification of tonality is expressed, for example, by emotional states such as "evil," "sad," and "happy."

Classification by binary scale

The polarity of the document can be determined by binary in the scale. In this case, two classes of assessments are used to determine the polarity of the document: positive or negative. One of the drawbacks of this approach is that the emotional component of a document can not always be uniquely determined, i.e. the document may contain signs of both positive and negative evaluation [2]. Early works in this area include the works of Turney [12] and Pang [11], who apply various methods of recognizing the polarity of product reviews and reviews of films, respectively. This is an example of work at the document level.

Classification by multiband scale

It is possible to classify the polarity of a document on a multiband scale, as was done by Pang [13] and Snyder [14] (among others). They expanded the main task of classifying film calls from the assessment of "positive or negative" in the direction of forecasting a rating on a 3 or 4-point scale. At the same time, Snyder conducted an in-depth analysis of restaurants reviews, predicting the ratings of their various properties, such as food and atmosphere (on a 5-point scale) [14].

Scaling systems

Another method of determining tonality is the use of scaling systems, whereby words usually associated with negative, neutral or positive keys are placed in

accordance with numbers on a scale of -10 to 10 (from the most negative to the most positive). First, a fragment of unstructured text is examined using tools and algorithms for processing natural language, and then the objects and terms extracted from this text are analyzed to understand the meaning of these words. [15].

Subjectivity / objectivity

Another research direction is the identification of subjectivity / objectivity [16]. This task is usually defined as the assignment of this text in one of two classes: subjective or objective. This problem can sometimes be more complicated than the classification of polarity: the subjectivity of words and phrases may depend on their context, and an objective document may contain subjective suggestions (for example, a news article citing people's opinions). Moreover, as Su said [16], the results are more dependent on the definition of subjectivity used in the annotation of texts. However, Pang [17] showed that removing objective sentences from the document before polarity classification helped to improve the accuracy of the results.

A more detailed analysis model is called a function / aspect analysis. This model refers to the definition of opinions or moods expressed by various functions or aspects of entities, for example, a cell phone, a digital camera, or a bank. A property / aspect is an attribute or component of an entity being examined for tonality, for example, a cell phone screen or the quality of a camera's shooting. This problem requires the solution of a number of problems, for example, identification of actual entities, extracting their functions / aspects and determining whether the opinion expressed for each function / aspect is positive, negative or neutral. More detailed discussions on this subject can be found in the NLP handbook, in the chapter "Analysis of tonality and subjectivity".

Approaches to the classification of the key

Computers can perform automatic analysis of digital texts using machine learning elements, such as hidden semantic analysis, support vector machine,

"bag-of-words" and semantic orientation in this area. More complex methods are trying to determine the possessor of the mood (that is, the person) and the goal (that is, the essence, in relation to which the feelings are expressed). To determine the opinion in context, use the grammatical relationship between the words.

Relations of grammatical connectivity are obtained on the basis of a deep structural analysis of the text. The tone analysis can be divided into two separate categories:

- manual (or analysis of the key by experts);
- automated tone analysis.

The most noticeable differences between them lie in the effectiveness of the system and the accuracy of the analysis. Computer programs for computer-aided tone analysis use machine learning algorithms, statistics tools, and natural language processing, which allows processing large arrays of text, including web pages, online news, discussion groups on the Internet, online reviews, web blogs and social media.

2.3. Use Cases

The analysis of tonality finds its practical application in different areas:

- sociology - we collect data from social. networks (for example, about religious views)
- political science - we collect data from blogs about political views of the population
- marketing - analyze Twitter to find out which notebook model is most in demand
- medicine and psychology - determine the depression in users of social. networks

Using sentiment analysis you control your business by knowing user reactions on some changes on your product. You able to know mood of users on some of

period time and give alternative product or new feature. Sentiment analysis very powerful tool and can be used in various use cases.

2.4 Methods

For sentiment analysis we have 4 methods, but here we will consider only two of them it lexicon-based approach and machine learning approach (Figure 1).

Lexicon-Based - use the so-called affective lexicons for text analysis. In its simplest form, a tonal dictionary is a list of words with a tonality value for each word (Table 1).

Words	Value(1-9)
beautiful	7.10
the best	9.18
monotonous	3.12
annoyed	2.31
depressed	1.78

Table 1: Words with tonality value

Also, method is based on the search for emotional vocabulary (lexical tonality) in the text according to pre-composed voice dictionaries and rules with the use of linguistic analysis. By the totality of the emotive language found, the text can be evaluated on a scale containing the number of negative and positive vocabulary. This method can use both lists of rules that are inserted into regular expressions, as well as special rules for combining tonal vocabulary within a sentence.

The main problem of methods based on dictionaries and rules is the complexity of the dictionary compilation process. In order to obtain a method that classifies the document with high accuracy, the terms of the dictionary must have a weight adequate to the subject area of the document. For example, the word "huge"

in relation to the amount of hard disk memory is a positive characteristic, but negative with respect to the size of the mobile phone. Therefore, this method requires considerable effort, since for the good operation of the system it is necessary to compile a large number of rules. There are a number of approaches that make it possible to automate the compilation of dictionaries for a particular subject area (for example, the theme of restaurants or the theme of mobile phones).

There are a number of thesauruses specially marked taking into account the emotional component. Such dictionaries, described below, are necessary for computer programs when analyzing the tonality of a text.

WordNet-Affect

An example for the development of WordNet-Affect was the multilingual extension of WordNet, called WordNet Domain. In the expansion of WordNet Domain, each synth is attributed to at least one domain label (English "domain label"), for example: sports, politics, medicine. In total, about two hundred subject litters were included in the hierarchically organized structure.

WordNet-Affect is a semantic thesaurus in which concepts related to emotions ("emotional concepts", English "affective concepts") are represented by words that have an emotional component ("emotional words", English "affective words"). WordNet-Affect consists of a subset of synths of WordNet, where each synth corresponding to the "emotional concept" can be represented using "emotional words".

Thus, WordNet-Affect was created on the basis of WordNet for English (there are also versions of WordNet-Affect for other languages) by selecting and assigning synonym sets (synsettes) to different emotional concepts. In particular, the synths of verbs, nouns, adjectives, adverbs, which are a description of emotions, have been manually marked with the help of special emotional labels (A-labels). These emotional marks characterize different states expressing moods, emotional responses

or situations that cause emotions. Examples of such emotional labels are given in the following table.

Also, in WordNet-Affect, additional emotional tags are used to separate the syncets according to their emotional valence. For this, four additional emotional tags are defined: positive, negative, ambiguous and neutral. The first corresponds to positive emotions, which are defined as emotional states characterized by the presence of positive hedonic signals (or pleasure). It includes such synths as joy # 1 or hobby # 1. Similarly, a negative label identifies negative emotions characterized by negative hedonic signals (or pain), for example, anger # 1 or sadness # 1. Shinsetts, representing emotional states whose valency depends on the semantic context (for example, surprise # 1) are marked as ambiguous. Finally, the synsettes that define psychological states and are always considered ambiguous, but not characterized by valence, are neutral (Table 2).

Emotional Mark	Example
cognitive state	noun confusion, adj. dazed
mood	noun animosity, adj. amiable
trait	noun aggressiveness, adj. competitive
physical state	noun illness, adj. all in
behaviour	noun offense, adj. inhibited

Table 2: Emotional Marks of Words

The syncets marked with emotional marks are additionally re-arranged in six emotional categories: joy, fear, anger, sadness, disgust, surprise. Thus, the physical structure of WordNet-Affect consists of six files: anger.txt, disgust.txt, fear.txt, joy.txt, sadness.txt, surprise.txt, where each file is a description of a category. At the moment WordNet-Affect contains 2874 syncets and 4787 words.

Researchers at the Technical University of Moldova translated WordNet-Affect syncets from English into Russian and Romanian, and aligned them: English - Romanian - Russian. The resource is available online for research purposes.

SentiWordNet

SentiWordNet is a lexical semantic thesaurus, the first version of which was developed in 2006. At the moment, the latest version of SentiWordNet is SentiWordNet 3.0, the use of which gives a more than 20% increase in accuracy compared to the first version.

This system is the result of the process of automatically annotating each WordNet sync (a set of synonyms) in accordance with its degree of positivity, negativity and objectivity. Thus, for each synonymic series from WordNet three numerical evaluations are assigned, where each of these evaluations respectively determines the objective, positive or negative component of the synset. Each of these estimates takes values in the range from 0 to 1, and in the sum they give 1 (one), that is, each of their estimates can have a nonzero value [26]. Terms that can have different meanings may have different values of estimates.

The learning process of SentiWordNet consisted of two steps:

- at the first stage of the development of the system, methods of machine learning with a weak (semi-supervised learning) technique for primary filling were used. At first, a small number of synsettes were selected, and numerical estimates were manually assigned to them. Then, on the basis of this set, several classifiers were trained whose task is to determine the degree of positivity, negativity, and objectivity of the synset. After that, through the obtained classifier models, numerical estimates for each WordNet synset were determined.
- to the data obtained in the first step, a random-walk step model was used, which resulted in the final evaluations of the objective, positive or negative

components of each syncet. More about this stage you can read in the next work.

SentiWordNet is distributed under the license of CC BY-SA 3.0. This license allows SentiWordNet to be used freely for commercial and scientific purposes, provided that the names of the creators are indicated. Anyone can download SentiWordNet files from the official site for free. You can also download a small Java class that demonstrates working with SentiWordNet.

SenticNet

SenticNet is another semantic thesaurus for working with sets of emotional concepts. SenticNet is a project launched in the Media Laboratory of the Massachusetts Institute of Technology in 2010. Since then, the SenticNet project has been further developed and applied to the design of intelligent applications designed to analyze the emotional component of the text and span the range of tasks from data mining to the organization of human-computer interaction. The main purpose of SenticNet is to simplify the procedure of computer recognition of conceptual and emotional information transmitted through natural language. If you compare other lexical thesauri, such as SentiWordNet and WordNet-Affect with SenticNet, then their main difference is that SentiWordNet and WordNet-Affect provide the linking of words and emotional concepts to the syntactic level, not allowing you to identify the semantic component, for example, "reaching the goal "" A bad feeling, "" celebrate a special occasion, "" lose self-control, "or" be in the seventh heaven of happiness, "while SenticNet connects concepts at a semantic level.

The latest version is SenticNet 2. Unlike the version of SenticNet 1, which simply assigns a key value of approximately 5700 terms from the OpenMind case, SenticNet 2 provides the linking of semantics and "sentics" (that is, cognitive and "emotional" information) to more than 14,000 concepts and allows for more a deep and multifaceted analysis of the text in natural language in comparison with

SenticNet 1. SenticNet 2 is built using "sentic-calculations", a paradigm that uses AI methods and semantic web to improve recognition, interpretation and processing of opinions in natural language.

