Nina Rizun, Yurii Taranenko

OPRACOWANIE ALGORYTMU WSTĘPNEGO PRZETWARZANIA TEKSTÓW RECENZJI FILMÓW W JĘZYKU POLSKIM

[słowa kluczowe: wstępne przetwarzanie; język polski, tokenizacja, lematyzacja, model wektorowy danych]

Streszczenie

Opracowano algorytm i oprogramowanie do przeprowadzania procedury wstępnego przetwarzania recenzji filmów w języku polskim. Algorytm zawiera następujące kroki: procedura adaptacji tekstu; procedura Tokenizacji; procedura przekształcania słów w format bajtów; tagowanie części mowy; procedura Stemmingu / lematyzacji; prezentacja dokumentów w formacie wektorowym (Vector Space Model); procedura tworzenia bazy danych modelów dokumentów. Przeprowadzono eksperymenty z zaproponowanym algorytmem na próbce testowej analizy recenzji filmów i sformułowano główne wnioski.

* * *

DEVELOPMENT OF THE ALGORITHM OF POLISH LANGUAGE FILM REVIEWS PREPROCESSING

[keywords: Preprocessing; Polish language; Tokenization; Lemmatization; Vector Space Model]

Abstract

The algorithm and the software for conducting the procedure of Preprocessing of the reviews of films in the Polish language were developed. This algorithm contains the following steps: Text Adaptation Procedure; Procedure of Tokeni-

zation; Procedure of Transforming Words into the Byte Format; Part-of-Speech Tagging; Stemming / Lemmatization Procedure; Presentation of Documents in the Vector Form (Vector Space Model) Procedure; Forming the Documents Models Database Procedure. The experiments of this algorithm conduction on the test sampling of reviews analysis was performed and the main conclusion was formulated.

Introduction

The volume of circulating in the world's telecommunications networks and the information stored on servers demonstrate the dynamics of explosive growth. As of Cisco Systems estimates that from 2010 to 2015, the monthly volume of transferred Internet traffic, including text and web data, increased from 2.4 to 8.6 exabytes. And by 2018, projected to double this number. Proportionally grow market indexes text analytics, which capacity according to International Data Corporation (IDC.com) in 2015 was $ 2.65 billion, and the forecast for 2020 – $ 5.9 billion. This is being analyzed less than 1% of the texts, and the market growth is mainly due to the analysis of social networking data [1].

All of the above leads to an increase in the composition and complexity of software solutions in the field of processing of texts in natural languages, which are based on a number of basic algorithms, including – text preprocessing algorithms for further analysis. From the standpoint of this article, first of all, we are talking about the Polish language has a number of features in relation to English, which was developed for the bulk of publicly available algorithms. Often, the authors claim on the high performance of their preprocessing algorithms, without giving any data on their use in certain products, or for an arbitrary set of foreign language texts trials [2-4].

The **objective** of this paper is in developing the Algorithm of Preprocessing the Films' Reviews context with taking account the in Polish language specifics, which is based on a combination of *linguistic* and *statistical* analysis and is intended to form the most *statistically significant* and *linguistically qualitative* model of the documents' corpus.

This work described in the paper was supported by the Polish National Centre for Research and Development (NCBiR) under Grant No. *PBS3/B3/35/2015*, project „*Structuring and classification of Internet contents with prediction of its dynamics*" (Polish title: „Strukturyzacja i klasyfikacja treści internetowych wraz z predykcją ich dynamiki").

Theoretical Background of the Research

It is well-known that the process of textual information analysis can be as well presented as levels (Fig. 1).

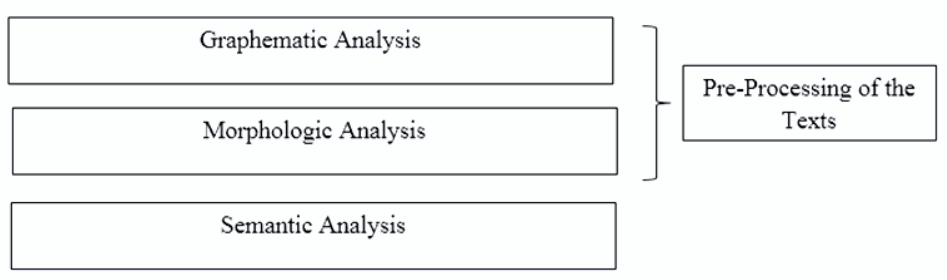

Figure 1. Process of Text Information Analysis

The complexity of text analysis increases with the growth of language level. Analysis at the upper level is impossible the analysis at previous levels, conducted before.

Graphematic Analysis

To start the morphologic analysis of a text it is necessary to divide the original unstructured text into sentences and words. At first sight, it is a very simple task, but it has its own specificities and plays an important role in the further analysis of a text.

Graphematic analysis includes:

- division of the original text into elements (words, separators);
- elimination of non-text elements (tags, meta-information);
- extraction and formalization of non-standard elements: structural elements: headlines, paragraphs, notes; numbers, dates, complexes of letters and numbers; names, patronymics, surnames; extraction of e-mail addresses, files' names;
- extraction of sustained phrases, words that are not used separately from each other.

In English sources we can meet the definition *tokenization*, which, by its content, is similar to the graphematic analysis. *Tokenization* – is a process of dividing the text stream into tokens: words, collocations and sentences [5].

Thus, the graphematic analysis is the initial analysis of an unstructured text, presented as a chain of symbols in any coding, elaborating information, which is necessary for further text processing.

