Бісікало О.В.

Яхимович О.В.

LEXICAL RELATIONSHIPS-BASED KEYWORDS SELECTION IN ENGLISH TEXTS

НАВЧАЛЬНИЙ ПОСІБНИК

Міністерство освіти і науки України

Вінницький національний технічний університет

Бісікало О.В.

Яхимович О.В.

LEXICAL RELATIONSHIPS-BASED KEYWORDS SELECTION IN ENGLISH TEXTS

НАВЧАЛЬНИЙ ПОСІБНИК

Вінниця

ВНТУ

2020

УДК 004.93:159.95

ББК 32.973.22я73

В26

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 6 від 30 січня 2020 р.).

Рецензенти:

С. В. Голуб, доктор технічних наук, професор

С. Д. Штовба, доктор технічних наук, професор

Р. В. Маслій, кандидат технічних наук, доцент

Бісікало О.В.

В26 Lexical relationships-based keywords selection in english texts : електронний навчальний посібник / Бісікало О.В., Яхимович О.В. – Вінниця : ВНТУ, 2020. – 50 с.

У навчальному посібнику наведено теоретичні основи та технологічні аспекти застосування двох нових взаємопов’язаних методів визначення ключових слів англомовного тексту. Навчальний посібник призначено для студентів англомовної освітньої програми підготовки магістрів “Інформаційні системи та Інтернет речей” зі спеціальності 151 – “Автоматизація та комп’ютерно- інтегровані технології”, а також для студентів спеціальностей галузі 12 – Інформаційні технології, що вивчають дисципліни, пов’язані з обробленням текстової інформації.

УДК 004.93:159.95

ББК 32.973.22я73

Table of contents

METHOD FOR DETERMINATION OF KEYWORDS IN ENGLISH TEXTS BASED ON THE DKPRO CORE

The new method of the keywords determining based on finding the connections between word forms of the English text with the instrumental capabilities of package DKPro Core [1] is suggested in this study book. The method, which is illustrated with examples of analysis, aimed at solving problems of efficient processing of text documents – indexing, abstracting, clustering and classification. As a result of theoretical and experimental studies it is found that the developed method found more keywords, specified by the author of the text, compared to analogues and had better quality characteristics [2]. The proposed method of determining of the keywords differs from existing in that it uses additional information about the complex relationships between members of an English sentence.

Keywords: method, keywords, English, linguistic package, DKPro Core, syntactic analysis.

1. Introduction

The task of keywords allocation from the text appears in librarianship, lexicography and terminology and in problems of information retrieval. Currently, the volume and dynamics of information to be processed in these areas makes particularly relevant problem of automatic identification of keywords. Results of this problem solving can be used for the creation and development of terminological resources and for efficient processing of documents: indexing, abstracting, clustering and classification.

2. Relevance

A large number of available linguistic systems oriented on processing of natural language texts, offer automatic selection of keywords. This functional is based on the methods of determining certain keywords, which are divided into linguistic and statistical. Linguistic methods are based on the meaning of words, especially using ontologies and semantic data of words. Unfortunately, these methods are intensive in the early stages: development of ontologies, for example, is very time-consuming process [4]. On the other hand, statistical methods involving large volumes of "verbal noise", which significantly affects the quality of the identification of keywords. Therefore, the most promising for the study for the author's opinion are hybrid methods for which the rate of statistical text processing is enhanced by the capabilities of modern language packages.

3. Description of the method

Consider the problem of determining keywords text as a information technology, which has the text input and output - the plural with *l* keywords

$$W^k = \{w_1^k, ..., w_l^k\}. \quad (1)$$

Without prejudice to the generality, we assume that the text *T* consists of *m* different words, and its separate *j* sentence with *k* has *n* words from *m* possible words, and $m >> n$ and $m >> l$. Most of the known methods for determining keywords taking as a basis the frequency dictionary of text that is actually a list or set of ordered pairs

$$D = \{< w_i, \quad f_i >\}, \quad i = \overline{1,m}, \quad (2)$$

where w_i – one word of *m*, and f_i – its frequency ($f_i \geq f_{i+1},\ i=\overline{1,m-1}$), defined for *T*. For some filtering insignificant individual categories of words as the first *l* keywords from the list *D*, namely simplistically have

$$W^k = \{w_1, ..., w_l\}. \tag{3}$$

However, the results of parsing natural languages using available modern linguistic packages for operating syntactic relationships between words in separate sentences on the program level [5]. In addition, the possibility of these packages can significantly reduce the value of *m* by associating words in word forms, and next - in the lemma and stemma. Therefore, it is necessary to find out advantages for formal definition

$$W^k = \{w_1^k, ..., w_l^k\}$$

provide soft- and linguistic ware procedures parsing of sentences of text *T* according to the (1)-(3).

From the information point of view, understanding of the meaning of the sentence for separate individual accompanied by recognition a) the individual words of which it is composed and b) relationships between pairs of words with the proper construction of connections tree [6]. We assume that these processes occur through a comparative analysis and attraction information from some of the general knowledge base of understanding of the subject. If each of these stages is accompanied by an increase in information, take a working hypothesis:

- The level of common understanding of *T* may vary from the minimum to the maximum possible depending on the volume and other parameters of the general knowledge base of the subject;
- Quality definition (4) proportional to the level of general understanding of the text that should be confirmed by formal features.

Let any *j* sentence with *k* consists of *n* different words, not being tough enough restraint. Then coherent paired dependency tree of the sentence has or *n-1* branches, without taking into account the inverse relationship between the subject and the predicate, or *n* - if we take. Accordingly, the total number of words in this sentence for further in-depth analysis or increases to $2 \times n-2$ or $2 \times n$. However, this increase is uneven - not for all terminal (end) node tree corresponding frequency words do not change, and for the terminal (intermediate) could rise significantly. In Table 1 shows the cases of change frequency words given pair of dependencies for different types of sentences.

Table 1

Analysis of the significant increase in the frequency of words due consideration paired dependencies for different types of sentences

№ з/п	Composition sentence / word count	Type of sentence and its dependency tree graph	Frequency formula	End frequency
1	*Ab / 2*	Collocation (Tree roots)	*A+b*	2
2	*Abc / 3*	Linear triple (Save treasures of nature)	*A+2b+c*	4
3	*Abcd / 4*	Linear tetrad (Got a strange word translation)	*2A+b+c+2d*	6
4	*Abcde / 5*	Branching (Thick forest ended abyss from nowhere)	*A+2b+3c+d+e*	8
5	*Abcdef / 6*	Group of the subject (Blue narrowed eyes of lover saying much)	*A+b+4c+d+2e+f*	10
6	*Abcdef / 6*	Predicative group (Experienced horse furrow quickly feel the smell)	*A+2b+c+d+4e+f*	10
7	*Abcdef / 6*	Both groups (Old grandfather Aeolus collected all the winds)	*A+3b+c+2d+e+2f*	10

Even a rudimentary analysis at one sentence shows that it increases the frequency of the words that may belong to the set of key. Draw a formal assessment of the restrictions imposed to increase the availability of extremely different words in a sentence and not taking into account the inverse relationship between the subject and the predicate:

1. The minimum increase is absent provided of *i* words with m among non-terminal (end) node trees of each sentence:

$$f_i^{\min} = 0, \quad f_i^{new} = f_i \quad i = \overline{1,m} . \tag{4}$$

2. If *i* word located in each of the *k* sentences text and, in addition, each sentence corresponds most extensive terminal nodes, the maximum increase in frequency is

$$f_i^{\max} = \sum_{j=1}^{k} (n_j - 2), \; i = \overline{1,m} \; . \tag{5}$$

Respectively

$$f_i^{new} = f_i \; + f_i^{\max} = k + \sum_{j=1}^{k} (n_j - 2) = \sum_{j=1}^{k} (n_j - 1) \; . \tag{6}$$

3. In general and more real case $f_i = z \mid z \le k$, is the *i* word located at *z* sentences with *k* have according to the (4)-(6)

$$f_i^{new} = z + \sum_{j=1}^{z} (n_j - 2) = \sum_{j=1}^{z} (n_j - 1)$$

as the upper bound increase in the frequency of *i* word.

When huge bodies of text need to be analyzed or when language processing capabilities are to be integrated into devices, automatic analysis is the way to go. Once a sufficiently large body of manually analyzed data is available, it can be used to train statistical models using machine learning algorithms and to evaluate their results against the manually created gold standard. Even for unsupervised machine learning algorithms, i.e. algorithms not requiring any training data, evaluation against a manually created gold standard is indispensable. In a research context, automatic analysis is often embedded into an experimental setup which is iteratively repeated

with different variations of analysis components or their configurations to reach optimal results.

When doing manual analysis, annotation guidelines define how to locate the phenomena to be annotated and which categories to use for classifying them. These guidelines are often maintained as simple text documents which can be understood by human annotators, but which are not machine readable. For automatic analysis, the annotation type system plays a similar role. It is a computer-processable, formal document describing the annotation types, their features and tag sets. A typical annotation type system defines a type Token which carries a feature part-of-speech that assumes a value such as noun. Annotation type systems are defined differently, depending on the formalism used to represent the actual annotations, and often depending on the specific analysis tool or processing framework they are written for. Sometimes, standard formats like XML Schema or the Web Ontology Language (OWL) are used to define an annotation type system.