"Sentic-computing" is an interdisciplinary approach to analyzing the tonality of the text at the crossroads between "affect computing" and "common sense computing". The term "common sense computing" refers to a series of initiatives aimed at ensuring that computers have knowledge of everything in the world as humans understand them and that computers are able to build logical conclusions based on this knowledge. Such an interdisciplinary approach involves the use of information and social science tools to improve the recognition, interpretation and processing of opinions and feelings. In particular, Sentic calculations involve the use of methods of artificial intelligence and semantic web - to represent knowledge and their output; mathematics - for solving such problems as processing graphs and lowering the dimension; linguistics - for discursive analysis and pragmatics; psychology - for cognitive and emotional modeling; sociology - for understanding the dynamics of social networks and social influence; and finally ethics - to understand the nature of the mind and create emotional machines. "Sentic-calculations" allow analyzing documents not only at the level of whole pages and texts, but also at the level of sentences, which allows us to evaluate texts at a higher level of detail.

In order to present SenticNet data in machine-readable form, which is easy to process by computer programs, the data is encoded into RDF triplets using XML syntax. An example of an XML file for the "love" concept can be found on the project website at the following link. For example, if the concept of "birthday" is encountered during the application process, then SenticNet will refer it to the concept of a high level of "event" and link it to a set of semantically related concepts, for

example, "sweet", "friendly surprise" or "clown" (which can be used as a source of additional / contextual information to improve search results).

SenticNet also associates each concept with a "sentic-vector" with numerical values such as Pleasantness, Attention, Sensitivity and Aptitude, as well as the tonality value (for tasks such as the analysis of the tonality of the text), the main and additional mood, as well as a set of emotionally related concepts, for example, "holiday" or "special case" (for tasks such as determining the key of the text). Anyone can freely download SenticNet 2 from the official site.

Supervised machine learning

In our time, the most frequently used methods in research are methods based on machine learning with the teacher. The essence of such methods is that in the first stage the machine classifier is taught (for example, Bayesian) on pre-marked texts, and then use the obtained model in the analysis of new documents. We describe a brief algorithm:

- first, a collection of documents is collected, on the basis of which the machine classifier is trained;
- each document is decomposed as a vector of signs (aspects), on which it will be explored;
- indicates the correct type of key for each document;
- a choice is made of the classification algorithm and a method for classifier training;
- the received model is used to determine the tonality of the documents of the new collection.

Unsupervised machine learning

At the heart of this approach is the idea that terms that are more common in this text and at the same time are present in a small number of texts throughout the

collection have the greatest weight in the text. Selecting these terms, and then determining their tonality, we can conclude that the tonality of the entire text.

Criteria	Lexicon-based	Machine learning
Domain	Independent	Dependent
Classification approach	Unsupervised	Supervised
Require prior training	No	Yes
Adaptive learning	No	Yes
Time of result generation	Fast	Slow
Maintenance	Need maintenance of corpus	Do not require maintenance
Accuracy	low	high then lexicon

Table 3: Lexicon-based vs. Machine learning

A method based on graph models

This method is based on the assumption that not all words in the text body of the document are equivalent. Some words have more weight and more influence on the tonality of the text. Using this method, the analysis of the tonality is divided into several stages: .

- the construction of a graph on the basis of the text under study;
- ranking of its vertices;
- classification of found words;
- calculation of the result.

More details on points 1 and 2 can be found in the work "Extracting terms from Russian texts using graph models" by D. Ustalov [18].

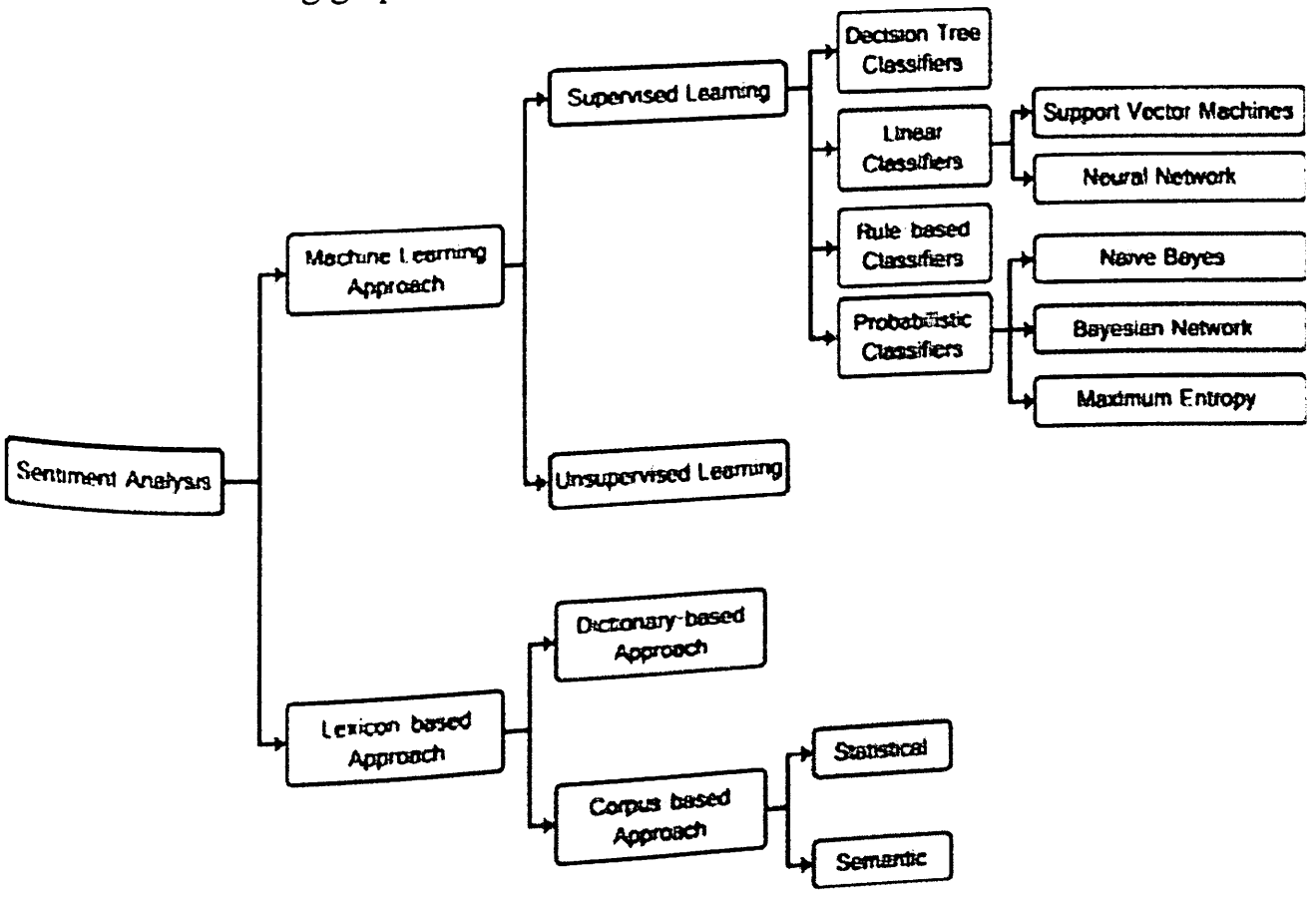


Figure 1: Methods

Hybrid - this approach combines all or several of the principles discussed above and consists in applying the classifiers on their basis in a certain sequence. It is the combination of machine learning and lexicon-based approach.

Methods	Features	Pros	Cons
Machine Learning (Bayesian Networks, Naive Bayes,	Term presence and frequency Part of speech information	the ability to adapt and create trained models for specific purposes and	the low applicability to new data beca use it is necessary the

Classification, Maximum Entropy, Neural Networks, SVM)	Negations Opinion words and phrases	contexts	availability of labeled data that could be costly or even prohibitive
Lexicon based (Dictionary based approach, Novel Machine Learning Approach, Corpus based approach, Ensemble Approaches)	Manual construction, Corpus - based Dictionary - based	wider term coverage	finite number of words in the lexicons and the assignation of a fixed sentiment orientation and score to words

Table 4: Pros and Cons of Methods

As you see machine learning approach more powerful than lexicon-based. Big cons of lexicon-based approach that we need to construct our dictionary manually. For machine learning we need just training set bigger than 10k, but here also we need to construct training set manually (Table 4).

Chapter 3. Word embedding

3.1 Bag of Words

The simplest model of the text "bag-of-words" is a summative unity (not a system) of text words composing.

The main object of the bag of word model is a word equipped with a single attribute, the frequency of occurrence of the word. In the "bag of word" text model, only the number of occurrences of specific words in the source text is taken into account, while ignoring:

- the order of words in the document
- morphological forms of the representation of words

The bag of word model was proposed in 1975 by J. Salton, and is currently one of the most common in a wide variety of areas of linguistic research and services, usually as a basis for more complex, primarily "vector test models".

Frequency model of the text

For each word in the "bag of word" model set, some "weight" may be indicated. Thus, the text model is a set of word-weight pairs. In this case, weights can be assigned to words or the basics of words. Methods for determining the weight of words:

- The binary method (the common notation is BI, from binary). Only the presence or absence of certain terms in the document is determined. It is used for logical information retrieval and automatic classification of texts by methods of neural network classifiers ART and SOM.
- Number of occurrences (words in the document). Assumes disproportionate evaluation for texts of different lengths - more weight will receive more voluminous texts, since there are more words in them;
- Frequency of occurrence of a word in the document (TF - term frequency). The frequency is calculated as the ratio of the word's occurrence to the total number

of words in the text. With relative simplicity, this characteristic provides an acceptable result for methods of information retrieval and classification (assuming disproportionate evaluation for texts of different lengths - long documents are underestimated, since they contain more words and mean frequency of words in the text below).

- Logarithm of word occurrence frequency (LOGTF). The weight of the incoming text is defined as 1 + log (TF), where TF is the frequency of the term. Using a logarithmic scale allows the model to be more resistant to the re-evaluation of texts of different volumes.
- The reverse frequency of documents (IDF - inverse document frequency). The parameter is the inversion of the frequency with which the term appears in the documents.

A bag of words is used in "machine learning based on texts" as one of the main objects of study.

It starts from the stove, that is, from the statement of the task. Where does the word embedding itself come from? By itself, embedding is the mapping of an arbitrary entity (for example, a node in a graph or a piece of a picture) to a certain vector (Figure 2).