There are almost no tools specializing exceptionally on graphematic analysis. Basically, graphematiic is included into integrated packages of text analysis: *NLTK*, *Stanford CoreNLP*, *Apache NLP, AOT, MBSP* etc. The function of division into tokens is also included into programs of text markup, for instance into the part-of-speech taggers.

In most cases the task of division can be solved in a trivial way: using a dictionary of separators and the dictionary of sustained phrases. Besides, the task can be solved with the help of regular phrases (Table 1).

Table 1. Tools for tokenization, working with the Polish language

Name	**Method**	**Languages**	**Platform**
Toki	Rules of tokenization	European languages, especially at Polish (the default configuration is for Polish)	C++
OmegaT	Lucene implementation of the Hunspell algorithm	European languages, especially at Polish	RE (Java Runtime Environment)
parser.rb	Polish notation for calcs	Polish	Ruby
Wordfreq	Rules of tokenization	Chinese, English, Greek, Polish, Swedish, and Turkish	Python
TokenizerPL.java	Rules of tokenization	Polish	Java

(Source: Personal research)

Morphologic Analysis

Morphologic analysis provides definition of the normal form, from which the word-form was created, and of the set of parameters, assigned to this word-form [6].

Stemming has been the most widely applied morphological technique for information retrieval. With stemming, the searcher does not need to worry about the correct truncation point of search keys. Stemming also reduces the total num-

ber of distinct index entries. With short queries and short documents, a derivational stemmer is most useful, but with longer ones the derivational stemmer brings in more non-relevant documents. Stemming increases search key ambiguity. Stemming may, however, is a non-optimal approach to the clustering of documents in agglutinative languages. Firstly, stemmers do not conflate compounds whenever the first components do not match exactly. Secondly, they are unable to split compounds, which typically have the head-modifier structure and the headword is the last and more important component for clustering [7]. The most widely-spread algorithm of stemming is the *Porter's algorithm*. Except for that algorithm there exists the *Lancaster's algorithm* (for English language) and the algorithms, working by the principle of a *"snowball"* (*snowball stemmers*) for other languages.

Lemmatization is another normalization technique: for each inflected word form in a document or request, its basic form, the lemma, is identified. The benefits of lemmatization are the same as in stemming. In addition, when basic word forms are used, the searcher may match an exact search key to an exact index key. Such accuracy is not possible with truncated, ambiguous stems. Homographic word forms cause ambiguity (and precision) problems – this may also occur with inflectional word forms [8, 9]. Another problem is owing to words that cannot be lemmatized, because the lemmatizer's dictionary does not contain them (Table 2).

Table 2. Tools of analysis for texts in Polish language

Name	**Method**	**Language**
Stempel	Algorithmic Stemmer	Polish
LAMETYZATOR	Dictionary-based stemmer	Polish
SAM-96	Morphological analyzer	Polish
Stempel	Heuristic stemmer	Polish
WASPELL	Dictionary-based stemmer	Polish
STEMPELATOR	Hybrid Stemmer	Polish
MORFEUSZ	Dictionary-based stemmer	Polish
LemmaPL	Lemmatization tool	Polish
LemmaGen	Lemmatization tool	for 11 EU languages
Morfologik	Morphological analyser	Polish
SAM	Morphological analyser	Polish

Although a report published over 10 years ago, by Hajnicz and Kups´c (2001),

(Source: Personal research)

Although a report published over 10 years ago, by Hajnicz and Kups´c (2001), already mentions 12 morphological dictionaries or analyzers for Polish, most of them are not publicly available or are not free even for non-commercial scientific purposes. Until recently only a few such resources of a reasonable size and quality were freely available for research, most notably:

- *UAM Text Tools* (http://utt.amu.edu.pl/; Zygmunt Vetulani and Tomasz Obrębski. Morphological tagging of texts using the lemmatizer of the 'POLEX' electronic dictionary. In [10, 11] with the underlying dictionaries now licensed under both Creative Commons (CC) Attribution Non-Commercial Share Alike (by-nc-sa) and GNU General Public License (GPL));
- *Morfeusz*, until recently free for non-commercial use, but not open source;
- *Morfologik*, until recently available on GNU Lesser General Public Licence (LGPL).

Morfologik is probably the first truly open source morphological dictionary of Polish. It is accompanied with an analyzer library, Morfologik-stemming. It contains 216,992 lexemes and 3,475,809 word forms. The dictionary was created by enriching the Polish ispell/hunspell dictionary with morphological information, which was possible thanks to the structure of the original dictionary that retained important grammatical distinctions [12, 13] The process of conversion relied on a series of scripts, and the resulting dictionary was later augmented with manually entered information. Unfortunately, the original source dictionary did not contain sufficient structure to allow reliable detection of some information, such as the exact subgender of the masculine for substantives. This information was added manually and using heuristic methods, however its reliability is low. Considering the fact that the substantives are about one third of the dictionary content (and almost half of them are masculine), this limitation is severe. The tagset of the dictionary is inspired by the IPI PAN Tagset [14].

However, Morfologik diverges from that tagset and from Morfeusz, as it never splits orthographic ("space-to-space") words into smaller dictionary words (i.e., so-called agglutination is not considered). Moreover, due to the lack of information in the ispell dictionary, some forms are not completely annotated, and are marked as irregular. There is, however, some additional markup added to reflexive verbs, which is not present in the original IPI PAN Tagset. This was introduced for the purposes of the grammar checker Language Tool that used the dictionary extensively. In contrast to SGJP, Morfologik was closely linked with a variant of the IPI PAN Tagset and adoption of a radically different tagset was not practical because of the flat textual representation of morphological data.