When designing an annotation type system, certain aspects are usually underspecified, in particular such aspects that are explicitly defined in annotation guidelines. Consider the part-of-speech feature mentioned above. An annotation guideline should explicitly define which part-of-speech tags exist and how they can be distinguished. This set of categories, or tag set, constitutes the values which the part-of-speech feature may assume. Annotation type systems, however, are often meant to be reusable in different contexts, e.g. for different languages or by scientists of different schools, which categorize parts of speech differently or using a different granularity. Hence, the annotation type system usually defines that the part-of-speech feature exists on the Token, but not the values it may assume. If the formalism underlying the annotation type system supports inheritance or some other form of extensibility, some designers may introduce a specialization of the Token, e.g. a FooToken which accepts only part-of-speech tags from a hypothetical foo tag set [7].

There has been developed software based on DKPro Cor for experimental verification of theoretical analysis.

DKPro Core is the set of software components for natural language processing based on the Apache UIMA framework. It was built to improve the productivity of researchers working on automatic analysis of language. DKPro Core approach is that researchers should be able to focus on their real scientific issues, not on technology development.

The collection aims to attain this goal by following these principles:

a) Choice – for most analysis steps, integrated multiple different tools from different vendors. DKPro Core 1.5.0 covers analysis tasks from coreference resolution, chunking, decompounding, language identification, lemmatization, morphological analysis, named entity recognition, syntactic parsing, dependency parsing, part-of-speech tagging, segmentation, semantic role labeling, spell checking, to stemming. For each task, up to seven different tools have been integrated. Additionally, 19 different data formats are supported.

b) Coverage – for many of the analysis components, multiple sets of resources for different languages have been packaged and integrated. DKPro Core 1.5.0 integrates 94 models in 15 languages.

c) Interchangeability – DKPro Core set up a naming convention for component parameters, and where feasible take care that they accept the same settings across different components. For example, the parameters to manually select a model or to override the mappings for elevated types have the same names regardless of the component. So a user substituting one component for another does not have to learn a completely new set of parameters. At times, even only changing the name of the implementation is sufficient, with the parameters and parameter values remaining the same.

d) Portability – analysis components are downloadable and run on different system platforms, either by means of the Java virtual machine, or by providing binaries compiled for different operating systems. DKPro Core integrates with Maven infrastructure in order to provide properly versioned artifacts and deploy these to the user’s system. This is an important step towards the creation of portable and

reproducible workflows. To maintain maximum control over the processing, NLP web services are explicitly excluded from DKPro Core.

e) Usability – analysis components require only minimal mandatory configuration and many components require no mandatory configuration at all because based on the processed data they can automatically determine which resources, e.g. parser or part-of-speech tagger models, are required. Many of the DKPro Core components are also capable of automatically downloading resources at runtime, depending on the data being processed [7, 8].

Identification of keywords has several stages:

a) Creating a multi-text markup.

b) Syntactic markup, taking into account the complex relationships between pairs of lemmas.

c) The replacement of pronouns in pairs obtained in accordance with these nouns.

d) Splitting pairs on individual words and determine the number of connections.

e) Removing words, that refer to Part of Speech(POS) Tags: CC, CD, DT, EX, IN, LS, MD, PDT, POS, PRP, PRP$, RP, SYM, TO, UH, WDT, WP, WP$, WRB.

f) Removing stop words [9].

g) Selection of the first n words with the largest number of connections, where n - number of required keywords.

POS Tags, for stage e):

CC – coordinating conjunction: and, but, nor, or, yet, plus, minus, less, times (multiplication), over (division). Also for (because) and so (i.e., so that).

CD – cardinal numeral (one, two, 2, etc.).

DT – determiner: articles including a, an, every, no, the, another, any, some, those.

EX – existential there: unstressed there that triggers inversion of the inflected verb and the logical subject (there was a party in progress).

IN – preposition or subordinating conjunction.

LS – list item marker: numbers and letters used as identifiers of items in a list.

MD – modal: all verbs that don't take an -s ending in the third person singular present: can, could, dare, may, might, must, ought, shall, should, will, would.

PDT – predeterminer: determiner like elements preceding an article or possessive pronoun; all/PDT his marbles, quite/PDT a mess.

POS – possessive ending: nouns ending in 's or '.

PRP – personal pronoun (I, you, she).

PRP$ – possessive pronoun, such as: my, your, his, his, its, one's, our, and their.

RP – particle: mostly monosyllabic words that also double as directional adverbs (about, off, up).

SYM – symbol: technical symbols or expressions that aren't English words (+, %, &).

TO – literal to.

UH – interjection, exclamation: such as my, oh, please, uh, well, yes.

WDT – wh-determiner (what, which).

WP – wh-pronoun: includes what, who, and whom.

WP$ – possessive wh-pronoun: includes whose.

WRB – wh-adverb: includes how, where, why. Includes "when" when used in a temporal sense [10].

Analogues of application can be SEO optimization websites, which have a function of determining keywords. For this experiment, selected services: advego.ru/text/seo/, rise-top.com/keywordstext.php та seotool.by/analiz/seo/key-wordstext.php.

For the experiment were taken text with 1460 words «A new pattern for historical geography: working with enthusiast communities and public history» [11].

The keywords specified by the author: Participation, Public history, Enthusiast communities, Museums, Heritage.

Results of keywords for our development and analogues are presented in Table 2. Near to the keywords, specified position of location of the keyword.

Table 2

Results of searching keywords

keywords specified by the author		our development		rise-top		advego		seotool	
1	Participation		work		historical		historical		historical
2	Public	5	community	4	enthusiast	4	enthusiast	4	enthusiast
3	history		geography	5	communities		for	5	communities
4	Enthusiast	1	participation	1	participation	5	community	1	participation
5	communities	4	enthusiast		geography		this		work
6	Museums		geographer		work	6	museum		geography
7	Heritage	6	museum		research		geography		new

Determine the quantitative characteristics of the results, namely completeness (for Jaccard and absolute) and accuracy (in Euclidean and Manhattan distances).

Jaccard completeness, in this case, is the fraction of the number of keywords found in the difference number of possible keywords specified by the author and found software (in this case to 7) and the number of found keywords.

Absolute completeness is a ratio of correctly found keywords to the number of keywords.

In Fig. 1 shows a histogram of completeness for Jaccard and absolute for our development and analogues.

Euclidean distance is given by:

$$d_e = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \quad (7)$$

where n – number of keywords.

x_i – position i keyword specified by the author.

y_i – position *i* keyword specified program.

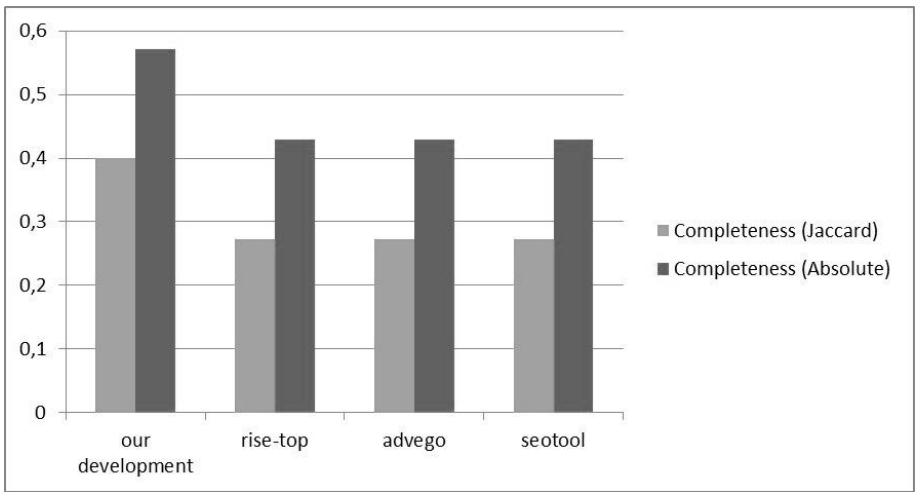

Fig. 1. Histogram of completeness for Jaccard and absolute

Manhattan distance is given by:

$$d_{\mathrm{m}} = \sum_{i=1}^{n} |x_i - y_i| \tag{8}$$

In Fig. 2 shows a histogram of Euclidean and Manhattan distances for our development and analogues according to the (7)-(8).

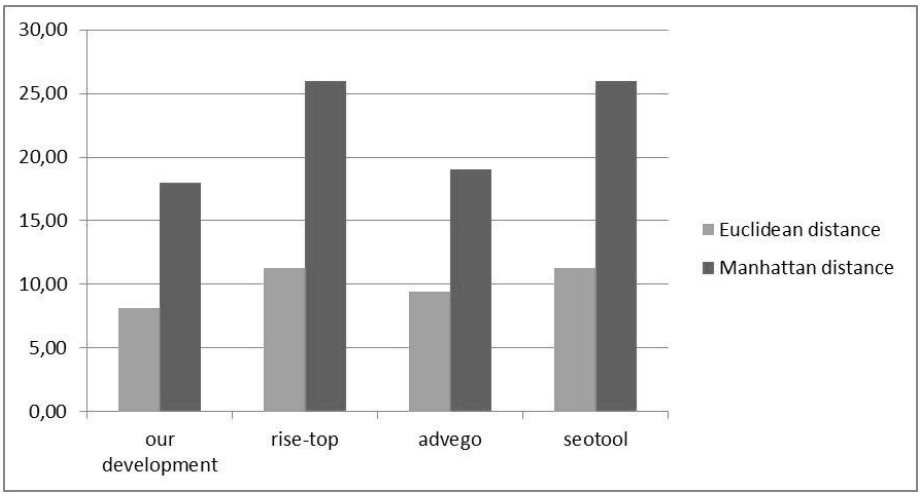

Fig. 2. Histogram of Euclidean and Manhattan distances

Completeness of keywords should be greatest possible, and the distance between the positions of keywords specified by the author and defined

programmatically as small as possible. As seen from the histograms our result has better quantitative characteristics over analog - by 31.8% and 25% for completeness, and 14% and 5.3% accuracy.