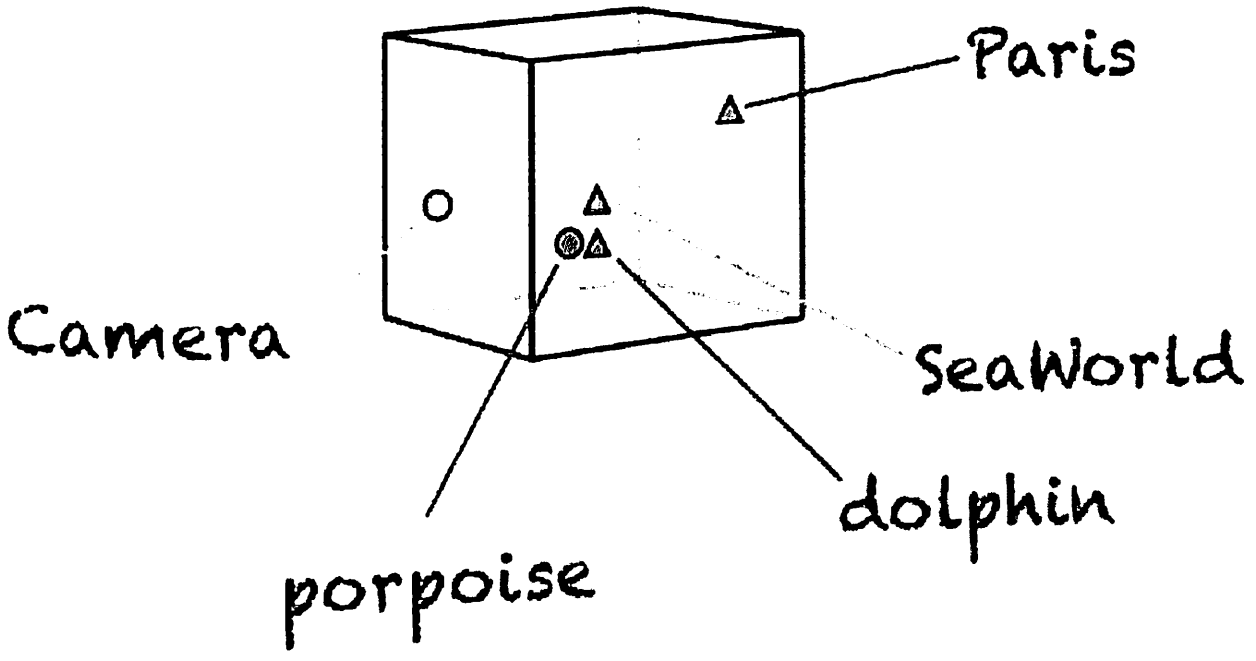


Figure 2: Word Embedding

Here we have words and there is a computer that must somehow work with these words. The question is, how will the computer work with words? After all, the computer does not know how to read, and in general is structured differently than man. The very first idea that comes to mind is simply to encode words in numbers in the order in the dictionary. The idea is very productive in its simplicity - the natural series is endless and you can renumber all words without fear of problems. (For a second we forget about type restrictions, especially since in a 64-bit word you can push numbers from 0 to 2^64 - 1, which is much larger than the number of all the words of all known languages.) But this idea has a significant drawback: words in the dictionary follow in alphabetical order, and when adding a word, you need to renumber the majority of words a new. But even this is not so important, and more importantly, the alphabetic writing of a word is not related to its meaning (this hypothesis was expressed already by the late 19th century by the famous linguist Ferdinand de Saussure). In fact, the words "rooster", "chicken" and "chicken" have

very little in common with each other and stand in the dictionary far apart, although they clearly designate the male, female and calf of one kind of bird. That is, we can distinguish two types of proximity of words: lexical and semantic. As we see in the example of a chicken, these intimations do not necessarily coincide. For clarity, it is possible to give a reverse example of lexically close but semantically distant words - "ash" and "gold." (If you have never thought about it, the name Cinderella comes from the first.)

In order to be able to imagine the semantic closeness, it was suggested to use embedding, that is, to match the word with a certain vector that reflects its meaning in the "space of meanings".

What's the easiest way to get a vector out of a word? It seems that it will be natural to take the vector of the length of our dictionary and put only one unit in the position corresponding to the number of the word in the dictionary. This approach is called one-hot encoding (ONE). ONE still does not have the semantic proximity properties (Figure 3):

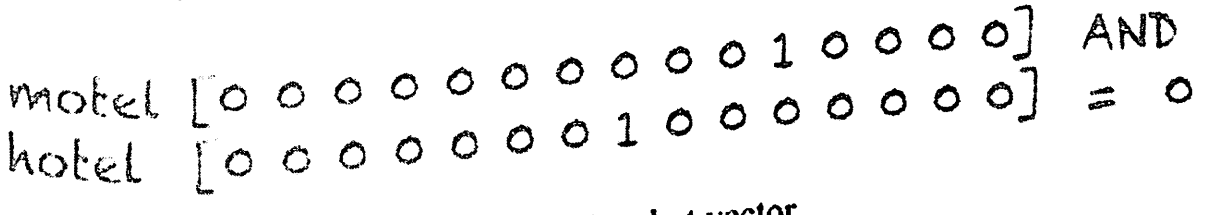


Figure 3: One hot vector

So we need to find another way of converting words to vectors, but ONE is still useful.

Let's go back a bit - the meaning of one word to us may or may not be so important, because speech (both oral and written) consists of sets of words, which we call texts. So if we want to somehow present the texts, then we'll take the ONE vector of each word in the text and put it together. Those. at the output, we will simply calculate the number of different words in the text in one vector. This approach is

called a "bag of words" (BoW), because we lose all information about the relative position of words inside the text (Figure 4).

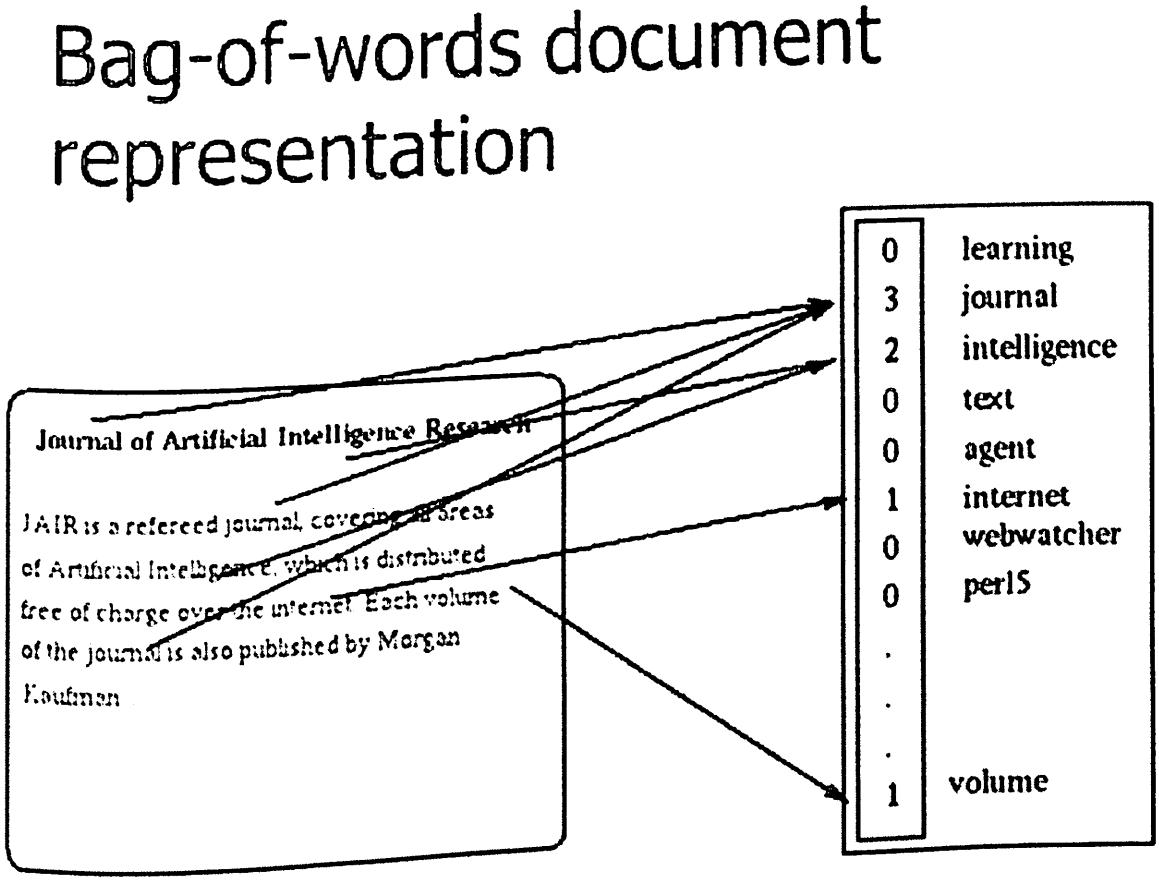


Figure 4: Bag-of-Words Representation

3.2 Word2Vec

The approaches described above have been (and still are) good for times (or areas) where the number of texts is small and the dictionary is limited, although, as we have seen, there also have their difficulties there. But with the advent of the Internet in our life, everything has become both more complicated and simpler: a great number of texts appeared in the access, and these texts with a changing and expanding dictionary. With this it was necessary to do something, and previously

known models could not cope with such a volume of texts. The number of words in English is very roughly one million - the matrix of joint occurrences of only a pair of words is 10^6 x 10^6. Such a matrix even now does not go very far into the memory of computers, but, let's say, 10 years ago it was possible not to dream of such a thing. Of course, many methods were invented, simplifying or parallelizing the processing of such matrices, but all these were palliative methods.

And then, as often happens, the way out was suggested on the principle "he who prevents us, he will help us!" Namely, in 2013, then very few well-known Czech graduate student Tomas Mikolov proposed his approach to word embedding, which he called word2vec. His approach is based on another important hypothesis, which in science is called the locality hypothesis - "words that occur in identical environments have similar meanings." Proximity in this case is understood very broadly, like the fact that only a combination of words can stand side by side. For example, for us the phrase "alarm clock" is familiar. And we can not say "clock orange" - these words do not combine. Based on this hypothesis, Tomas Mikolov proposed a new approach that did not suffer from large amounts of information, but rather won.

The first approach is a fairly common method and is fairly simple in imple - mentations, but it is not excluded from shortcomings. When comparing two vectors, the exact match of words is used, and we lose important information. One of these "missed" information is the semantics of the word. For example, we can easily replace the word "black" with the word "dark", since their meaning is very similar. Such words can be called - semantic similar words. A group of such words includes synonyms, hyponyms, hyperonisms, etc.

In an alternative approach, we will try to replace each word in the list with the number of its semantic group. In the end, we get something like a "bag of words," but with a deeper meaning. For this, Word2Vec technology from Google is used (Table 5).

Words	Measures
France	0,6056
London	0,4935
England	0,4893
Amsterdam	0,4632
Barcelona	0,4574
Madrid	0,4487
Moscow	0,4288
Singapore	0,4166
Washington	0,3853
Astana	0,3829

Table 5: Semantic group of word "Paris"

Word2Vec is a software tool for analyzing the semantics of natural languages, which is a technology that is based on distributive semantics and vector representation of words. This tool was developed by a group of Google researchers in 2013. The work on the project was headed by Tomas Mikolov (now he works on Facebook). Word2Vec is sharpened for statistical processing of large arrays of textual information. W2V collects statistics on the joint occurrence of words in phrases, after which the methods of neural networks solve the problem of reducing the dimension and gives out compact vector representations of words, which maximally reflect the relationship of these words in the processed texts.

Word2Vec is a tool (a set of algorithms) for calculating vector representations of words, implements two main architectures - Continuous Bag of Words (CBOW)

and Skip-gram. The input is a body of text, and the output is a set of word vectors (Figure 5).