The tagset of Morfologik, on the contrary, is hardly defined. The Readme file gives the mnemonics of the grammatical classes and the attribute values. Attributes as such are not explicitly enumerated, and, in some cases it is hard to infer which attribute some values belong to. Not all the actual classes are documented. The positionality is not respected and in the actual data the forms of the same grammatical class are likely to occur with quite a number of combinations of attributes whose values are specified. In one case the distinction between an attribute and a class is blurred (refl is declared as a value, although, technically, it occupies a class position in the dictionary). Despite this inexact frame of Morfologik tagset, the actual tags closely resemble those of the IPIC tagset. This is intended and some additional remarks on the differences are given in the Readme file. To obtain a sketch of the real tagset (i.e. classes that describe the actual data) we developed a Python script that reads a morphological dictionary and outputs a list of value usage patterns. Each of the patterns is a subclass of one grammatical class that has a fixed number of values provided. A pattern is described by sets of values that appeared at subsequent positions (the script naively assumes that the tagset is positional).

The first line states that there are occurrences of the adj class with no attributes. The second one presents a pattern corresponding to three-value adj tags, whose first attribute can be recognized as grammatical number, the second attribute as case and third – gender. A comparison of the grammatical classes appearing in both tagsets is presented in Table 3.

Table 3. A comparison of IPIC and Morfologik grammatical classes

IPIC	***Morfologic***	***Name (IPIC)***	***Example form***
adj	*adj*	*adjective*	biały
adja	*missing*	*ad- adjectival adj.*	biało (-czerwony)
adjp	*adjp*	*post-prep. adj.*	(po) polsku
adv	*adv*	*adverb*	biało
agit	*segmentation*	*agglut. być*	czytał(em)
bedzie	*verb: bedzie*	*future być*	będziesz
conj	*conj*	*conjunction*	lub
depr	*subst: depr*	*deprec. noun*	posły
fin	*verb: fin*	*non-past form*	czyta
ger	*subst: ger*	*gerund*	picie
ign (unknown)	*ing (unreliable)*	*unknown*	xyz123
imps	*verb: imps*	*impersonal form*	czytano
impt	*verb: impt*	*imperative*	czytaj
inf	*verb: inf*	*infinitive*	czytać

ciąg dalszy table 3.

IPIC	***Morfologic***	***Name (IPIC)***	***Example form***
num	*num*	*numeral*	sześć
numcol or num (rare)	*num*	*collective num.*	sześćioro
pcon. pant	*pcon. pant*	*adv. participle*	pijąc, wypiewszy
pcon. ppas	*pcon. ppas*	*adv. participle*	pijąc, pity
ppron12, ppron3	*ppron12, ppron3*	*personal pronoun*	ciebie, oni
praet *praet+aglt* *prate+by* *prate+by+aglt*	*verb: prate* *verb: prate* *verb: prate: pot* *verb: prate: pot*	*l-participle (past form)* *(conjunctive)* *(conjunctive)*	czytał czytałem czytałby czytałbym
pred	*pred*	*predicative*	widać
pred	*pred*	*predicative*	na
qub	*qub (unrealible)*	*particle-adverb*	się, nawet
siebie	*siebie*	*pronoun siebie*	sobą
subst	*subst*	*noun*	mięso
winien	*winien*	*winien-like verb*	powinni

Source: Developing free morphological data for Polish, Adam Radziszewski, Marek Maziarz

Algorithm of Preprocessing the Films' Reviews

Realization of the following stages is put into the basis of the suggested algorithm (Figure 2)

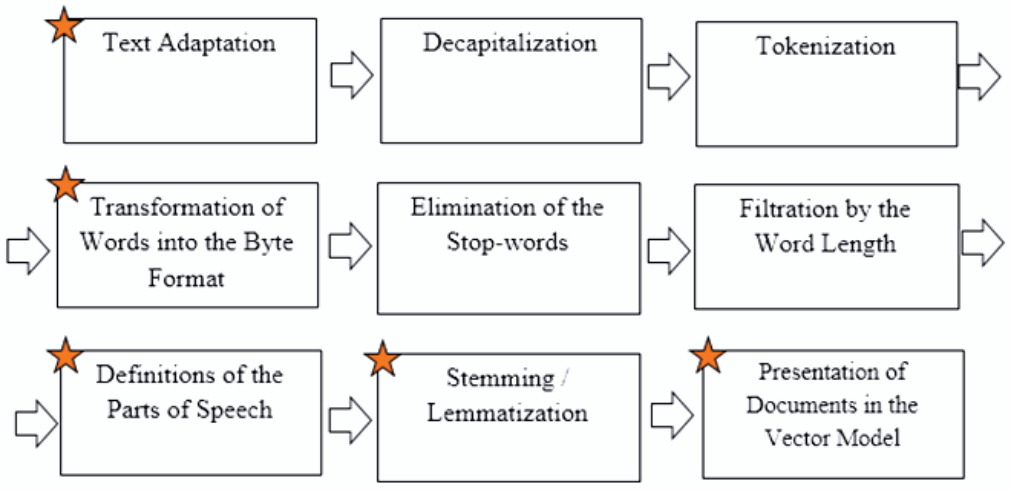

Figure 2. Stages of the Preprocessing the Films' Reviews Algorithm

The symbol ☆ in this scheme marks the stages that present the results of improvement (introduced by the authors) of the standard procedure of algorithm of text Preprocessing before conduction of the semantic analysis and cauterization.