For the experiment were taken text with 59892 words «The Hound of the Baskervilles». The etalon keywords: Sherlock Holmes, Baskerville, hound, doctor Watson, Mortimer, family curse, servant Barrymore [12].

Results of keywords for our development and analogues are presented in Table 3.

Table 3

Results of searching keywords for «The Hound of the Baskervilles»

etalon keywords		our development		advego		rise-top		seotool	
1	sherlock	3	baskerville		that		his		his
2	holmes		agent		this	2	holmes	2	holmes
3	baskerville		stranger		for		said		said
4	hound		gentleman		will		your		your
5	doctor	7	mortimer		sir		man		man
6	watson	2	holme		there		had		had
7	mortimer	11	barrymore	2	holme		very		stick
8	family		night		said		stick		very
9	curse		time		one		been		has
10	servant		baronet		can	7	mortimer	6	watson
11	barrymore		science	3	baskerville		has		our

In Fig. 3 shows a histogram of completeness for Jaccard and absolute for our development and analogues.

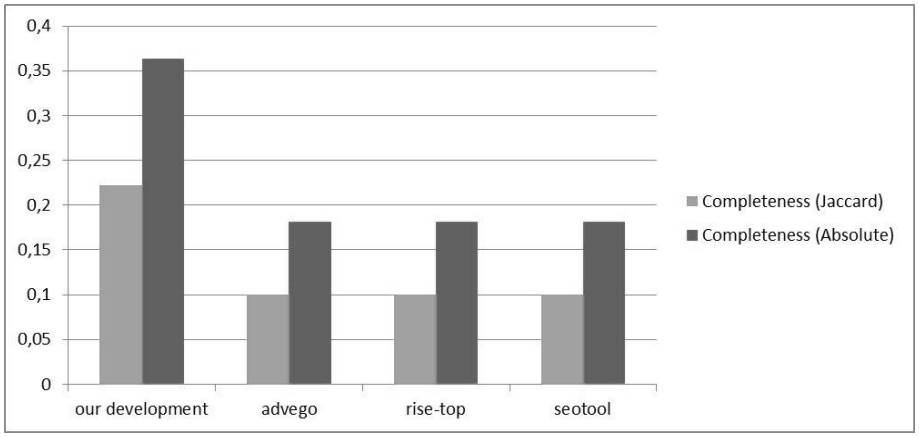

Fig. 3. Histogram of completeness for «The Hound of the Baskervilles»

In Fig. 4 shows a histogram of Euclidean and Manhattan distances for our development and analogues.

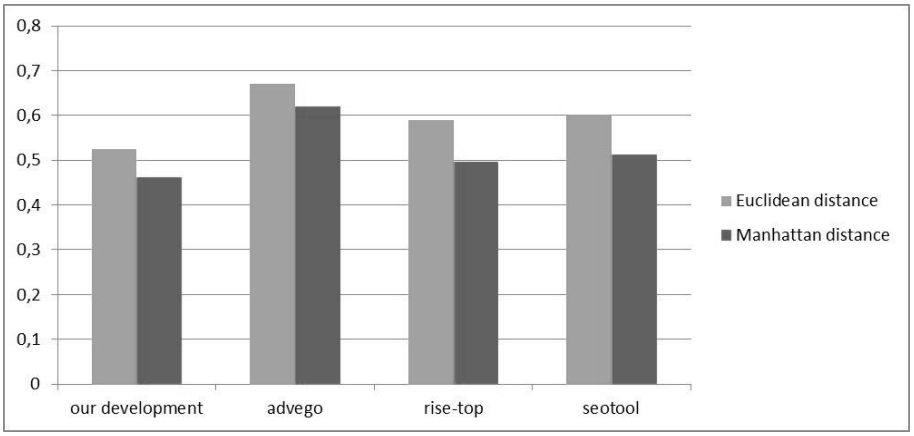

Fig. 4. Histogram of Euclidean and Manhattan distances for «The Hound of the Baskervilles»

Our result has better quantitative characteristics over analog – by 22.2% and 36.4% for completeness, and 52.6% and 46.3% accuracy.

4. Conclusions

1. As best quality is achieved linguistic text processing methods or in combination with statistical, the system of automatically determine of the key phrases from the text natural language should be developed using morphological dictionary (lexicon) and syntax rules. This data previously determined and stored in a database. The text to be processed analyzer that produces information about the division of the text into paragraphs, sentences and single words, it is necessary for further processing. Every word highlighted analyzer undergoes morphological analysis to build morphological interpretation, definition and formation stemming lemma. Based

on the current interpretation of the text is performed construction and filling of syntactic groups and identify relationships between them.

2. The study book shows a method for determining the keywords based on the use of additional information about the complex relationships between members of the English sentence. For the functional implementation of text analyzer selected popular linguistic package DKPro Core. Experimental study of theoretical substantiation method confirmed its qualitative and quantitative advantages in comparison with known counterparts. For the English text of the amount received in 1460 words increase the completeness determination keywords (31.8% for the Jaccard and 25% in absolute value) and improve accuracy (14% in the Euclidean and Manhattan distances 5.3%).

3. Enlarge the text to around 60000 words demonstrate similar results in determining the completeness of keywords (22.2% for the Jaccard and 36.4% in absolute terms) and a appreciable increase in the accuracy of identifying keywords (52.6% in the Euclidean distance and Manhattan 46 3%) in comparison with analogues. These results suggest a minimum of stability of the proposed method.

4. The quality of the results could potentially increase by a separate analysis of parts of speech, because the probability of relevance of the keyword, for example, noun and adverb will be different. In addition, it is necessary to estimate the increase in frequency indicators for keywords by implementing available DKPro Core components to determine relationships.

5. Questions for self-control and improvement

1. What tasks of efficient processing of text documents does the proposed method allow?

2. What additional information does the proposed method use to improve quality indicators of keyword determination?

3. What recognition types are equal to understanding the meaning of the sentence for separate individual?

4. What is the computational complexity of the process of keyword determination?

5. How is the computational complexity of keyword determination decreased by the proposed method?

6. What do you know about Apache UIMA framework?

7. What do you know about DKPro Core language package?

8. What principles are in the basis of the DKPro Core 1.5.0 collection?

9. How many models and languages are supported by DKPro Core 1.5.0?

10. What stages assure the identification of keywords by the proposed method?

11. What experimental studies have been conducted to test the proposed method?

12. How much better are quantitative characteristics of the proposed method compared to analogues by completeness and accuracy?

METHOD FOR FILTERING VERBAL NOISE DURING THE PROCESS OF FINDING KEYWORDS FOR AN ENGLISH TEXT

The object of study is the processing of verbal information to identify keywords in the text. The most important step in the search for key terms is the calculation of their weights in the document in question, which makes it possible to evaluate their significance relative to each other in this context [1]. To solve this problem, there are many approaches that are conditionally divided into two groups: they require learning and do not require learning. Learning implies the need to pre-process the original body of texts in order to extract information about the frequency of occurrence of terms in the entire body. An alternative approach is using linguistic ontologies, which are more or less approximate models of the existing set of words in a given language. On the basis of both approaches, systems are created for the automatic extraction of key terms. Nevertheless, in the direction of searching for keywords, research is not stopped in order to improve the accuracy and completeness of the results, as well as to use methods of extracting information from the text to solve new problems.

Existing approaches to the definition of keywords are characterized. The best quality of text processing is achieved by linguistic methods or when their combinations are statistical. A system for automatically determining key phrases from natural language text should be developed using the morphological dictionary and syntax rules.

The study uses an approach to defining keywords based on finding syntactic links between word forms in sentences in English text using the instrumental capabilities of modern linguistic packages [3]. In the framework of the general approach to reducing verbal noise in the method, it is proposed that it is achieved with the help of formalized operations: the replacement of pronouns with the corresponding nouns; removal of noise connections; removing noise words; withdrawal of stop words. The described operations can be used as additional

modules that improve the results of finding keywords for both the developed method for determining keywords of English text and other algorithms for finding keywords.

Keywords: verbal noise filtering, English text keywords, linguistic package, DKPro Core, syntactic analysis.

1. Introduction

At present, the volume and dynamics of information to be processed in lexicography and terminology, as well as in information retrieval tasks, make the task of automatically determining keywords especially important. Very actively in modern information technologies (IT) they use keywords to create and develop terminological resources for efficient processing of documents, in particular, indexing, summarization, clustering and classification [4].