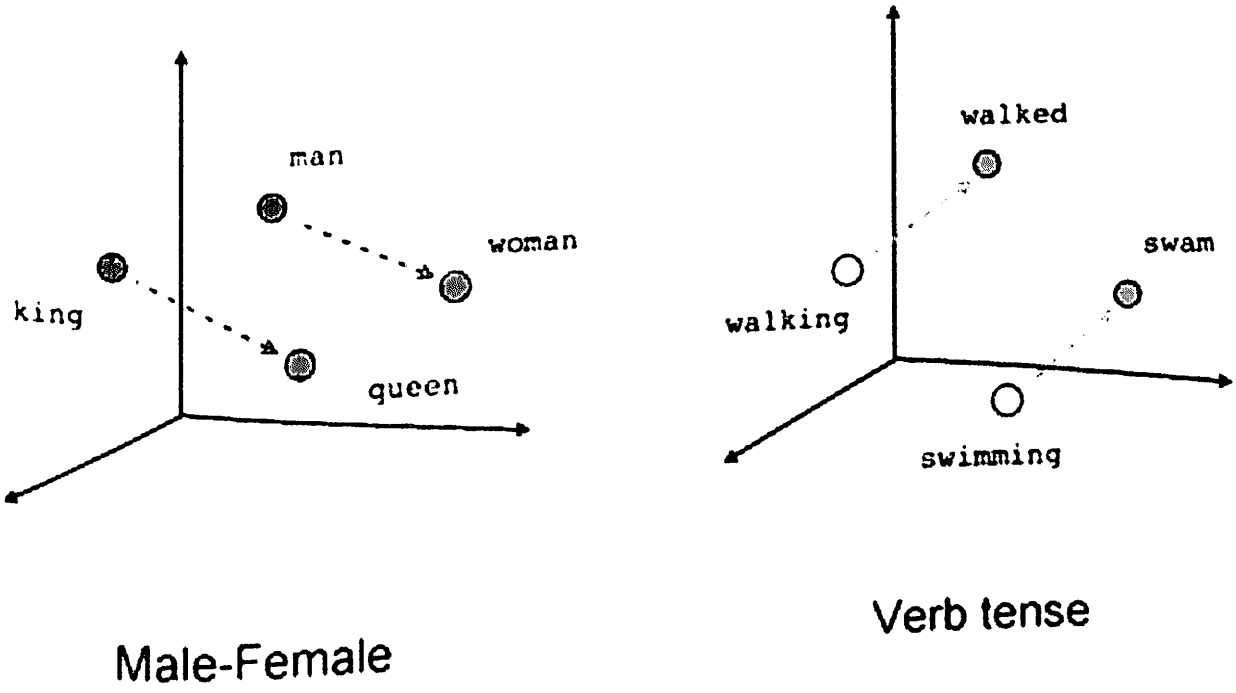


Figure 5: Words in one semantic group

Principle of operation:

Finding connections between the contexts of words according to the assumption that words that are in similar contexts tend to mean similar things; be semantically close. More formally, the task is to maximize the cosine proximity between the word vectors (scalar product of vectors) that appear side by side, and minimize the cosine proximity between word vectors that do not appear next to each other. Next to each other in this case means in close contexts.

For example, the words "analysis" and "research" are often found in similar contexts, such as "Scientists have analyzed the algorithms" or "Scientists have studied the algorithms." Word2vec analyzes these contexts and concludes that the words "analysis" and "research" are similar in meaning. Since word2vec makes

similar conclusions on the basis of a large amount of text, the conclusions turn out to be quite adequate. For example, if train word2vec on a collection of 10,000 small scientific texts (which, in general, is not enough), then for the word "Spain" on the finished model, the following list of words closest in meaning - Italy, Australia, Netherlands, Portugal, France. That is, everything is quite clear, except, perhaps, Australia.

On the basis of the above, we can conclude that for the learning of the word2vec model of good quality, a large, large case is needed. Wikipedia, for example, is quite suitable.

Word2Vec can be perfectly applied for various tasks of natural language processing, such as:

- clustering words according to the principle of their semantic intimacy
- identifying the semantic proximity of words (for example, what is semantically closer to the word cat - to food or to space pirates)
- for sentiment analysis

3.2.1 How does it all work?

- The body is read and the occurrence of each word in the case is calculated (that is, the number of times the word was encountered in the case - and so for each word)
- The array of words is sorted by frequency (words are stored in a hash table), and rare words are deleted (they are also called hapaks)
- A Huffman tree is being constructed. The Huffman Binary Tree is often used to encode a dictionary - this greatly reduces the computational and temporal complexity of the algorithm.
- From the body is read the so-called. sub-sentence and sub-sampling of the most frequent words. A sub-proposal is a certain basic element of the hull,

usually just a sentence, but it can be a paragraph, for example, or even an entire article. Sub-sampling is the process of removing the most frequent words from the analysis, which speeds up the learning process of the algorithm and contributes to a significant increase in the quality of the resulting model.

- Under the sub-proposal, we go through the window (the window size is specified by the algorithm as a parameter). In this case, the window is the maximum distance between the current and predicted word in the sentence. That is, if the window is three, then the analysis (that is, the application of algorithms based on one of the CBOW or Skip-gram architectures) for the sentence "I sell my house on krisha.kz" will pass inside the block in three words - for "I sell my" "sell my house on krisha.kz," etc. The default window is five, the recommended value is ten.
- A Feedforward Neural Network with a hierarchical softmax activation function and / or negative sampling is used. Hierarchical softmax behaves better when working with not very frequency words, but it works slower, negative sampling works better with frequency words and loves word vectors of small dimensions (say 50-100), but faster.

3.2.2 Types

In word2vec, there are two basic learning algorithms: CBOW (Continuous Bag of Words) and Skip-gram. CBOW is a "continuous bag of words" a model architecture that predicts the current word based on the context surrounding it. The Skip-gram type of architecture works differently: it uses the current word to anticipate the words surrounding it. User word2vec has the ability to switch and choose between algorithms. The order of context words does not affect the result in any of these algorithms.

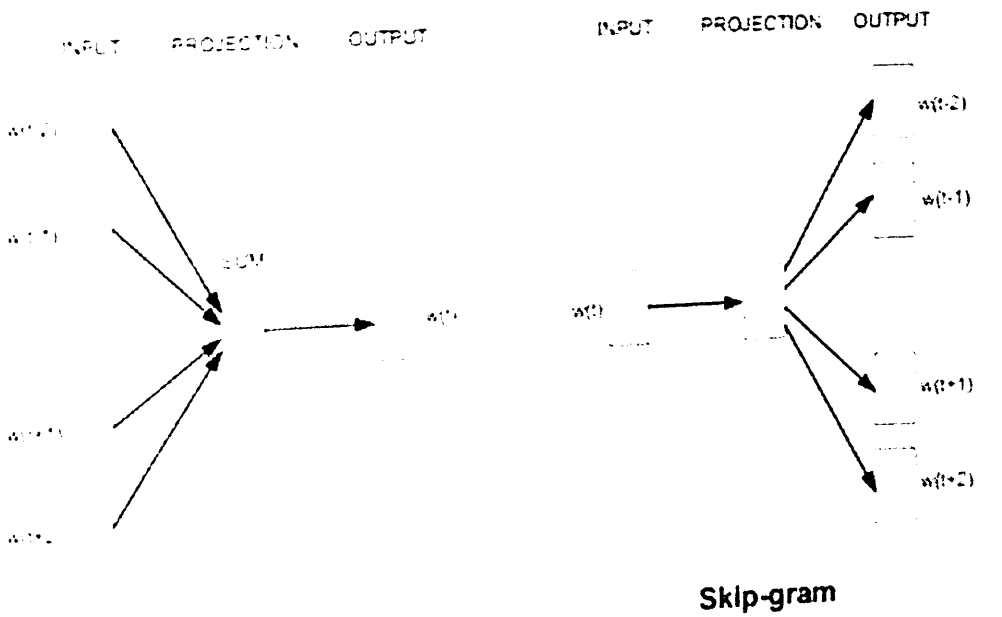


Figure 6: CBOW and Skip-gram algorithms

In general, CBOW and Skip-gram are neural network architectures that describe how the neural network "learns" on the data and "remembers" the word representations. Principles of both architectures are different. The principle of CBOW operation is the prediction of a word in this context, and skip-gram on the contrary - the context for this word is predicted (Figure 6).

Continuous Bag of Words (CBOW) - the usual model of a bag of words, taking into account the four closest neighbors of the term (the two previous and two subsequent words) without taking into account the order of the sequence.

k-skip-n-gram is a sequence of length n, where the elements are at a distance of no more than k from each other. For example, given the sentence "I sell my house on krisha.kz". For this sentence, 1-skip-2-gram will look like a set of bigrams ("I sell", "sell my", "my house", "house on", "on krisha.kz") plus the following sequences: "I my, sell house, my on, house krisha.kz" - that is, skip one (1-skip) word, and from the left to the right and left of the missed we make a bigram (2-gram).

What are the parameters that influence and which parameters are optimal?

- CBOW works faster, but Skip-gram works better, especially for relatively rare words
- Hierarchical softmax is well suited for creating a better model of relatively rare words, negative sampling is better for modeling more frequency words.
- The use of sub-sampling improves performance. Recommended sub-sampling parameter from 1e-3 to 1e-5
- The size of the vector is larger, the better (though not always, it depends on the body)
- The size of the window - for Skip-gram the optimal size is about 10, for CBOW - around 5

How all the same it is better to train word2vec?

Comrade Mikolov himself (the inventor of word2vec, if anything) believes that the CBOW model is better suited for large text cores (one hundred million, one billion words or more), because it works faster and slightly better with more frequency words, and Skip-gram is better for small shells of texts (less than one hundred million words), since it better takes into account rare words and works slower.

3.3 Fasttext library

The **FastText** model was first introduced by Facebook in 2016 as an extension and supposedly improvement of the vanilla Word2Vec model. Based on the original paper titled *'Enriching Word Vectors with Subword Information'* by Mikolov et al. which is an excellent read to gain an in-depth understanding of how this model works. Overall, FastText is a framework for learning word representations and also performing robust, fast and accurate text classification. The framework is open-sourced by Facebook on **GitHub** and claims to have the following.

- Recent state-of-the-art English word vectors.
- Word vectors for 157 languages trained on Wikipedia and Crawl.
- Models for language identification and various supervised tasks.

For Kazakh language they collected 34 million tokens from Wikipedia and 196 million tokens from Common Crawl after deduplication and tokenization. Binary file size is 7.2GB and vector file size is 4.5GB. Used CBOW as an algorithm for learning. Also, they have vector files which is trained only on Wikipedia dumps using Skip-gram model. Here we displayed our vector using t-SNE algorithm (Figure 7).