Text reviews with the following structure of document layout were used as the examined material (Figure 3):

According to the figure 3, the suggested algorithm presupposes interpretation of 9 main steps:

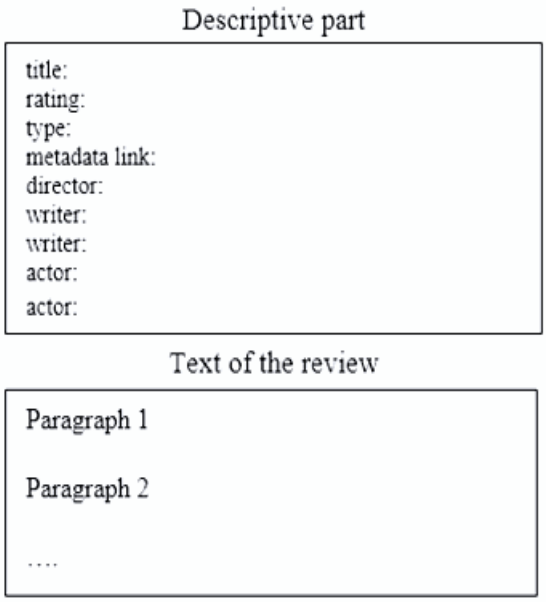

Figure 3. Structure of Text Reviews Markup

Text adaptation procedure

Presupposes usage of the *Descriptive part of the review* for replacement of the *Film's name* and *the names of creators/actors* of the film into the corresponding positions in the reviewed film (for example, the title of the film is replaced by the word "Film", surname of the actor – by the word "Actor" etc.).

To illustrate the results of the experiments on Preprocessing stages realization we will use the following fragment of a review:

> **„Podwójne zagrożenie"** jest właśnie typowym przykładem tej mechanicznej zamiany miejsc. **Libby**, równie dobrze mogłaby być mężczyzną. Jej przygody rozwijają się według mało oryginalnego, znakomicie znanego z męskich filmów tego rodzaju, schematu – banalna kryminalna intryga, niesprawiedliwy wyrok, odsiadka, ucieczka i wielki pościg za prawdziwym winowajcą. Dodano co prawda wątek macierzyńskich cierpień **Libby**, która tęskni za synkiem i pragnie go odnaleźć

Then, the example of the result of conducting the stage of the text adaptation is reflected in the following fragment:

Film jest właśnie typowym przykładem tej mechanicznej zamiany miejsc. **bohater**, równie dobrze mogłaby być mężczyzną. Jej przygody rozwijają się według mało oryginalnego, znakomicie znanego z męskich filmów tego rodzaju, schematu – banalna kryminalna intryga, niesprawiedliwy wyrok, odsiadka, ucieczka i wielki pościg za prawdziwym winowajcą. Dodano co prawda wątek macierzyńskich cierpień **bohater**, która tęskni za synkiem i pragnie go odnaleźć.

Procedure of decapitalization

It presupposes transformation of all the symbols in the documents' collection into the lower case. The example of the results of this stage are as follows:

film jest właśnie typowym przykładem tej mechanicznej zamiany miejsc. **bohater**, równie dobrze mogłaby być mężczyzną. jej przygody rozwijają się według mało oryginalnego, znakomicie znanego z męskich filmów tego rodzaju, schematu – banalna kryminalna intryga, niesprawiedliwy wyrok, odsiadka, ucieczka i wielki pościg za prawdziwym winowajcą. dodano co prawda wątek macierzyńskich cierpień **bohater**, która tęskni za synkiem i pragnie go odnaleźć

Procedure of tokenization

Presupposes definition of the words' borders with the help of a number of stop-signs of the word: pace and punctuation marks are the separators of words-tokens; chains of letters with numbers are understood as one token; double proper names are understood as one token; date and time are understood as one token; special abbreviations (for example, P.S.) are understood as one token; fractional numbers are understood as one token; proper names, including initials.

In this algorithm classical rules of tokenization and function *split()* of Python are used.

As a result, for every document D_j in text collection τ we receive a list of tokens $\phi_t = \left(t\,t_1^{D_j}, t\,t_2^{D_j}, .., t\,t_R^{D_j}\right)$, where R_j – is a total number of words in the document D_j, and $m = \sum_{j=1}^{n} R_j$ – is a total number of words in the reviews collection τ.

```
φt = [ , film′ , , jest′ , , właśnie′ , , typowym′ , , przykładem′ ,
, tej′ , , mechanicznej′ , , zamiany′ , , miejsc′ , , bohater′ ,
, równie′ , , dobrze′ , , mogłaby′ , , być′ , , mężczyzną′ ,
, jej′ , , przygody′ , , rozwijają′ , , się′ , , według′ , , mało′ ,
, oryginalnego′ , , znakomicie′ , , znanego′ , , z′ , , męskich′ ,
, filmów′ , , tego′ , , rodzaju′ , , schematu′ , , banalna′ ,
, kryminalna′ , , intryga′ , , niesprawiedliwy′ , , wyrok′ ,
, odsiadka′ , , ucieczka′ , , i′ , , wielki′ , , pościg′ , , za′ ,
, prawdziwym′ , , winowajcą′ , , dodano′ , , co′ , , prawda′ ,
, wątek′ , , macierzyńskich′ , , cierpień′ , , bohater′ , , która′ ,
, tęskni′ , , za′ , , synkiem′ , , pragnie′ , , go′ , , odnaleźć′ ]
```

Procedure of transforming words into the byte format

It is **added** (by the authors) into the standard algorithm of Preprocessing in connection with the necessity of solving the problem of reading and displaying words in Polish language. For this purpose, the procedure, which conducts a character-by-character transformation of words into a byte format, was introduced into the text Preprocessing program:

```
q = []
    for w in b:
      word=""
      for k in w:
        if k.isalpha():
          word=word+k
      q.append(word)
```

Elimination of the stop-words procedure

Stop-words are the words, which are met in the language so frequently that their information value is almost equal to zero – in other words, their entropy is very low. Besides, the words, which have fewer than 2 symbols, are eliminated.