There are a large number of automatic keyword extraction systems available that are designed and developed for processing natural languages. These systems are based on certain methods for determining keywords, which are divided into linguistic and statistical. Linguistic methods are based on the meanings of words, in particular, they use ontologies and semantic data about a word. These methods are resource-intensive at the early stages: development of ontologies, for example, is a very labor-intensive process [4]. On the other hand, statistical methods are accompanied by significant amounts of "verbal noise", which significantly affects the quality of the definition of keywords. Therefore, hybrid methods are the most promising for research, for which the speed of statistical text processing is enhanced by the capabilities of modern linguistic packages.

The relevance and practical value of the research direction is that the found keywords can be used to improve the accuracy of the analysis of site content and raise the position of the site in search results.

Keyword – a word in the text that can, in conjunction with other keywords, represent the text. The set of keywords is close to the annotation, plan and outline,

which also represent a document with less detail, but, unlike keywords, associated with syntactic structures.

Verbal noise or noise words – a term from the theory of information search by keywords. These are words that do not carry a semantic load, so their use and role for the search is irrelevant [13].

In the process of processing, an exception is made to the words from the text under study, which, by definition, cannot be meaningful to what constitutes "noise". In contrast to the key, these words are called neutral or stop (stop words). These are words related to the official parts of speech, as well as pronouns [14].

2. Object of research and its technological audit

The object of research is the process of processing verbal information to identify keywords in the text.

The subject of research is methods for finding keywords in the text, as well as approaches to reducing verbal noise in the process of searching for keywords.

Keywords have a number of essential features:

- high degree of repeatability of these words in the text, the frequency of their use;

- ability of a sign (words as verbal signs of a certain concept) to condense, collapse information, expressed in whole text, to combine "its main content". This feature is particularly pronounced in the keywords in the title position.

Having a properly selected set of keywords will allow to:

a) quickly find the article to the user when searching the database;

b) to see the article when viewing other similar articles;

c) rather, understand the thematic and terminological area of both one article and the journal as a whole.

All this serves one purpose: to attract the attention of readers to the article, which is the main task of any media [15].

However, the choice of keywords is a very difficult operation and requires a balanced approach. It is necessary to choose the keywords that most accurately reflect the specifics of the topic in question. It is necessary to avoid random and common phrases, it is not recommended to repeat the same keywords several times. So, the process of searching for keywords is analytical [16].

3. Review of existing solutions of the problem

Among the main directions of solving the problem of searching keywords in the text, identified in the resources of the world scientific periodicals, can be highlighted [17, 18]. To separate single keywords using methods based on Zipf's law. Such methods depend on the setting of the range of frequencies in which words significant for the text are found. Since words that occur very often, basically turn out to be verbal noise, and words that occur rarely, in most cases, do not have a decisive semantic meaning. Therefore, in each case, it is necessary to use a number of heuristics to determine the width of the range, as well as techniques that reduce the influence of this width. One of the ways, as indicated in [19], is exceptions, with candidates for keywords, words that can’t be meaningful to the volume components of the noise. But in this paper, noise reduction based on syntactic information is not considered.

The work [20] is devoted to improving the results of calculating the weights of terms based on the TF-IDF algorithm. However, a common feature of such systems is that they require the availability of information obtained from the entire collection of documents. In other words, if the method based on TF-IDF is used to create a document view, then the arrival of a new document in the collection requires recalculation of the term weights in all documents. So, any applications based on the weights of the terms in the document will also be affected. This greatly hinders the use of methods for extracting key terms that require learning in systems where dynamic data flows must be processed in real time [21].

To solve this problem, the TF-ICF algorithm is proposed in [22]. As a development of this idea in [23, 24] it is proposed to use Wikipedia as a learning thesaurus. For calculations, the information contained in the annotated encyclopedia articles with manually selected key terms is used. However, the order of passage of terms in the document and their syntactic role are not taken into account.

An alternative solution to the problem, outlined in [25], involves the use of linguistic ontologies that are more or less approximate models of the existing set of words of a given language. However, these methods are resource-intensive at the early stages: the development of ontologies is a very laborious process.

A method that serves to automatically form a thematic body frame with a WEB is shown in [26]. However, the selection is governed by the timing relationship threshold.

The authors of work [27] emphasize the importance of using nominal groups selected with a parser as candidates for keywords. Although this statement may be considered by other syntactic units used in the definition of keywords.

Seotool is a free online service that will help check the relevant written text of the key words (automatically generating the keys for the specified text). This will help to get a higher ranking in the search engines Yandex and Google, since the page will be the keywords that correspond to the content of the page on which they are placed. Also, this service will help in generating the semantic core of the site (when enabled, remove the HTML code). However, in the generation of keywords and phrases only the first thousand words of the entered text are used.

There is a possibility of a percentage comparison of words with a template. The words of the analyzed text (content) will be compared as a percentage with the list of words of the entire template (text) by morphological analysis. If the percentage equality with any of the words in the template is taken into account, the word is taken into account, otherwise it is not taken into account. The maximum number of words of a pattern should not exceed 250 words [28].

Rise-Top will help to make "sketches" of keywords for the site based on the use of the specified text for the analysis. As a selection of keywords, the words with

the highest density in the order of decreasing their density are applied to the entire text [29]. But in the generation of keywords, only the first 1000 words of the processed text are also used.

Advego is the largest provider of content and related services for Internet sites in RuNet. For optimizers and site owners are offered unique articles, reviews, publications. Promotion in search engines and promotion in social networks are provided. The resource also has the ability to define keywords [30].

Thus, the results of the analysis allow to conclude that the question of developing a method for filtering verbal noise in the process of searching for keywords is promising and requires further study.

4. Description of the method

To improve the accuracy of determining keywords, statistical text processing methods are involved, the speed of which is enhanced by the capabilities of modern linguistic packages.

One such package is the DKPro Core, a set of software components for natural language processing, based on the Apache UIMA framework.

The DKPro Core package is more than a set of analysis components that interact with each other. It was built to improve the productivity of researchers working with automatic language analysis. The approach of DKPro Core is that researchers should be able to focus on their real scientific issues, and not on the development of appropriate technologies [7].

The quantitative characteristics of the relevance of the results, based on the analysis of the literature, are completeness (in Jacquard and absolute) and accuracy (in Euclidean and Manhattan distances). The interpretation of the selected criteria to the conditions of the task of defining keywords is carried out.

Jacquard completeness, in this case, is determined for two sets of keywords – given by the author (reference) and programmatically defined, equal to the ratio of

the number of elements of the intersection of these sets with the number of elements of their union. That is, it is the quotient of the division, where the numerator contains the number of keywords correctly found by the program, and the denominator is the difference between the sum of the elements in the two sets and the number of keywords correctly found.

Absolute completeness is found as a ratio of the number of keywords correctly found by the program with the number of keywords.

The Euclidean distance is determined by the formula:

$$d_e = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2},$$

where n – the number of keywords;

x_i – the position of the i-th keyword defined by the author;

y_i – the position of the i-th keyword defined programmatically.

Manhattan distance is determined by the formula:

$$d_m = \sum_{i=1}^{n} |x_i - y_i|.$$

The use of a pair of formal criteria for completeness and accuracy, will allow a more objective assessment of the relevance of the obtained search results for keywords.

According to [2], this approach to defining keywords is proposed, there are three main stages:

1) creating a multi-level markup of the text;

2) the use of syntactic markup, taking into account the complex dependencies between pairs of lemmas;

3) reduction of verbal noise.

The essence of the approach, in contrast to the known analogues, is determination of the number of links for individual words and the subsequent selection of the first n words with the largest number of links, where n is the number of necessary keywords.

Creation of multilevel markup of text and syntactic markup, taking into account the complex dependencies between pairs of lemmas, is achieved by means of DKPro Core [7].

Verbal noise filtering is proposed to provide with the following operations:

- replacing pronouns with their corresponding nouns;
- removal of noise connections;
- removal of noise words;
- withdrawal of stop words.

Replacing pronouns with their corresponding nouns (replace pronouns) allows reducing the number of pronouns, as well as increasing the number of nouns that can be keywords. For the method of reducing verbal noise when defining key words in English text, it is proposed that the replacement of pronouns is performed by means of DKPro Core [7].

Let’s consider the removal of phrases with types of links that do not carry significant semantic load. As a result of the research, it is revealed that such connections are DET, EXPL, FIXED, PUNCT, REF, ROOT.

DET – the connection of the determinant that exists between the nominally main word and its determinant. Most often, a word that has a tag part of a DET speech will have the same DET identifier connection and vice versa. A well-known exception is that in some of the data sets, the possessive determinant (for example, such as “my”) at some point receives the tag of a part of the DET speech, but the NMOD link, which is parallel to other possessive constructions. But this is not completely the same for different languages, in some languages it is much clearer than in English, it is expressed how the possessive determinants relate to adjectives, therefore the relation NMOD is not subject to doubt [31].

Examples of DET links are shown in Fig. 1.