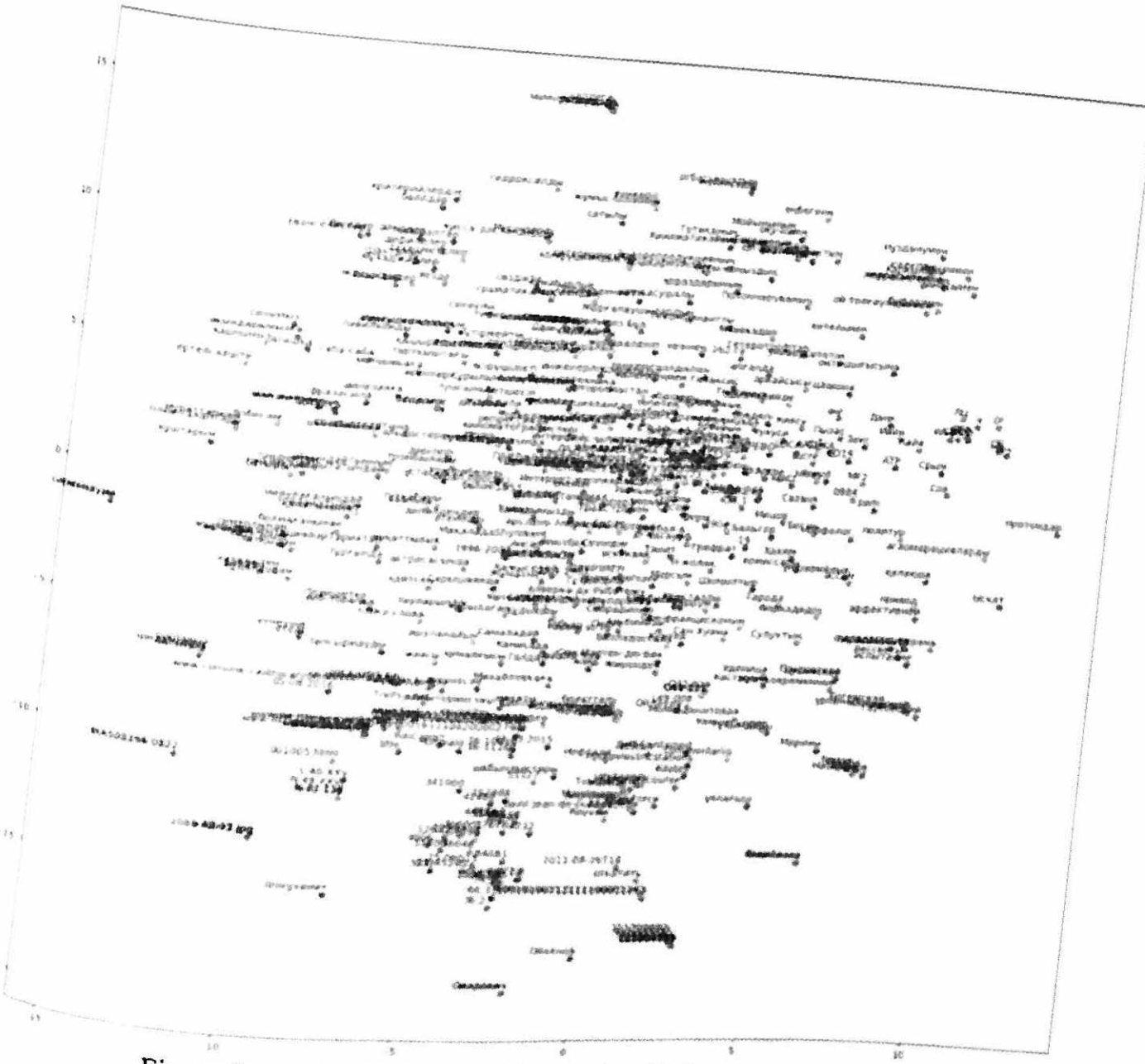


Figure 7: Apply t-SNE algorithm on Kazakh language vectors

Chapter 4. Practical part

4.1 Neural Network

Neural networks are one of the areas of research in the field of artificial intelligence, based on attempts to reproduce the human nervous system. Namely: the ability of the nervous system to learn and correct mistakes, which should allow simulating, albeit roughly enough, the work of the human brain (Figure 8).

The class of tasks that can be solved with a neural network is determined by how the network works and how it learns. At work, the neural network takes the values of the input variables and outputs the values of the output variables. Thus, the network can be used in a situation where you have certain known information, and you want from it to get some information that is not yet known. Here, example of some tasks:

- Classification. The task of a neural network is the distribution of objects over several pre-established non-overlapping classes.
- Prediction of specific numerical values. In such problems, it is required to predict the value of a variable that takes continuous numerical values: the outcome of elections for a particular lot, the stock price, the value of the rating policy, etc.
- Cluster analysis is a multidimensional statistical procedure that collects data containing information about a selection of objects, and then arranges objects into relatively homogeneous groups. The task of clustering refers to statistical processing, as well as to a wide class of learning tasks without a teacher.

Now, to understand how neural networks work, let's take a look at its components and their parameters.

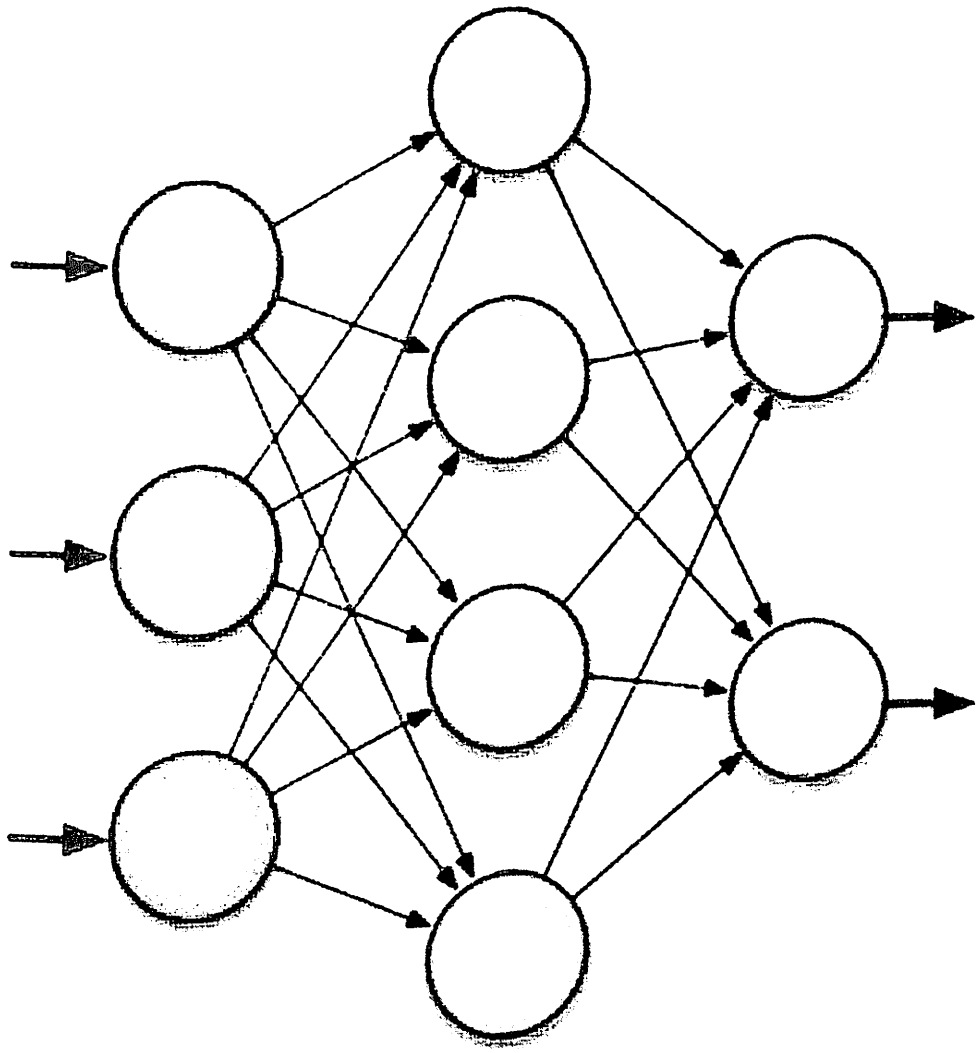


Figure 8: Neural Network

An artificial neuron is a structural unit of an artificial neural network and is an analog of a biological neuron.

From the mathematical point of view, the artificial neuron is the adder of all incoming signals, applying to the resulting weighted sum some simple, in general, nonlinear function that is continuous throughout the domain of definition. Usually, this function increases monotonically. The result is sent to a single output.

Artificial neurons (hereinafter neurons) are combined in a certain way, forming an artificial neural network. Each neuron is characterized by its current state by analogy with the nerve cells of the brain, which can be excited or hindered.

An artificial neuron has a group of synapses - unidirectional input connections connected to the outputs of other neurons, and also has an axon-output connection of a given neuron, with which the signal enters the synapses of the following neurons.

Each synapse is characterized by the magnitude of the synaptic connection or its weight wi, which is the equivalent of the electrical conductivity of biological neurons.

4.2 RNN

Recurrent neural network (RNN) is a kind of neural networks where the links between elements form a directed sequence. This makes it possible to process a series of events in time or successive spatial chains (Figure 9). Unlike multilayer perceptrons, recurrent networks can use their internal memory to process sequences of arbitrary length. Therefore, networks RNN are applicable in such problems, where something integral is divided into segments, for example: handwriting recognition or speech recognition. It was proposed many different architectural solutions for recurrent networks from simple to complex. Recently, the most common network has a long-term and short-term memory (LSTM) and a managed recurrent unit (GRU).

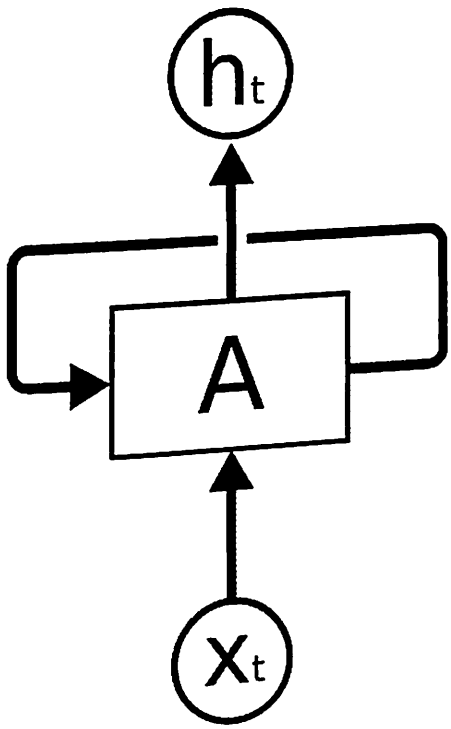


Figure 9: Recurrent Neural Network contain loops

In the 1980s, the development of recurrent networks began. John Hopfield in 1982 proposed the Hopfield Network. In 1993, the neural system for storing and compressing historical data was able to solve the problem of "very deep learning", in which more than 1000 consecutive layers unfolded in the recurrent network.

There are many varieties, solutions and constructive elements of recurrent neural networks.

The difficulty of the recurrent network is that if you take into account each step of time, it becomes necessary for each step of time to create your own layer of neurons, which causes serious computational difficulties. In addition, multilayered implementations are computationally unstable, since they usually disappear or scale off the weights. If you limit the calculation to a fixed time window, the resulting models will not reflect long-term trends. Different approaches try to improve the model of historical memory and the mechanism of memorization and forgetting (Figure 10).