As a result of performing this operation we get the multitude $\phi_t^{D_j} = \left(t_1^{D_j}, t_2^{D_j}, .., t_{RU}^{D_j}\right)$, where RU – is the number of terms in the document D_j, which are left after the elimination of stop-words and those that have fewer than 2 symbols.

Part-of-speech tagging procedure

It is **added** (by the authors) with the objective to increase the flexibility of the semantic analysis process, which allows to examine and increase the efficiency of clusterization and contextual analysis of texts by using different sets of parts of speech.

As it was already stated before, for the Preprocessing the dictionary **pyMorfologik** was used (c=[,adj','subst','verb']).

The example of the results of the procedure conduction are as follows:

```
[ (, film' , {, film' : [, subst:sg:acc:m3+subst:sg:nom:m3']}),
  (, typowym' ,                                    {, typowy' :
[, adj:pl:dat:m1.m2.m3.f.n1.n2.p1.p2.p3:pos+adj:sg:inst:m1.
m2.m3.n1.n2:pos+adj:sg:loc:m1.m2.m3.n1.n2:pos']}),
 (, przykładem' , {, przykład' : [, subst:sg:inst:m3']}),
 (, mechanicznej' , {, mechaniczny' : [, adj:sg:dat:f:pos+adj:sg:gen
 :f:pos+adj:sg:loc:f:pos']}),
 (, zamiany' , {, zamian' : [, subst:pl:acc:m3+subst:pl:nom:m3+subst
 :pl:voc:m3'] , , zamiana' : [, subst:pl:acc:f+subst:pl:nom:f+subst
 :pl:voc:f+subst:sg:gen:f']}),
 (, miejsc.' , {}),
 (, bohater' , {, bohater' : [, subst:sg:nom:m1']}),
```

Stemming / Lemmatization procedure

Taking into consideration the advantages of the lemmatization process for semantic analysis of large texts, as well as the peculiarities of Polish language and the availability of realization of this procedure, the authors have made a decision to apply the procedure of **lemmatization** in the algorithm of films reviews preprocessing.

As a result, for each document D_j of the text collection τ we get a list of terms $\phi_t^{D_j} = \left(t_1^{D_j}, t_2^{D_j}, .., t_R^{D_j}\right)$,

ϕ^{D_j} = [, film′ , , typowy′ , , przykład′ , , mechaniczny′ , , zamiańa′ , , miejsce′ , , bohater′ , , dobrze′ , , mężczyzną′ , , przygoda′ , , rozwijać′ , ‘mało′ , , oryginalny′ , , znakomicie′ , , znany′ , , męski′ , , film′ , , rodzaj′ , , schemat′ , , banalny′ , , kryminalny′ , , intryga′ , , niesprawiedliwy′ , , wyrok′ , , odsiadka′ , , ucieczka′ , , wielki′ , , pościg′ , , prawdziwy′ , , winowajca′ , , dodać′ , , prawda′ , , wątek′ , , macierzyński′ , , cierpienie′ , , bohater′ , ‘, tęsknić′ , , synek′ , , pragnąć′ , , odnaleźć′]

Presentation of documents in the Vector form (Vector Space Model) procedure

Each document is presented as a **Plane model** $Mod_t = \langle \phi_t, F_D \rangle t$, which is offered as a combination of the following elements:

– the one-dimensional vector of the terms $\phi_t^D = \left(t_1^D, t_2^D, .., t_{S_D}^D\right)$, which text-opinions contain, where S_D – the total number of dominant terms in the document D;

– the one-dimensional vector of the terms weight (importance) for the documents $\boldsymbol{F}_D = \left(\boldsymbol{F}_{D_1}, \boldsymbol{F}_{D_{2i}}, .., \boldsymbol{F}_{D_S}\right)$.

One of the options of presentation of weight coefficients, which reflect the significance of a word in the text collection τ, is the *relative frequency* of the *t-th* term occurrence in document D_j:

$$\boldsymbol{R}_{t_1}^{\tau} = \frac{k(t, L_D)}{S(L_D)} \tag{1}$$

where $k(t, L_D)$ – the number of *t-th* term occurrences in the document D_j; $S(L_D)$ – the total number of terms in the text of D_j;

According to the law of Zipfe reduction of the total number of words for the analysis allows to increase the number of significant words, i.e. to increase the resolution of the text analysis method.