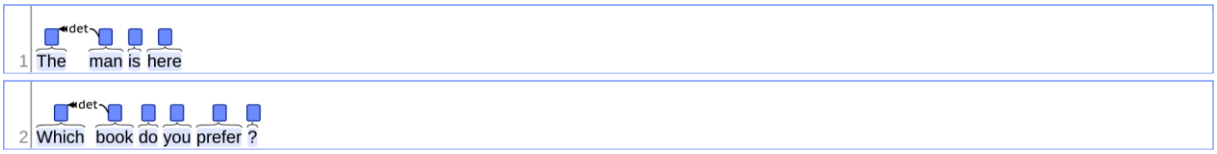

Fig. 1. Examples of DET noise links

EXPL is a relationship that fixes plugin or pleonastic values. Such nominal values appear in the argument position of the predicate, but do not fulfill any of the predicate's semantic roles. The main sentence predicate (verb or predicative adjective or noun) is the main word. In English, this applies to some ways of using it and there: existential there, as well as it when used in exhibition constructions [32].

Some languages have no such English-like expressions, this applies to most pro-drop languages (a speech in which certain classes of pronouns can be omitted when they are pragmatic or grammatically inertial). Also, this phenomenon is often referred to as zero or zero anaphora [33]. In languages with similar utterances, they can be located where the main argument usually appears: the subject and the direct (and even indirect) application [34].

Examples of EXPL links are shown in Fig. 2.

FIXED is used for certain constant grammatical expressions that behave as functional words or short adverbs.

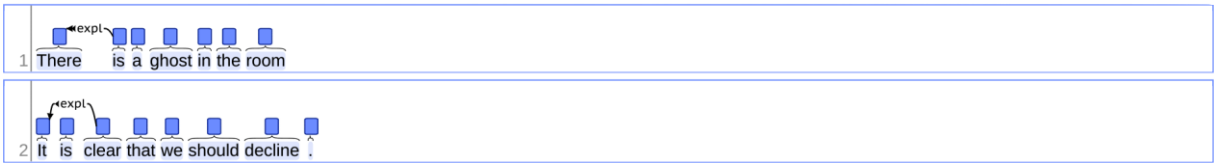

Fig. 2. Examples of EXPL noise links

The verbose expressions are annotated in an equal structure, where all subsequent words in the expression are attached to the first one using a permanent label. The assumption is that these expressions do not have an internal syntactic structure (except from a historical point of view) and the structural annotation is in

principle arbitrary. However, in practice, it is very important to use the consistent instruction of all constant verbose expressions in all languages [35].

Examples of FIXED links are shown in fig. 3

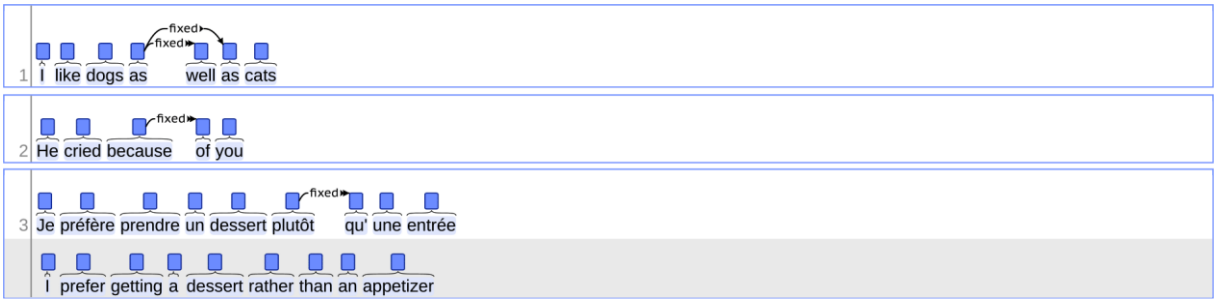

Fig. 3. Examples of FIXED noise links

PUNCT is used to denote any part of the punctuation in a sentence or part of the text if the punctuation is stored in typed dependencies.

PUNCT ratio tokens are always attached to the content of words and can never have dependencies. Since PUNCT is not a normal dependency relationship, the usual criteria for defining a headword are not applied. But the following principles are used:

1. A punctuation mark separating coordinated units is added to the following link.

2. A punctuation mark preceding or following an independent unit is attached to this unit.

3. Within the relevant division, the punctuation mark is attached to the highest possible node that maintains perspective.

4. Paired punctuation marks (for example, quotations and brackets, sometimes also hyphens, commas, etc.) should be attached to one word, if this does not violate the perspective [36].

Examples of PUNCT links are shown in Fig. 4.

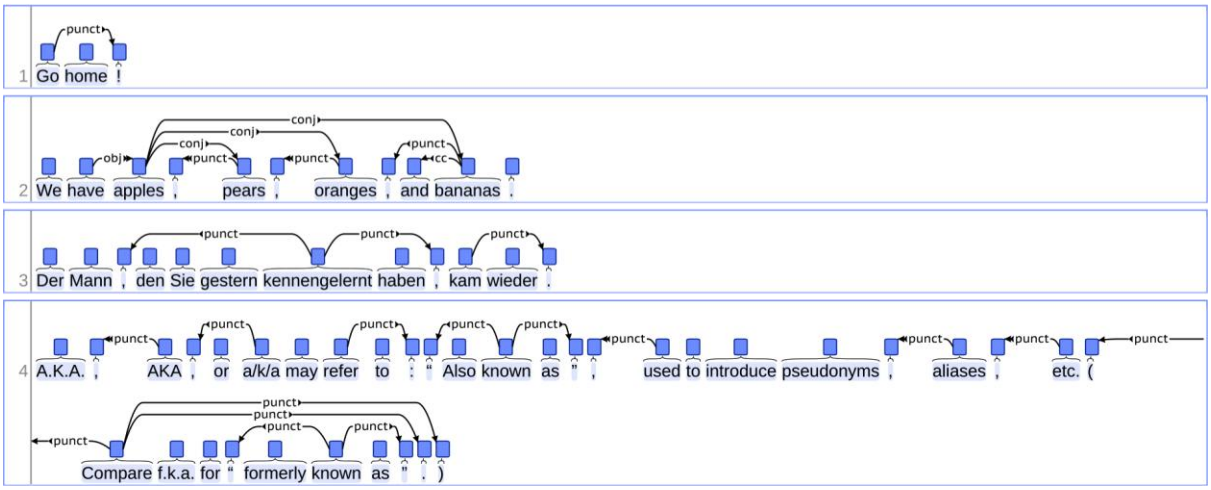

Fig. 4. Examples of PUNCT noise links

REF is referent of the main word to the noun phrase, which is a relative word that introduces a relative position by modifying the noun phrase. For example, for the sentence: “I saw the book which you bought”, the REF link will be between the words book and which [34].

ROOT is root grammatical relation, indicates the root of a sentence. The fake ROOT node is used as the main node. The ROOT node has an index of 0, since the indexation of real words in a sentence begins with 1. There should be only one root node in each tree. If the main predicate is absent, but there are many single dependencies, then one of them rises to the position of the main (root) one, and other singles join it [37].

An example of a ROOT link is shown in Fig. 5.

Fig. 5. Examples of ROOT noise links

Let’s consider deleting noise words related to non-informative parts of speech, have tags: CC, CD, DT, EX, IN, LS, MD, PDT, POS, PRP, PRP$, RP, SYM, TO, UH, WDT, WP, WP$, WRB, -LRB-, -RRB-.

CC – coordinating combinations: and, but, nor, or, yet, plus, minus, less, times (multiplication), over (division), also for (because), so (i. e., so that), &, 'n, both, either, et, neither, therefore, v., versus, vs., whether.

CD – number, count, quantity: one, two, 2, mid-1890, nine-thirty, forty-two, one-tenth, ten, million, 0.5, forty-seven, 1987, twenty, '79, zero, 78-degrees, eighty-four, IX, '60s, .025, fifteen, 271, 124, dozen, quintillion, DM2,000.

DT – determinant: a, an, every, no, the, another, any, some, all, both, del, each, either, half, la, many, much, nary, neither, such, that, them, these, this, those.

EX – existential there: unstressed there, which causes the inversion of the verb in the appropriate form and logical entity. For example: «There was a party in progress».

IN – prepositions or submission unions: among, around, astride, atop, behind, below, by, despite, for, if beside, if like, inside, into, near, next, on, out, pro, throughout, towards, until, upon, whether, within.

LS – list element, marker, numbers and letters that are used as identifiers of elements in the list: A, A., B, B., C, C., D, E, F, First, G, H, I, J, K, One, SP-44001, SP-44002, SP-44005, SP-44007, Second, Third, Three, Two, *, a, b, c, d, first, five, four, one, six, three, two.

MD – modal auxiliary verbs. All verbs do not accept the ending -s in the form of a third person singular: can, could, dare, may, might, must, ought, shall, should, will, would, cannot, couldn't, need, ought, shouldn't.

PDT – prefix determinant. Determinants, as elements, preceding clauses or possessive pronoun: all, both, half, many, quite, such, sure, this. For example: «all his marbles», «quite a mess».

POS – possessive ending of nouns ending in a marker ' or 's.

PRP – personal pronoun: he, her, hers, herself, him, him, himself, hisself, I, it, itself, me, myself, one, oneself, ours, ourselves, ownself, self, she, she, thee, theirs, them, themselves, they, thou, thy, us, you.

PRP$ – possessive pronoun: her, his, its, mine, my, one’s, our, ours, their, thy, your.