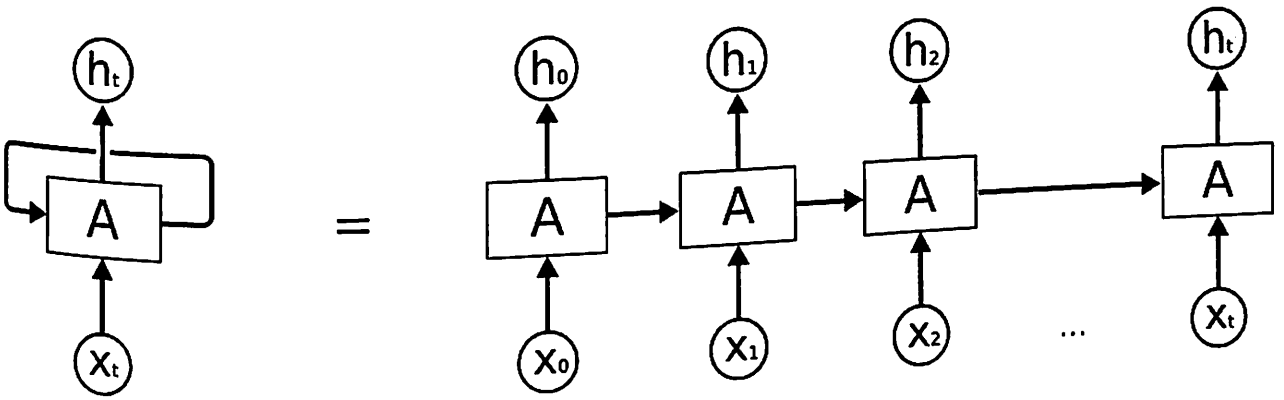


Figure 10: Recurrent Neural Network in the sweep

But the memory realized in this way is very short. Since each time information in memory is mixed with information in a new signal, after 5-7 iterations the information is completely overwritten (Figure 11).

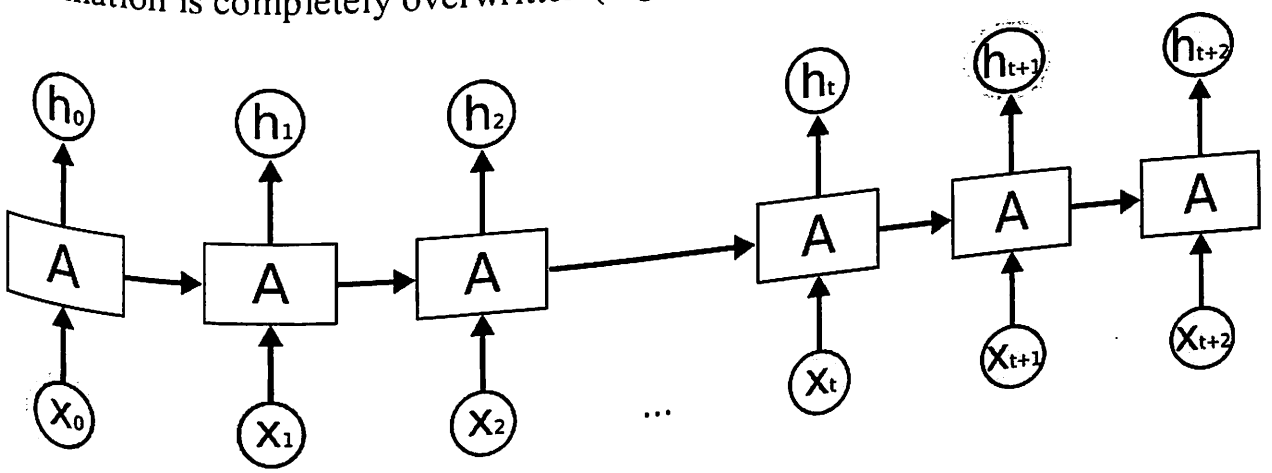


Figure 11: Long-term dependencies in RNN

In theory, problems with the treatment of long-term dependencies in RNN should not be. A person can gently pick up network parameters for solving artificial tasks of this type. Unfortunately, in practice it is impossible to teach RNN to these parameters. This problem was investigated in detail by Sepp Hochreiter (1991) and Yoshua Bengio (co-authors) (1994); they found unquestionable reasons why it could be difficult.

4.3 LSTM

To overcome the flaw with memory, the LSTM-RNN network (Long Short-Term Memory Recurrent Neural Network) was invented, in which additional internal transformations were added which operate with memory more cautiously (Figure 12).

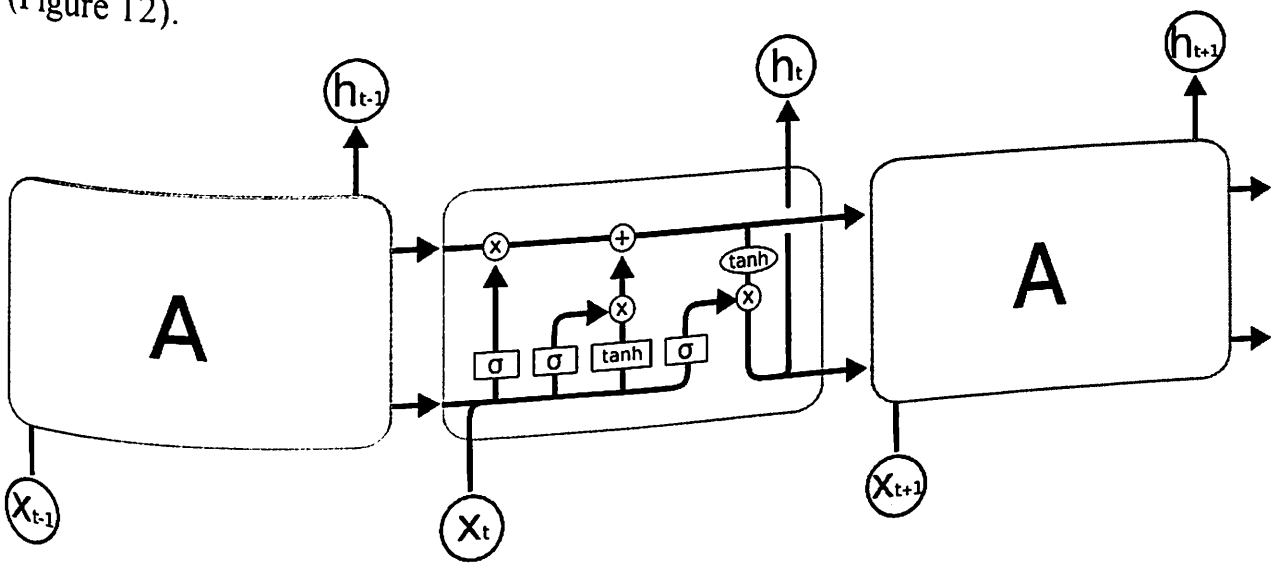


Figure 12: LSTM

Long short-term memory (LSTM) is a kind of architecture of recurrent neural networks, proposed in 1997 by Sepp Hohrater and Jurgen Schmidhuber [29]. Like most recurrent neural networks, the LSTM network is universal in the sense that, with a sufficient number of network elements, it can perform any calculation that a conventional computer can do, which requires an appropriate matrix of weights that can be considered as a program. Unlike traditional recurrent neural networks, the LSTM network is well suited to learning the tasks of classifying, processing and predicting time series in cases where important events are separated by time lags with uncertain duration and boundaries. Relative immunity to the duration of time discontinuities gives LSTM an advantage over alternative recurrent neural networks, hidden Markov models and other training methods for sequences in various fields of application. Of the many achievements of LSTM-networks, one can distinguish the best results in recognition of non-segmented combined handwriting, and the victory

in 2009 in the handwriting recognition competition (ICDAR). LSTM networks are also used in speech recognition tasks, for example, the LSTM network was the main component of the network, which in 2013 reached a record error threshold of 17.7% in the task of recognizing phonemes on the classical body of natural speech TIMIT. As of 2016, leading technology companies, including Google, Apple, Microsoft and Baidu, use the LSTM network as a fundamental component of new products

4.4 Libraries

Keras is a high-level neural networks API, written in Python. As a backend you can run Keras on top of TensorFlow, CNTK, or Theano. Keras is good library for deep learning programs and have high-level API. Main principle is you create model and add to this model your layers. It supports CNN and RNN networks and can run both of them in various combinations. Also, you can run your program on GPU if you want. In my sentiment analysis program I use Sequential model and LSTM layer for Neural Network. Keras have Embedding layer for word embedding and you can add your vector file weights to this layer.

TensorFlow is an open source software library for machine learning developed by Google to solve neural network construction and training tasks with the purpose of automatically finding and classifying images, achieving the quality of human perception. [1] It is used for research and development of Google's own products. The main API for working with the library is implemented for Python, there are also implementations for C ++, Haskell, Java, Go and Swift.

It is a continuation of the closed project DistBelief. Initially, TensorFlow was developed by the Google Brain team for internal use in Google, in 2015 the system was transferred to free access with the open license of Apache 2.0.

Pandas is a high-level Python library for data analysis. Why do I call it high-level, because it is built on top of the lower-level NumPy library (written in C),

which is a big plus in performance. In the Python ecosystem, pandas is the most advanced and fast-growing library for data processing and analysis.

Gensim is a Python library for modeling. More precisely, the thematic modeling of documents and the extraction of similarity from large buildings.

Target audience - people engaged in natural language processing (NLP) and the IR community. In Gensim popular algorithms of NLP are implemented. For example, word2vec. Most implementations of algorithms can use multiple cores.

Gensim I use for reading 300 dimension vector file.

NLTK - it is the powerful library for Natural Language Processing. It contains many datasets, stemmer, tools for lemmatization and bunch of functions for with texts.

NumPy it is the open-source library for working mostly with matrix. Matrix multiplication and other stuff. This library do this very fast, that why it is very popular in data science community.

Capabilities:

- support for multidimensional arrays (including matrices);
- support for high-level mathematical functions intended for working with multidimensional arrays.

Beautiful Soup library mostly used for cleaning data from HTML/XML tags.

Scikit-Learn it is open source library where implemented all the basic algorithms of machine learning.

4.4. Training Data

At the moment there are very few studies in the natural language processing for the Kazakh language. I found impressive researches from Nazarbayev University about Kazakh Language Corpus, Spell Checker and etc. Then, I came up with that nobody made research about sentiment analysis for Kazakh Language.

Main problem with this research that you should have a huge amount of data (>100 million tokens) for word embedding and marked training data (>10k). First, I tried to create my own corpus by parsing Wikipedia articles and get about 21 million tokens, but it is not enough for word embedding. Then, I read Facebook developers paper "Learning Word Vectors for 157 Languages" [4]. They created word vectors for 157 languages trained on Wikipedia and Common Crawl resources. Used CBOW as an algorithm for learning. Also, they have vector files which is trained only on Wikipedia dumps using Skip-gram model.