In this connection one of the attributes, which characterizes the documents set τ, is the frequency model *FR,* which allows to forecast the frequency of a word by its rang (ordinal number in the frequency list, sorted by descending frequency) in the frequency list of the analyzed documents set τ:

$$FR_{t}^{\tau} = \{\{(t_1, FR_{t_1}^{\tau},), (t_2, FR_{t_2}^{\tau},),..,(t_m, FR_{t_m}^{\tau},)\}, \{(Len_1, FR_{1}^{\tau},),$$
$$\{(Len_1, FR_{1}^{\tau},), (Len_2, FR_{2}^{\tau},),..,(Len_{m_1}, FR_{m_1}^{\tau},)\}\} \quad (2)$$

where $FR_{t_i}^{\tau} = \sum_{j=1}^{n} FI_{t_i}^{D_j}$ – the total frequency of occurrence of the *t-th* term in a set of text collection τ, Len_{m_1} –number of terms which have the total frequency $FR_{m_1}^{\tau}$

The frequency model *FR* can be presented as a frequency distribution chart (figure 4). The authors suggest the following rule of defining the **dominant** terms in the documents set τ: *It is suggested that the dominant are the terms, which have the total occurrence frequency as follows*:

$$FR_{t_i}^{\tau} < \max(Len_j), j = \overline{1, m_1}$$

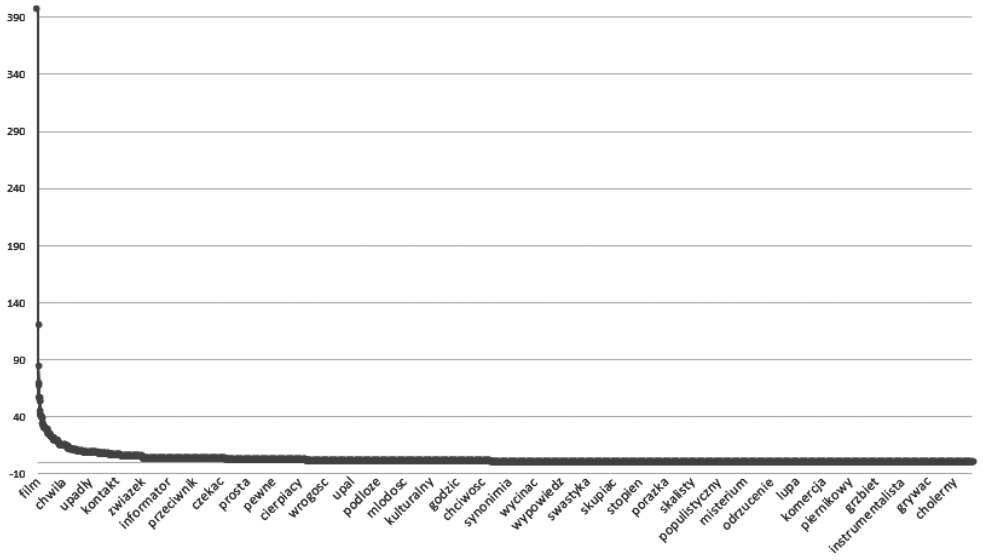

Figure 4. Frequency Model FR

Forming the documents models database procedure

With the objective of possible fixation and conduction of further research of the results of Preprocessing procedure, the authors have decises to save the vector models of the document in a database.

The database format is presentes in the figure 5:

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	number	integer							*NULL*
2	firstword	text							*NULL*
3	counword	integer							*NULL*

Figure 5. Documents Models Database STRUCTURE

Each *Number* of the document is matched with a certain set of dominant terms *FirstWord* and the frequency of this word occurrence in the document – *CountWord*.

Preprocessing Results Analysis

The *complexity* of the texts preprocessing algorithm is on the average *O(N)*. An interesting fact is that in connection with the introduction of additional preprocessing stages of the algorithm, the speed of program execution can be increased an average of 10-15% (especially due to additional procedures of text adaptation and part-of-speech tagging).

As an initial sampling for development of the algorithm of Preprocessing of film's text reviews, the corpus of 1000 reviews was used.

For conduction of the test experiment the program in Python language was used (the code of the program is attached); the database *BagOfWord* was created by the results of testing the Preprocessing algorithm.

To test the developed algorithm the sample (55 reviews) were processed (7 of them appeared to be empty). With the objective to increase the resolution of the developed algorithm, three parts of speech were selected as the conditions for selection – c=[*,adj','subst','verb'*]. As a result, the database including 6327 entries, was obtained.

The conclusion on the obtained results are the following:

1. The percentage of the dominant terms, remaining after the pre-processing, is averagely from 4,161% to 31,754% (Table 4).

Table 4. Structure of Distribution of the Number of Words Remaining After the Preprocessing

Bin	Frequency
4,161%	1
17,957%	26
31,754%	10
45,551%	8
59,347%	2
73,144%	0
92,00%	1

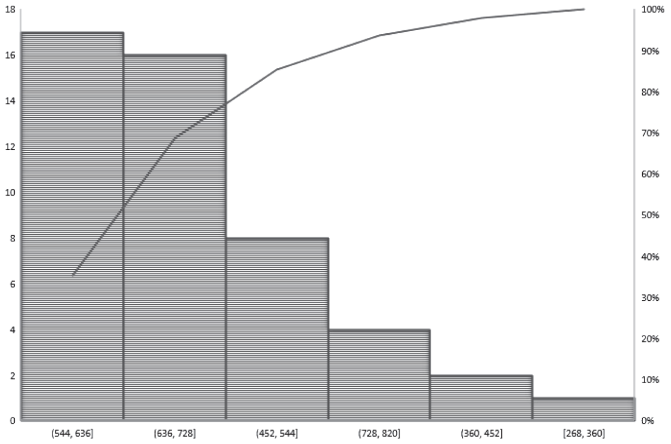

Figure 6. Structure of Distribution of the Number of Words Remaining After the Preprocessing

2. Zipfe law for the obtained documents models has the following peculiarities (table):
   - the largest percentage is for the words, which repeat once – 41,11% (in the range from 0% to 70,61%) and 2 times – 22,11% (from 9,96% to 58,59%);
   - maximum number of repetitions is 17 and is observed only in one document;
   - the average number of repetitions (from 5 to 12) is in average 12,23%.