RP – share. Mostly monosyllabic words, also disyllabic, as adverbs: aboard, about, across, along, apart, around, aside, at, away, back, before, behind, by, crop, down, ever, fast, for, forth, from, go, high, i. e., in, into, just, later, low, more, off, on, open, out, over, per, pie, raising, start, teeth, that, through, under, unto, up, up-pp, upon, whole, with you.

SYM – symbol. Technical characters or expressions that are not words (% & ' " * + , . < = > @ A[fj] U.S U.S.S.R * ** ***).

TO – literal to, as a preposition or infinitive marker.

UH – interjection: amen, anyways, baby, dammit, diddle, Goodbye, Goody, Gosh, heck, Hey, honey, howdy, Hubba, huh, hush, Jee-sus, Jeepers, Kee-reist, man, my, oh, Oops, please, shucks, sonuvabitch, uh, well, whammo, whodunnit, Wow, yes.

WDT – wh- determinant: that, what, whatever, which, whichever.

WP – wh- pronoun: that, what, whatever, whatsoever, which, who, whom, whosoever.

WP$ – possessive wh-pronoun: whose.

WRB – wh-adverb, including *when*, when used figuratively: how, however, whence, whenever, where, whereby, wherever, wherein, whereof, why.

-LRB- – open bracket.

-RRB- – closed bracket [38–40].

On the removal of words belonging to the list of stop words - this question has already been investigated. The list of such words for English texts is justified and given in [41].

We illustrate the results of the definition of keywords at each step of the method proposed in a small text, consists of two sentences: “Obama is a lawyer at the Harvard Law Review. He was a community organizer in Chicago before earning his law degree.

The found phrases and parts of speech of the corresponding words of the first sentence are given in Table 1, and for the second – in the Table 2.

The types of connections between the main and dependent words in the phrases given in the unchanged, basic form of the word are given for the first and second sentences in the Tables 3 and 4.

Let's divide the phrases into separate words and count the number of links for each word, that is, in how many phrases the word occurs. Sorting the words by the number of links, let's obtain the results, which are listed in the Table 5.

Conventionally, the phrase can be designated:

G-[T]->D,

where G – Governor word T – Dependency Type; D – Dependent word.

Table 1

Phrases and parts of speech corresponding words of the first sentence

Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was president of the Harvard Law Review			
Governor word	Tag of part of speech of governor word (Governor POS)	Dependent word	Tag of part of speech of dependent word (Dependent POS)
graduate	NN	born	VBN
born	VBN	honolulu	NNP
honolulu	NNP	hawaii	NNP
graduate	NN	obama	NNP
graduate	NN	is	VBZ
graduate	NN	a	DT
university	NNP	columbia	NNP
graduate	NN	university	NNP
school	NNP	harvard	NNP
school	NNP	law	NNP
university	NNP	school	NNP
graduate	NN	school	NNP
president	NN	where	WRB

president	NN	he	PRP
president	NN	was	VBD
university	NNP	president	NN
review	NNP	the	DT
review	NNP	harvard	NNP
review	NNP	law	NNP
president	NN	review	NNP

Table 2

Phrases and parts of speech corresponding words of the second sentence

He was a community organizer in Chicago before earning his law degree			
Governor word	Tag of part of speech of governor word (Governor POS)	Dependent word	Tag of part of speech of dependent word (Dependent POS)
organizer	NN	he	PRP
organizer	NN	was	VBD
organizer	NN	a	DT
organizer	NN	community	NN
organizer	NN	chicago	NNP
organizer	NN	earning	VBG
degree	NN	his	PRP$
degree	NN	law	NN
earning	VBG	degree	NN

Table 3

Links in the phrases of the first sentence

Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was president of the Harvard Law Review					
Governor word	Dependent word	Dependency Type	Governor word	Dependent word	Dependency Type
graduate	bear	vmod	university	school	conj_and
bear	honolulu	prep_in	graduate	school	prep_of
honolulu	hawaius	appos	president	where	advmod

graduate	obama	nsubj	president	he	nsubj
graduate	be	cop	president	be	cop
graduate	a	det	university	president	rcmod
university	columbium	nn	review	the	det
graduate	university	prep_of	review	harvard	nn
school	harvard	nn	review	law	nn
school	law	nn	president	review	prep_of

Table 4

Links in the phrases of the second sentence

He was a community organizer in Chicago before earning his law degree		
Governor word	Dependent word	Dependency Type
organizer	he	nsubj
organizer	be	cop
organizer	a	det
organizer	community	nn
organizer	chicago	prep_in
organizer	earn	prepc_before
degree	his	poss
degree	law	nn
earn	degree	dobj

Table 5

Candidate keywords after dividing phrases

Word	Number of links	Word	Number of links	Word	Number of links
graduate	6	degree	3	hawaius	1
organizer	6	a	2	community	1
president	5	honolulu	2	the	1
university	4	earn	2	his	1
school	4	bear	2	columbium	1
review	4	harvard	2	where	1
be	3	he	2	chicago	1

law	3	obama	1	–	–

At the stage of replacing pronouns with their corresponding nouns (replace pronouns):

- the phrase president- [nsubj] -> he is replaced by president- [nsubj] -> obama;

- the phrase organizer- [nsubj] -> he is replaced by organizer- [nsubj] -> obama;

- the phrase degree- [poss] -> his is replaced by degree- [poss] -> obama.

Candidate keywords, after replacing pronouns with their corresponding nouns, are listed in Table 6

Table 6

Candidate keywords after pronoun substitutions

Word	Number of links	Word	Number of links	Word	Number of links
graduate	6	be	3	harvard	2
organizer	6	law	3	hawaius	1
president	5	degree	3	community	1
university	4	a	2	the	1
obama	4	honolulu	2	columbium	1
school	4	earn	2	where	1
review	4	bear	2	chicago	1

After the replacement of pronouns, the number of candidates for keywords decreased from 23 to 21. Before the replacement of pronouns, the word obama is not enough 1 link, and after – 4 links. Conversely, the words he with 2 bonds and his with a friend after the replacement of pronouns have zero connections, because the phrase with them has been replaced with equivalents with nouns.

Deleting phrases with types of connections that do not carry a significant semantic load (deleting noise relationship). For this text, phrases are deleted: graduate- [det] -> a, review- [det] -> the, organizer- [det] -> a.

As a result, the number of candidates for keywords will decrease to 19, which is reflected in Table 7

Table 7

Candidate keywords after deleting noise links

Word	Number of links	Word	Number of links
graduate	5	honolulu	2
organizer	5	earn	2
president	5	bear	2
university	4	harvard	2
obama	4	hawaius	1
school	4	community	1
be	3	columbium	1
law	3	where	1
degree	3	chicago	1
review	3	–	–

With drawal of words related to noise parts of speech (deleting noise POS keywords). At this point, the word where is deleted with the WRB part speech tag. Candidate keywords will have the form given in Table 8.

Table 8

Candidate keywords after deleting noise parts of speech

Word	Number of links	Word	Number of links	Word	Number of links
graduate	5	be	3	bear	2
organizer	5	law	3	harvard	2
president	5	degree	3	hawaius	1
university	4	review	3	community	1
obama	4	honolulu	2	columbium	1
school	4	earn	2	chicago	1

At the stage of deleting stop words –the stop word is removed and Table 9 contains 17 candidate keywords.

Table 9

Candidate keywords after deleting stop words

Word	Number of links	Word	Number of links	Word	Number of links
graduate	5	law	3	harvard	2
organizer	5	degree	3	hawaius	1
president	5	review	3	community	1
university	4	honolulu	2	columbium	1
obama	4	earn	2	chicago	1
school	4	bear	2	–	–

As a result, after all the proposed steps of the method, it is possible to reduce the number of candidates for keywords from 23 to 17, and also to remove noise words.

Let’s now consider a relatively large text in order to determine the quantitative characteristics of the relevance of the results obtained in comparison with analogues. For this, the text “A Workingman's Poet” was chosen, which consists of 3299 words, and the keywords specified by the author are known: american, literature, literature, chicago, poetry, publishing, twentieth century, united states. According to the results of the experiment there are the first ten candidates for keywords found by the developed method: sandburg, poem, write, poet, poetry, book, life, lincoln, learn, speak. The search for keywords in the same text was implemented using similar programs.

The results of finding the keywords developed by the method and analogues are given in Table 10.

The results of the completeness and accuracy of the obtained keywords are given in Tables 11 and 12 and in Fig. 6 and 7.