Then, using Sketch Engine Online tool I parsed several news sites in the Kazakh language like nur.kz, zakon.kz, massaget.kz and etc. Totally, I collected about 4 million words. Then, I cleaned text from HTML tags and do other preprocessing tasks. After, divide documents into sentences using sent_tokenize from NLTK library. Then, I marked each sentence with a number. If sentence negative -1, positive 1, neutral 0. Totally, I marked more than 20k sentences and get ~10k neutral, 5993 negative, 4422 positive sentences (Figure 13).

	sentiment	text
1		Еуровидение 2010 ән конкурсы Евровидениенің 55-ші конкурсы болады .
2	0	Еуровидение 2010 ән конкурсы Евровидениенің 55-ші конкурсы болады .
3	-1	Қой , қазақ бұлармен не ғып соғыса алсын ?
4	1	Қазақстан осы өңірдегі бейбітшілікті қолдайды .
5	0	Дмитрий Медведевтің Астанаға сапары 22 мамырға жоспарланып отыр .
6	1	Біздің елде сізге ерекше құрметпен қарайды .
7	0	- Біздің Үкімет бұл жобаны міндетті түрде қолдайды .
8	0	- Көші-қон мәселелерін реттеу Қазақстан үшін маңызды болып саналады .
9	1	Қазіргі кезде қаланың ішкі өнірлік өнімі 20 есе өсті .
10	0	Кәсіпорында сапа менеджментінің жүйесі енгізілген .
11	0	Жобаны 5 кезеңге бөліп , 12 жылда жүзеге асыру жоспарланып отыр .
12	1	Бұл өте тиімді келісімшарт .
13	0	Әлемде мұндай тәжірибе бар .
14	-1	Есірткі бизнесі де Орталық Азия аймағы үшін қауіп болып отыр .
15	0	Кірпіш өндіруге жергілікті шикізат пайдаланылады .
16	1	- Бас прокуратураның және Бас прокурордың жұмысын қолдаймын .
17	1	Президент бүгінде Астана бүкіл еліміздің игілігі болып табылатының атап өтті .
18	1	Қалыптасу кезеңі табысты аяқталды .

Figure 13: Training Set

4.5 Experiments, comparisons and results

In machine learning we have bunch of algorithms for classification tasks. As I mentioned above Scikit-Learn library implements all the basic algorithms of machine learning. I tried different algorithms with different sizes of training set. Let's train MultinomialNB algorithm with test set size 2% from training set (Figure 14).

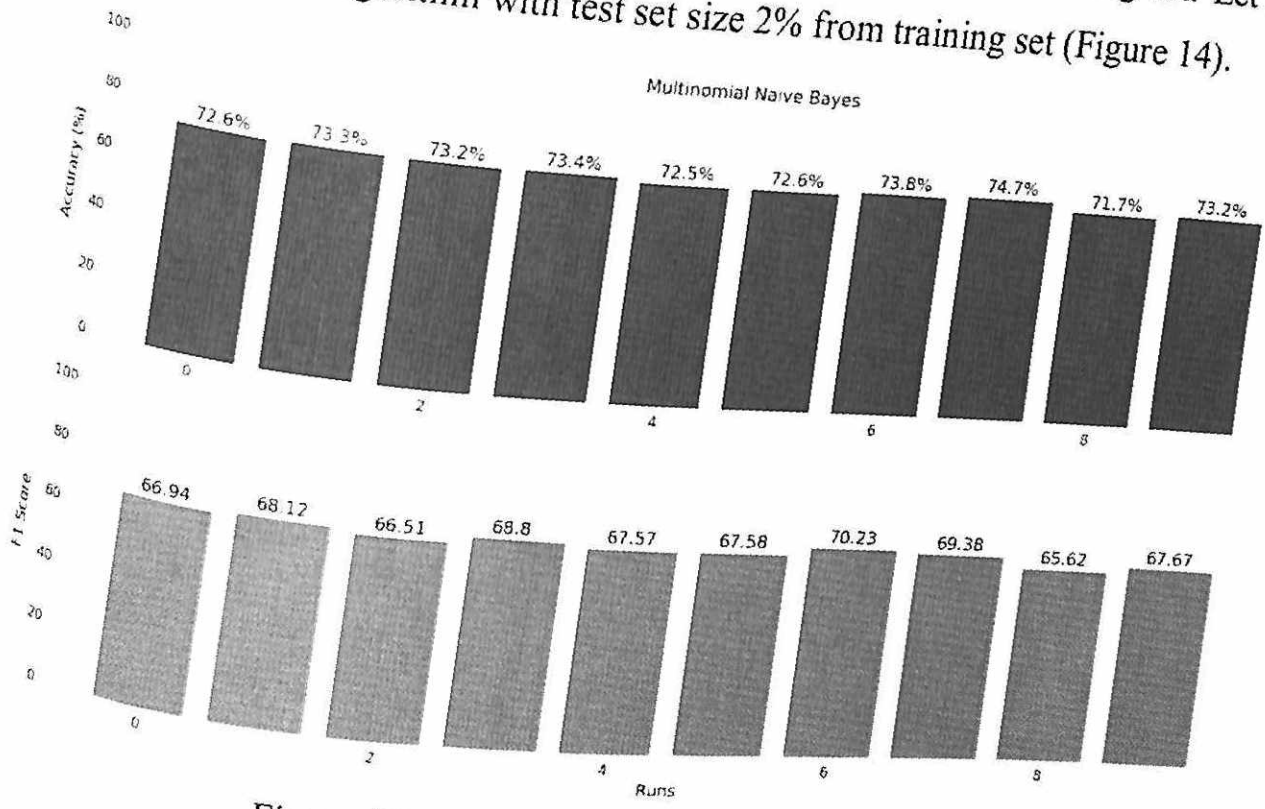


Figure 14: Multinomial Naive Bayes Accuracy

And other classification algorithms (Figure 15).

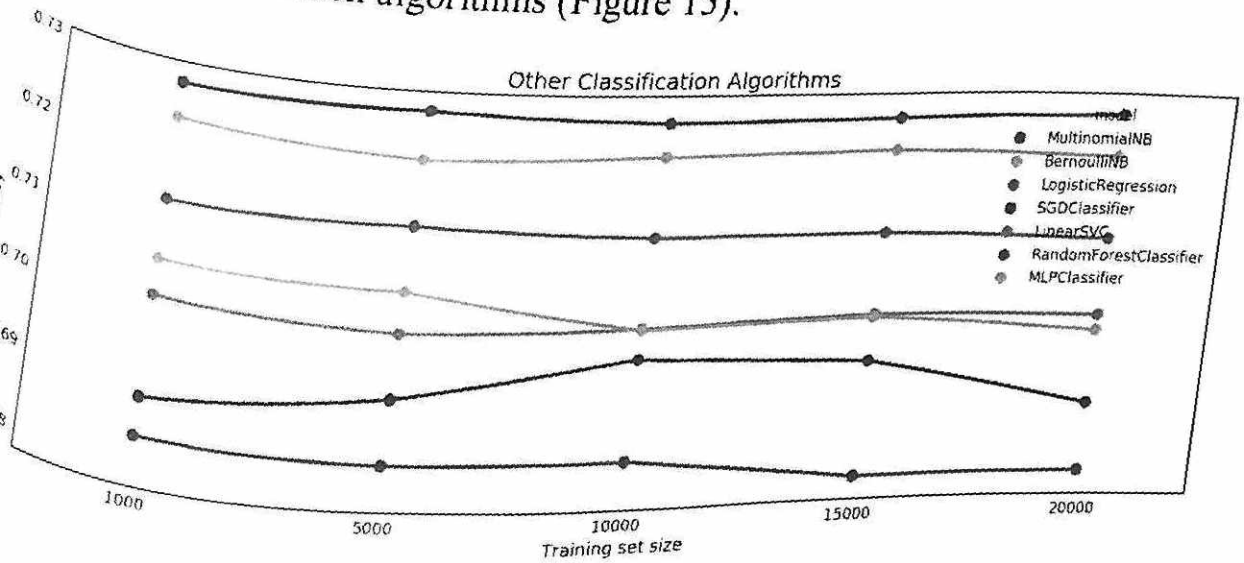


Figure 15: Other Classification Algorithms Accuracy

As you see in Figure 15. MultinomialNB and BernoulliNB algorithms show best results. Let's show confusion matrix for this algorithms (Figure 16, 17).

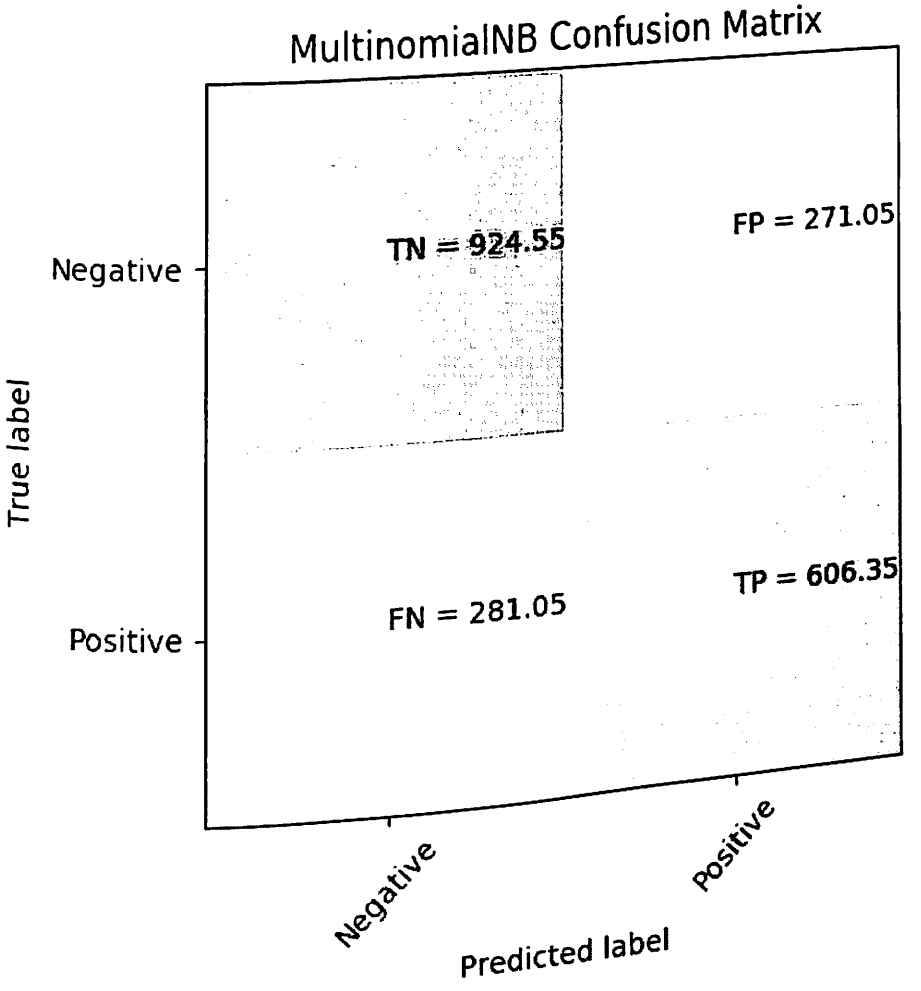


Figure 16: MultinomialNB Confusion Matrix

- 924,55 predicted negative, and was negative. **True Negative.**
- 606,35 predicted positive, and was positive. **True Positive.**
- 281,5 predicted negative, but was positive. **False Negative.**
- 271,5 predicted positive, but was negative. **False Positive.**

All these algorithms trained without word2vec model. Let's now train LSTM network with word embedding layer. Vector dimension is 300 and we have 2 million unique words. LSTM network with Word2Vec show 63,83% accuracy for our full training set. It is less by 10% than MultinomialNB show, but I think LSTM network

in combination with Word2Vec more powerful. We just need more data for word embedding (Figure 18).