The following classification of documents in accordance with the law of words distribution in the text, is suggested:

– "*Uneven Distribution*" – documents, characterized by the presence of several very frequently repeated words and the low frequency of the other words (figure 7):

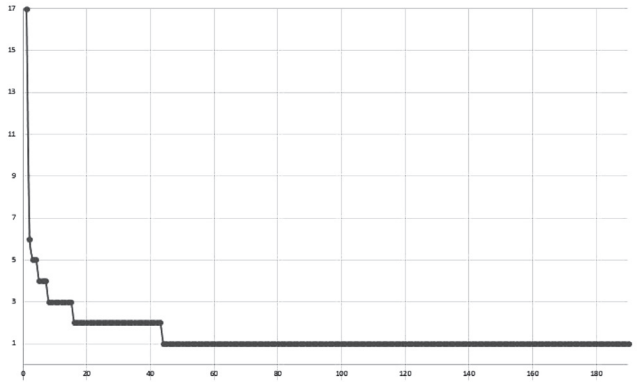

Figure 7. The Example of Zipfe law for the Group of Documents with the "*Uneven Distribution*"

– "*Uniformly Medium Distribution*" – documents, characterized by a uniform frequency of most of the words, and this frequency is average (figure 8):

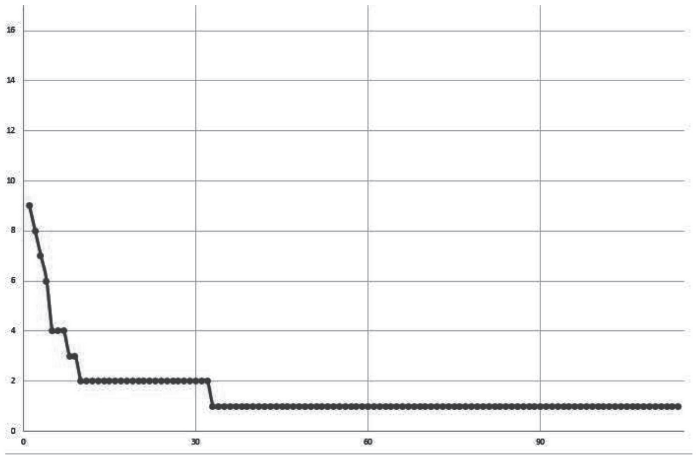

Figure 8. The Example of Zipfe Law for the Group of Documents with the "Uniformly Medium Distribution"

– "*Uniformly Low Distribution*" – documents, characterized by a uniformly low frequency of most of the words (figure 9):

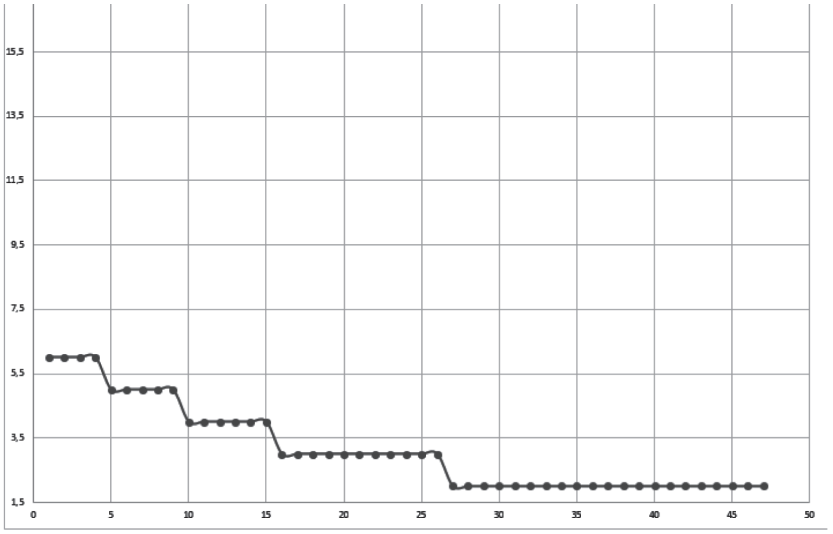

Figure 9. The Example of Zipfe Law for the Group of Documents with the *"Uniformly Low Distribution"*

Generalized structure of documents in terms of their classification into groups of Zipfe distribution laws in the tested sample is as follows (Table 5):

Table 5. Documents Classification into Groups of Zipfe Distribution Laws

Classes	% of documents
Uneven Distribution	33,33%
Uniformly Medium Distribution	45,83%
Uniformly Low Distribution	20,83%

3. The most frequently observed Dominant terms (from the sample) were defined, with their division into speech parts (Table 6):

Table 6. Most Frequently Observed Dominant Terms

Dominant Term	Count	Parts of Speech
film	338	subst
bohater	75	subst
widz	57	subst
scena	51	subst
reżyser	50	subst
kino	42	subst
akcja	36	subst
historia	29	subst
kultura	24	subst
postać	23	subst
obraz	23	subst
rola	20	subst
rzeczywistość	20	subst
aktor	20	subst
scenariusz	19	subst
gra	18	subst
wielki	37	adj
amerykanski	29	adj
prawdziwy	28	adj
dobry	27	adj
mozny	20	adj
duzy	19	adj
musiec	17	adj
kolejny	16	adj
wazny	15	adj
spoleczny	15	adj
moc	39	verb
zostac	35	verb
chciec	24	verb
wiedziec	22	verb
widziec	25	verb

By the results of the analysis it was revealed that the largest share of dominant terms with high frequency of occurrence in the documents belongs to the nouns (*subst*) (Table 7).