Table 10

The results of finding keywords by developed method and analogues

Etalon keywords		Advego		Rise-top		Seotool		Our development	
1	American	–	sandburg	–	sandburg	–	his	–	sandburg
2	Literature	–	that	–	his	–	sandburg	–	poem
3	Books	–	for	–	lincoln	–	lincoln	–	write
4	Chicago	–	poem	5	poetry	–	poems	–	poet
5	Poetry	–	lincoln	–	poems	5	poetry	5	poetry
6	Publishing	5	poetry	–	who	–	who	3	book
7	Twentieth	–	work	1	american	1	american	–	life
8	Century	–	write	–	where	–	where	–	lincoln
9	United	1	american	–	had	–	years	–	learn
10	States	–	where	–	years	–	had	–	speak

Table 11

Keyword completeness results

Name	Advego	Rise-top	Seotool	Our development
Completeness (Jaccard)	0,111111111	0,111111111	0,111111111	0,111111111
Completeness (Absolute)	0,2	0,2	0,2	0,2

Table 12

Keyword accuracy results

Name	Advego	Rise-top	Seotool	Our development
Euclidean distance	0,577061522	0,59749477	0,589067059	0,577061522
Manhattan distance	0,49	0,51	0,49	0,47

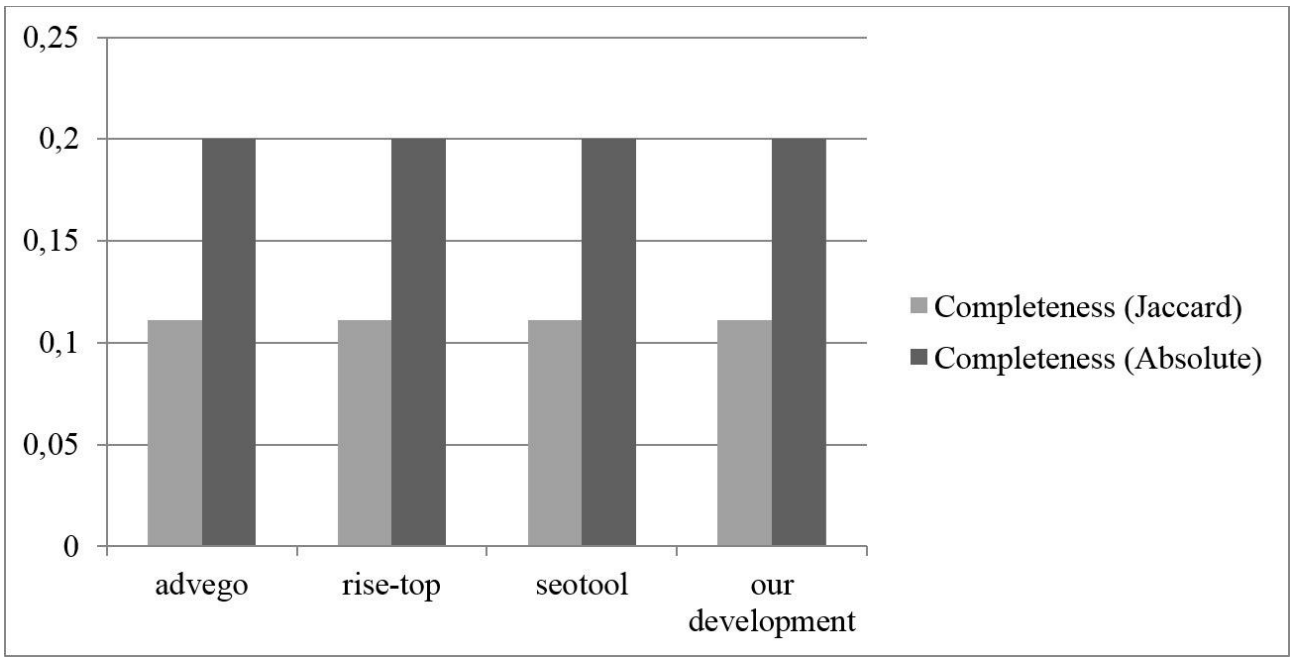

Fig. 6. Histograms of completeness by Jacquard and absolute

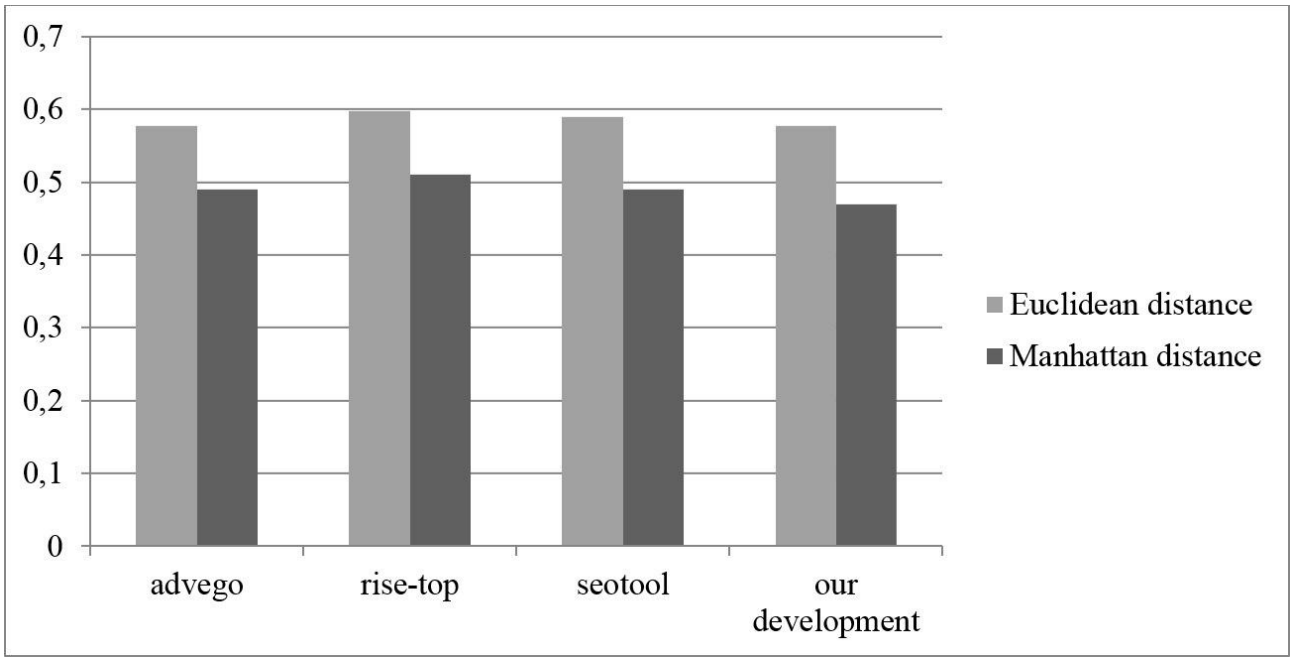

Fig. 7. Histograms of accuracy by Euclidean and Manhattan distances

The completeness of finding the keywords should be as large as possible, and the distance between the positions of the keywords given by the author and certain programmatically possible is less.

As can be seen from the histograms in Fig. 6, 7 and Tables 11, 12, own development for this text has the same completeness as analogs – 11% and 20%,

however, the best quantitative characteristics in terms of accuracy are 57.71% and 47% than analogs of rise-top (59.75 %; 51%); and seotool (58.91%; 49%). Also, its own development has the same accuracy for the Euclidean distance, as well as the analogue advego, but, in contrast, the best characteristics for the Manhattan distance.

5. SWOT analysis of the method results

Strengths. Compared to analogs, the development is presented, according to the results of the experiment conducted with the text of 3299 words, has the same completeness as the analogs, however, the best quantitative characteristics in terms of accuracy than analogs of rise-top and seotool. Also presented the development has the same accuracy for the Euclidean distance, as well as analogue advego, but, in contrast, the best characteristics for the Manhattan distance. Another advantage compared with analogues is that the presented technology allows to completely eliminate noise words.

Weaknesses. The weaknesses of the method include the speed of its practical implementation by means of DKPro Core, in particular, it is relatively long for the online mode to create a multilevel text markup. But this, in turn, can be corrected through the use of more powerful hardware or cloud computing platforms, allowing you to have a virtual cluster of computers. This is not difficult to achieve, because applications for defining keywords and reducing verbal noise are written in Java and can be easily deployed on such platforms.

Opportunities. The opportunity of further research on the definition of keywords is conducting larger-scale experiments for texts of various categories in order to determine additional ways to increase the relevance of the method. It is also advisable to use new linguistic packages that support more languages, including Ukrainian.

Threats. The process of defining keywords by the proposed method is independent of the processes of defining keywords by other methods, therefore there is no threat of a negative impact on the object of study of external factors.

The implementation of the proposed methodology does not require additional costs for the company.

An analogue of the developed method can be SEO optimization sites with the ability to determine keywords.

6. Conclusions

1. A method is proposed for filtering verbal noise in which it is provided by such formalized operations:

- replacing pronouns with their corresponding nouns;
- removal of noise connections;
- removal of noise words;
- withdrawal of stop words.

The described operations can be used as additional modules that improve the results of finding keywords for the method of determining keywords of English text based on the tools of the DKPro Core packages and also for other algorithms for finding keywords.

2. The calculation of the numerical indicators of the connections between words and the analysis of the results obtained at each stage of the method proposed is illustrated by the example of a text from two sentences. According to the results considered in the example, it is possible to reduce the number of candidates for keywords from 23 to 17, and also to completely eliminate noise words.

3. According to the results of the experiment, a development for text with 3299 words is presented, which has the same completeness as the analogs – 11% and 20%, however, the best quantitative characteristics in terms of accuracy are 57.71% and 47%, than the analogues rise-top (59.75%; 51%) and seotool (58.91%; 49%). The

presented development also has the same accuracy for the Euclidean distance, as well as the analogue advego, but, in contrast, the best characteristics for the Manhattan distance.

7. Questions for self-control and improvement

The aim of study is improvement of the accuracy of determining keywords from English text based on the development of a method for reducing the influence of verbal noise.