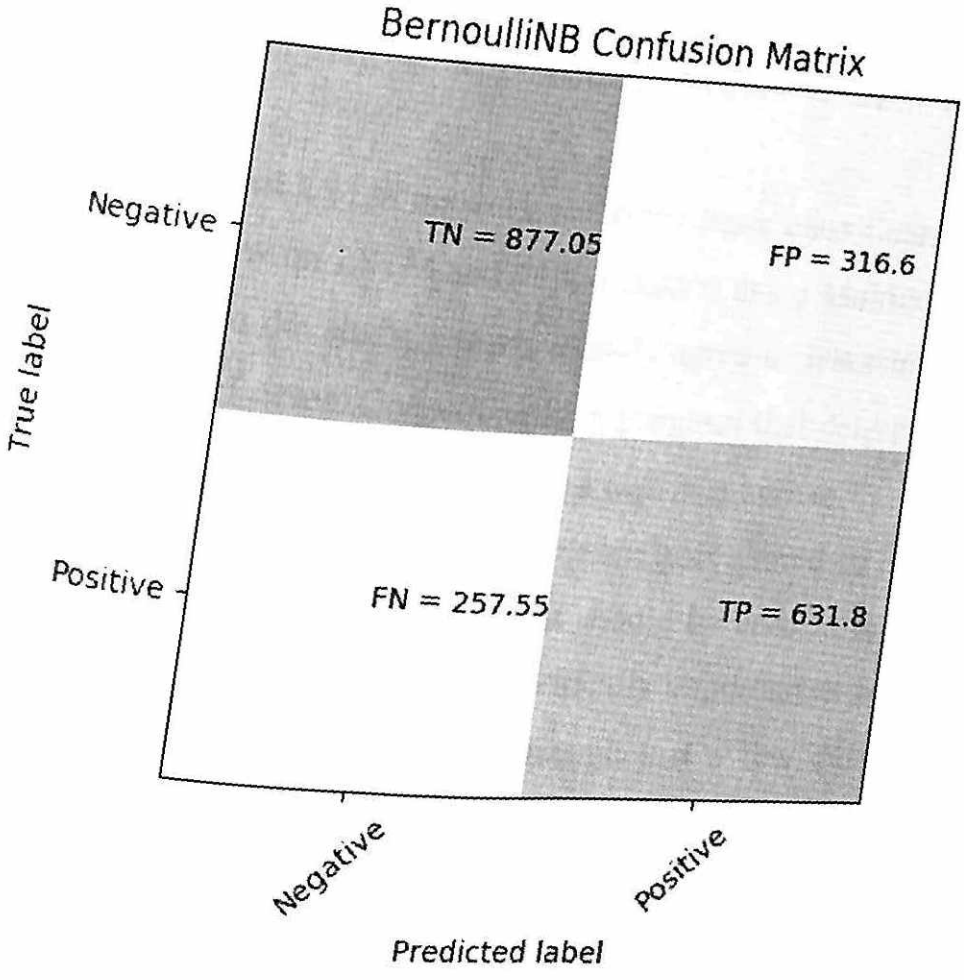


Figure 17: BernoulliNB Confusion Matrix

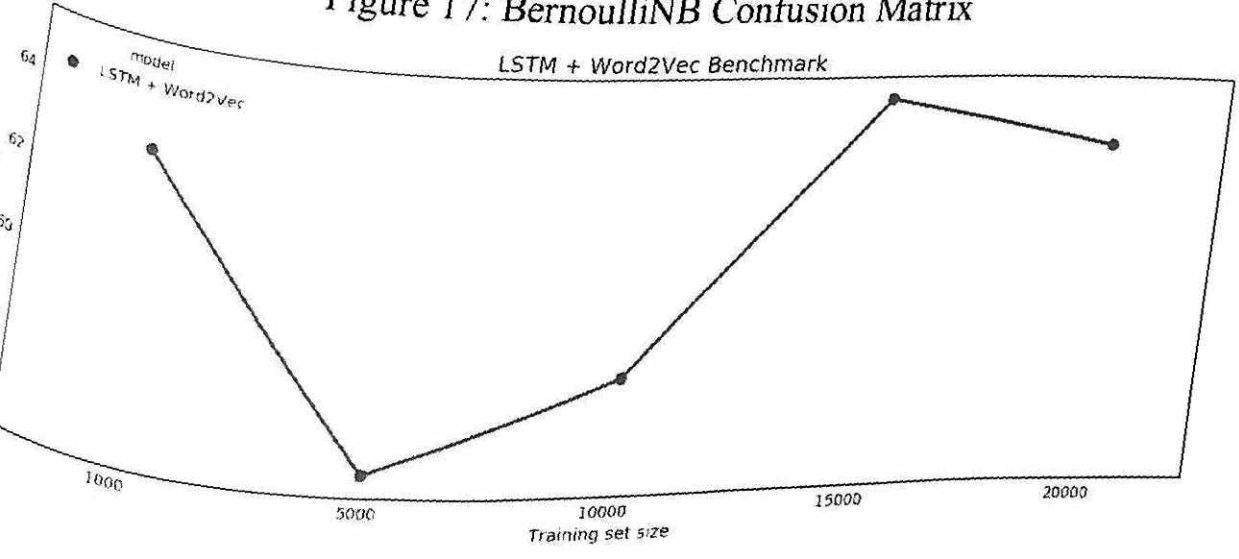


Figure 18: LSTM + Word2Vec Accuracy

Conclusion

Sentiment analysis it is very powerful tool in customer service, social media, politics and etc. And we need to use it in our market and business on users which use Kazakh language everyday. It will open new doors between users and product owners.

For now I trained LSTM network and other basic classification algorithms and get about 63% accuracy on LSTM and 73% accuracy using Multinomial Naive Bayes. However, according to the study, experts usually agree in assessing the tonality of a specific text in 79% of cases. Consequently, a program that determines the tone of a text with an accuracy of 70% does it almost as well as a human.

It needs to be mentioned that sentiment analysis should of course not be used in isolation because it is just one metric that should be used in tandem with others. Additionally, using it in the right context is critically important as well.

When monitoring feedback, going through just a few data points including tweets, comments, reviews and so on, can be most effectively done through simply reading the comments. However, if you have thousands and sometimes even hundreds of thousands of data points on a consistent basis, reading all the feedback can be difficult if not impossible. This is where sentiment analysis can be of great help in providing directional insight and for paving the way for further analysis, insight and appropriate action.

References

1. Janyce Wiebe M. (1994). Tracking point of view in narrative. *Computational Linguistics 20* (2): 233-287.
2. Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2, 1-2 (January 2008), 1-135. DOI=http://dx.doi.org/10.1561/1500000011
3. Alexandra Balahur, Andrés Montoyo. 2008/3. Applying a culture dependent emotion triggers database for text valence and emotion classification. Procesamiento del lenguaje natural 40: 107-114
4. Benko, Vladimir. "A R A N E A :" *О Проекте - Сообщество "Письменное Наследие"*, Vladimir Benko, textualheritage.org/content/view/641/276/lang,russian/
5. Francis, W. and Kucera, H. *Brown Corpus Manual*. [online] Clu.uni.no. Available at: http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM [Accessed 21 May 2018]
6. Burnard, L. *[bnc] About the British National Corpus*. [online] Natcorp.ox.ac.uk. Available at: http://www.natcorp.ox.ac.uk/corpus/index.xml [Accessed 21 May 2018].
7. Ruscorpora.ru. *Russian National Corpus*. [online] Available at: http://ruscorpora.ru/en/index.html [Accessed 21 May 2018].
8. Olzhas Makhambetov, Aibek Makazhanov, Zhandos Yessenbayev, Bakhyt Matkarimov, Islam Sabyrgaliyev, and Anuar Sharafudinov. 2013. Assembling the Kazakh Language Corpus. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1022–1031, Seattle, Washington, USA, October. Association for Computational Linguistics.
9. Madieva, G., Daniel, M. and Umatova, Z. (n.d.). *Almaty Corpus of Kazakh*. [online] Available at: http://web-corpora.net/KazakhCorpus/search/?interface_language=ru [Accessed 21 May 2018]
10. Sketchengine.eu. (n.d.). *kkWaC – Kazakh corpus from the web | Sketch Engine*. [online] Available at: https://www.sketchengine.eu/kkwac-kazakh-corpus/ [Accessed 21 May 2018]
11. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10* (EMNLP '02), Vol. 10. Association for Computational Linguistics, Stroudsburg, PA, USA, 79-86. DOI: https://doi.org/10.3115/1118693.1118704
12. Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. arXiv preprint arXiv:cs/0212032v1

13. Bo Pang and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (ACL '05). Association for Computational Linguistics, Stroudsburg, PA, USA, 115-124. DOI: https://doi.org/10.3115/1219840.1219855
14. Snyder, B., & Barzilay, R. (2007). Multiple Aspect Ranking Using the Good Grief Algorithm. *HLT-NAACL*.
15. Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.* 61, 12 (December 2010), 2544-2558. DOI=http://dx.doi.org/10.1002/asi.v61:12
16. Fangzhong Su and Katja Markert. 2008. From words to senses: a case study of subjectivity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1* (COLING '08), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 825-832.
17. Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (ACL '04). Association for Computational Linguistics, Stroudsburg, PA, USA, Article 271 . DOI: https://doi.org/10.3115/1218955.1218990
18. D. Ustalov, (2012). Извлечение терминов из русскоязычных текстов при помощи графовых моделей. 62-69.
19. Ilya Menshikov. Анализ тональности текста на русском языке при помощи графовых моделей. УРФУ, Екатеринбург, Россия : конференция. — 2012.
20. Tomáš Mikolov, Kai Chen, Greg D. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781v3
21. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606v2
22. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. arXiv preprint arXiv:1310.4546v1
23. Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. arXiv preprint arXiv:1802.06893v2
24. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606v2
25. Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759v3

26. Yergesh, Banu & Bekmanova, Gulmira & Sharipbay, Altynbek. (2017). Sentiment analysis on the hotel reviews in the Kazakh language. 790-794. 10.1109/UBMK.2017.8093531.
27. Yergesh, Banu & Bekmanova, Gulmira & Sharipbay, Altynbek & Yergesh, Manas. (2017). Ontology-Based Sentiment Analysis of Kazakh Sentences. 669-677. 10.1007/978-3-319-62398-6_47.
28. Abdullin Y.B., Ivanov V.V. Deep learning model for bilingual sentiment classification of short texts. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2017, vol. 17, no. 1, pp. 129–136. doi: 10.17586/2226-1494-2017-17-1-129-136
29. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (November 1997), 1735-1780. DOI=http://dx.doi.org/10.1162/neco.1997.9.8.1735
30. Felix A. Gers and Juergen Schmidhuber. 2000. *Recurrent Nets that Time and Count.* Technical Report. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.
31. Dumps.wikimedia.org. (n.d.). Index of /kkwiki/. [online] Available at: https://dumps.wikimedia.org/kkwiki/ [Accessed 21 May 2018].
32. Chapayev, D. (2018). chapayevdauren/kazakh-language-corpus. [online] GitHub. Available at: https://github.com/chapayevdauren/kazakh-language-corpus [Accessed 21 May 2018].
33. Aibek Makazhanov, Olzhas Makhambetov, Islam Sabyrgaliyev, and Zhandos Yessenbayev. 2014. Spelling Correction for Kazakh. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8404* (CICLing 2014), Alexander Gelbukh (Ed.), Vol. 8404. Springer-Verlag, Berlin, Heidelberg, 533-541. DOI: https://doi.org/10.1007/978-3-642-54903-8_44