Table 7. Structure of the Most Frequently Occurring Parts of Speech

Parts of Speech	% in the Document's Models
subst	69,84%
adj	19,05%
verb	11,11%

Conclusions

Thus, the authors have developed the algorithm and the software for conducting the procedure of Preprocessing of the reviews of films in Polish language.

The analysis of results of this procedure conduction on the test sampling of reviews was performed. It allowed to define the major specificities of the processed texts, namely:

- the structure of documents in terms of **them containing non-significant words** (percentage of the dominant terms remaining after the Preprocessing is in average from 4.161% to 31.754%). These results show the high quality of the Preprocessing procedure;
- the structure of documents in terms of **dominant terms distribution frequency** in the documents, which allowed to define classes of the documents possessing similar structure;
- the structure of documents in terms of **them containing different parts of speech** (the large share of dominant terms – around 70% – with the high frequency of occurrence in the documents belongs to the nouns (subst)).

In future, these features will be taken into account during the performance of semantic analysis of the corpus of reviews.

In general, the main result of this phase of research is the existence of a „bag of words", prepared for the analysis of semantic similarity of the documents.

Bibliography

1. Vanyushkin A. S., Grashchenkov L.A. (2016); *Methods and algorithms extracted keywords*. New information technologies for automated. № 19.
2. Rizun N., Kapłanski P. & Taranenko Y. (2016); *The Method of a Two-Level Text-Meaning Similarity Approximation of the Customers' Opinions*. Czasopismo „Studia Ekonomiczne – Zeszyty Naukowe". Uniwersytet Ekonomiczny w Katowicach. 296, pp.64-85.
3. Rizun N., Kapłanski P. & Taranenko Y. (2016); *Development and Research of the Text Messages Semantic Clustering Methodology,* The Third European Network Intelligence Conference (ENIC 2016). Proceedings. DOI: 10.1109/ENIC.2016.33. In book: 2016 Third European Network Intelligence Conference, Publisher: ENIC.2016.33, pp.180-187.
4. Kapłanski P., Rizun N., Taranenko Y. & Seganti A. (2016); *Text-mining Similarity Approximation Operators for Opinion Mining in BI tools*. Proceeding of the 11th Scientific Congerence "Internet in the Information Society-2016", Publisher: University of Dąbrowa Górnicza, Editors: Maciej Rostancki, Piotr Pikiewicz, Krystian Mączka, Paweł Buchwald, pp.121-141.
5. Feinerer, I., Hornik, K. & Meyer, D. (2008); *Text mining infrastructure in*: "R Journal of statistical software." 25(5). American Statistical Association.
6. Segalovich I. (2003); *A fast-morphological algorithm with unknown word guessing induced by a dictionary for a web search engine*. MLMTA-2003.
7. Koreniu T., Laurikkala Y, Järvelin K. & Juhola M. (2004); *Stemming and Lemmatization in the Clustering of Finnish Text Documents*. CIKM'04, November 8-13, Washington, DC, USA.
8. Alkula, R. (2001) *From plain character strings to meaningful words: Producing better full text databases for inflectional and compounding languages with morphological analysis software*. "Information Retrieval" № 4, pp.195-208.
9. Weiss D. & Stempelator A. (2013); *Hybrid Stemmer for the Polish Language*.
10. Lewandowska-Tomaszczyk B., James Melia P. (1997) PALC'97: *Practical Applications in Language Corpora,* pages 496–505, Łódź University Press.
11. Hajnicz, E. & Kupść, A. (2001); *Przegląd analizatorów morfologicznych dla języka polskiego*. IPI PAN Research Report 937, Institute of Computer Science, Polish Academy of Sciences, Warsaw.
12. Vetulani, Z. & Obrębski, T. (1997); *Morphological tagging of texts using the lemmatizer of the 'POLEX' electronic dictionary*. In: Lewandowska-Tomaszczyk, B. & Melia P. J. (Eds.) Practical Applications in Language Corpora, Proceedings, University Press, pp. 496-505.
13. Obrębski, T. & Stolarski, M. (2006); *UAM text tools – a flexible NLP architecture*. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, pages 2259-2262, Genoa. ELRA
14. Miłkowski, M. (2010); *Developing an open-source, rule-based proofreading tool. Software*: "Practice and Experience". 40(7): pp. 543-566.
15. Wolinski, M, Miłkowski, M., Ogrodniczuk, M., Przepiórkowski, A. & Szałkiewicz, L. (2010) *PoliMorf: a (not so) new open morphological dictionary for Polish.*

16. Przepiórkowski A. & Wolinski, M. (2003); *The unbearable lightness of tagging*: A case study in morphosyntactic tagging of Polish. In Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora. EACL 2003, pp. 109- 116.
17. Radziszewski A. & Maziarz M. (2011); *Developing free morphological data for Polish,* "Cognitive Studies / Etudes Cognitives" (lista ERIH), 11.
18. Rizun N., Taranenko Y. & Waloszek, W. (2017); *The Algorithm of Modelling and Analysis of Latent Semantic Relations: Linear Algebra vs. Probabilistic Topic Models*. Knowledge Engineering and Semantic Web. Knowledge Engineering and Semantic Web, Publisher: Proceedings of the 8th International Conference (KESW 2017), pp.53-68. DOI: 10.1007/978-3-319-69548-8 5.