To achieve this aim it is necessary to answer the following questions:

1. What approaches to a reduction of verbal noise when finding keywords do you know?
2. How to calculate the numerical indicators of the connections between words?
3. How to analyze the main results obtained as the basis of the method?
4. How to formalize the operations for each stage of the method?
5. How to determine the quantitative characteristics of the relevance of the results obtained in comparison to analogues?
6. What do you know about free online services that can automatically generate the keys for the specified text as analogues for the method?
7. What does verbal noise consist of?
8. What is the main idea of the TF-IDF algorithm?
9. How to calculate the weights of terms based on the TF-IDF algorithm?
10. How to calculate Euclidean and Manhattan distances?
11. What operations are proposed to be use for verbal noise filtering in the method?
12. How many types of connections between the main and dependent words in the phrases do you know?

References

1. Oleh Bisikalo, Alexander Yahimovich. Keyword search based on lexical relationships in the text. – Mauritius : Lap Lambert Academic Publishing, 2019. – 57 p. – ISBN 978-620-0-00314-0.

2. Oleg V. Bisikalo ; Waldemar Wójcik ; Olexand V. Yahimovich ; Saule Smailova "Method of determining of keywords in English texts based on the DKPro Core", *Proc. SPIE* 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016, 100314T (September 28, 2016); doi:10.1117/12.2249225.

3. Bisikalo O. Development of the method for filtering verbal noise while search keywords for the English text / O. Bisikalo, A. Yahimovich, Y. Yahimovich // Technology audit and production reserves: Information and control systems. – 2018. – Vol. 2, № 6(44). – P. 33–41. – ISSN 2226-3780.

4. Ershov Yu. S. Vydelenie klyuchevykh slov v russkoyazychnykh tekstakh // Molodezhnyy nauchno-tekhnicheskiy vestnik. – 2014. Issue FS77-51038. P. 70–79.

5. Bisikalo O. V. Conceptual model of pattern analysis and synthesis of natural language constructions / O. V. Bisikalo // Mathematical Machines and Systems. 2013. № 2. P 184–187. ISSN 1028-9763.

6. Bisikalo, O. V. Formal methods of image analysis and synthesis of natural language constructions / O. V. Bisikalo Vinnytsia: VNTU, 2013. – 316 p. ISBN 978-966-641-528-1.

7. Natural Language Processing: Integration of Automatic and Manual Analysis. (2014). Technischen Universität Darmstadt. Available: http://tuprints.ulb.tu-darmstadt.de/4151/1/rec-thesis-final.pdf. Last accessed 06.07.2015.

8. Gurevych, I., Muhlhauser, M., Muller, Ch., Steimle, J., Weimer, M., Zesch, T. (2007, February 9). Darmstadt Knowledge Processing Repository Based on UIMA. Available:

https://www.ukp.tu-
darmstadt.de/fileadmin/user_upload/Group_UKP/publikationen/2007/gldv-uima-ukp.pdf. . Last accessed 06.07.2015.

9. K. Bougé. “Lists of stop words.” [Electronic resource]. – Access mode: \www/URL: https://sites.google.com/site/ kevinbouge/ stopwords-lists.

10. Word level. Bracketing Guidelines for Treebank II Style Penn Treebank Project. [Electronic resource]. – Access mode: \www/URL: http://www.surdeanu.info/mi-hai/teaching/ista555-fall13/readings/PennTreebankConstitu-ents.html.

11. Geoghegan, H. (2014). A new pattern for historical geography: working with enthusiast communities and public history. Journal of Historical Geography, № 46. Available:
http://ac.els-cdn.com/S0305748814001029/1-s2.0-S0305748814001029-main.pdf?_tid=d45ec9e6-ba7b-11e4-b56200000aab0f01&acdnat=1424600353_48bb4ef54fffbc- 3b800698d175c3c052. Last accessed 06.07.2015.

12. The Hound of the Baskervilles: Plot Keywords. [Electronic resource]. – Access mode: \www/URL:: http://www.imdb.com/title/tt0052905/keywords.

13. Grashhenko L. A. O model’nom stop-slovare // Izvestiya Akademii nauk Respubliki Tadzhikistan. Otdelenie fiziko-matematicheskikh, khimicheskikh, geologicheskikh i tekhnicheskikh nauk. 2013. Issue 1 (150). P. 40–46.

14. Modeli i metody avtomaticheskoy klassifikatsii tekstovykh dokumentov / Andreev A. M. et. al. // Vestn. MGTU. Seriia Priborostroenie. 2003. Issue 3. P. 64–94.

15. Abramov E. G. Podbor klyuchevykh slov dlya nauchnoy stat’i // Nauchnaya periodika: problemy i resheniya. 2011. Issue 1 (2). P. 35–40.

16. Darkulova K. N., Ergeshova G. Neobkhodimost’ vydeleniya klyuchevykh slov dlya svertyvaniya teksta: Proceedings // Lingvisticheskiy analiz nauchnogo teksta. Yuzhno-Kazakhstanskiy gosudarstvennyy universitet im. Mukhtara Auezova Shymkent, 2014. P. 30–35.

17. Halkidi M., Batistakis Y., Vazirgiannis M. On clustering validation techniques // Journal of intelligent information systems. 2001. Vol. 17, Issue 2-3. P. 107–145. doi: http://doi.org/10.1023/a:1012801612483.

18. Barahnin V. B., Tkachev D. A. Clustering of text documents based on composite key terms // Vestnik NSU. Series: Information Technology. 2010. Vol. 8, Issue 2. P. 5–14.

19. Grashhenko L. A. O model’nom stop-slovare // Izvestiya Akademii nauk Respubliki Tadzhikistan. Otdelenie fiziko-matematicheskikh, khimicheskikh, geologicheskikh i tekhnicheskikh nauk. 2013. Issue 1 (150). P. 40–46.

20. Guo A., Tao Y. Research and Improvement of Feature Words Weight Based on TFIDF Algorithm // 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference. Chongqing, 2016. doi: http://doi.org/10.1109/itnec.2016.7560393.

21. Sifting Micro-blogging Stream for Events of User Interest / Grineva M. et. al. // Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. Boston, 2009. P. 327–333. doi: http://doi.org/10.1145/1571941.1572157.

22. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams / Reed J. et. al. // 2006 5th International Conference on Machine Learning and Applications. Orlando, 2006. P. 258–263. doi: http://doi.org/10.1109/icmla.2006.50.

23. Mihalcea R., Csomai A. Wikify!: linking documents to encyclopedic knowledge // Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. Lisbon, 2007. P. 233–242. doi: http://doi.org/10.1145/1321440.1321475.

24. Astrakhantsev N. Automatic term acquisition from domain-specific text collection by using Wikipedia // Proceedings of the Institute for System Programming of RAS. 2014. Vol. 26, Issue 4. P. 7–20. doi: http://doi.org/10.15514/ispras-2014-26(4)-1.

25. Özgür, Arzucan, Junguk Hur, and Yongqun He. The Interaction Network Ontology-supported modeling and mining of complex interactions represented with multiple keywords in biomedical literature // BioData Mining. 2016. Vol. 9, Issue 1. doi: http://doi.org/10.1186/s13040-016-0118-0.

26. Wong W., Liu W., Bennamoun M. Ontology learning from text // ACM Computing Surveys. 2012. Vol. 44, Issue 4. P. 1–36. doi: http://doi.org/10.1145/2333112.2333115.

27. Korobkin D. M., Fomenkov S. A., Kolesnikov S. G. Method of ontology-based extraction of physical effect description // Vestnik Komp'iuternykh i Informatsionnykh Tekhnologii. 2015. P. 28–35. doi: http://doi.org/10.14489/vkit.2015.02.pp.028-035.

28. Besplatnyy onlayn-generator klyuchevykh slov s teksta. URL: http://seotool.by/analiz/seo/keywordstext.php.

29. Generator klyuchevykh slov s teksta. URL: http://www.rise-top.com.

30. Advego. URL: http://wiki.advego.ru/index.php/Адвего.

31. Determiner. URL: http://universaldependencies.org/u/dep/det.html.

32. Expletive and Reflexives. URL: http://universaldependencies.org/u/dep/expl.html.

33. Welo E. Null Anaphora // Encyclopedia of Ancient Greek Language and Linguistics. 2013. doi: http://doi.org/10.1163/2214-448x_eagll_com_00000254.

34. Manning C., de Marneffe M. Stanford typed dependencies manual. 2016. URL: https://nlp.stanford.edu/software/dependencies_manual.pdf.

35. Fixed multiword. URL: http://universaldependencies.org/u/dep/fixed.html.

36. Punctuation. URL: http://universaldependencies.org/u/dep/punct.html.

37. Root. URL: http://universaldependencies.org/u/dep/root.html.

38. Taylor A., Marcus M., Santorini B. The Penn Treebank: An Overview // Text, Speech and Language Technology. 2003. P. 5–22. doi: http://doi.org/10.1007/978-94-010-0201-1_1.

39. Penn Treebank II Constituent Tags: Word level. URL: http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html#Word.

40. Alphabetical list of part-of-speech tags used in the Penn Treebank Project. URL: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

41. Boug K. Lists of stop words. URL: https://sites.google.com/site/kevinbouge/stopwords-lists.