Automatic Universal Decimal Classification based on frequency advanced method

Fiodor Tretyakov, Liya Serebryanaya

Abs*tract*—Text classification is one the most valuable domain of Text Mining. It can helps identify a category of a text. Universal Decimal Classification (UDC) is wide-spread system at ex USSR territory. UDC—is a category tree. Each scientific publication should be related to the class. In Belarus, publication acquires UDC code only in libraries of universities with librarian worker. This article dedicated to automation of this task.

***Keywords*—**Classification, Artificial Neural Network, Text Mining, Universal Decimal Classification.

I. Introduction

Today, there are large number of disordered text information. Therefore, a search and a classification of the information by keywords are the most important tasks. It is too actually, because researchers often have to read a lot of scientific articles, before finding something important for them. Sometimes they can just look for an article, to identify its sense, and, sometimes, that is necessary to read most of the text to understand its meaning.

To identify a class of an article, it was invented Universal Decimal Classification (UDC). It is a mandatory attribute of any printed scientific work. With UDC it is easy to classify the information in the world of science, literature and art, periodicals, different kinds of documents and scientific articles [1].

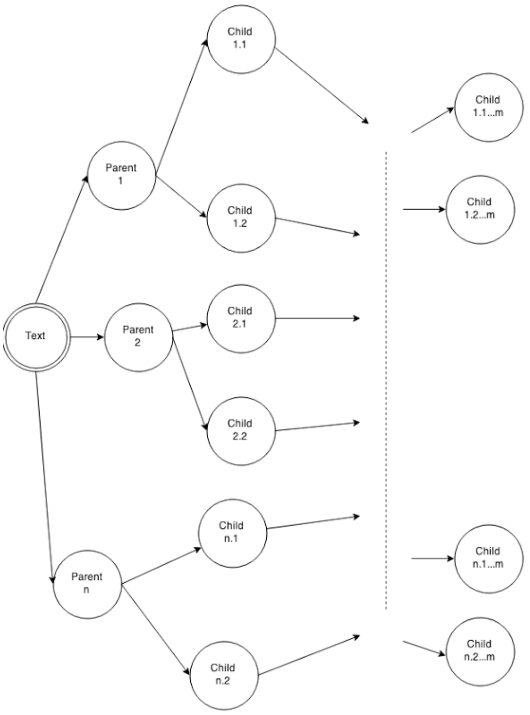

Fig. 1 UDC hierarchy

F. Tretyakov, Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus, (e-mail: Fiodor.Tretyakov@gmail.com)
L. Serebryanaya, Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus, (e-mail: l silver@mail.ru).

Currently, UDC is assigned manually at librarians or specially trained people. This article dedicated to methods and models to automatically assign the UDC, without involving humans. Therefore, the purpose of the work can be defined as the automation of the Universal Decimal Classification (Fig. 1).

II. Ways to Solve a Problem

The objective is to assign for each text from *n* texts a category *m* with UDC.

Subject of research work is Universal Decimal Classification of texts.

There are a lot ways to solve the problem. At first, there is should be defined resolving rule and a dividing function. The system processes texts and searches metrics, which will be substituted as parameters in the dividing function, and as a result text will be related to one of classes.

One of the significant factors influencing the choice of the class is the language of the text [2]. This article deals it with Russian text. Therefore, to build a classification algorithm, there are will be used features of Russian language.

One of the most popular ways of classification is to find the full match. However it can born the collision. For example, the user specifies the search word *koshka* (cat—in English). So, in the text or in the set of texts will be highlighted words which consist of a *koshka* (cat) and any extension of the word. This search is clear, but it cuts off the results, which could be useful to the user. For example, words *koshachiy* (cat's), *kot* (cat). This method is the fastest, but with low accuracy. But software that will created based on this method will not real-time based, so high-performance is not main goal [3].

Russian language has difficult structure and a lot of word-parts. So, the situation, when two words should relate to one class is too often. And to increase accuracy, the best solution is to select a part of a word that reflect its sense. This can often be the root of the word, but it is difficult to select it. For example, some part of a word like a fugitive vowel can't be easy separated, so it decreases accuracy [4] [5].

Stemming—is a one way to select a sensible part of the word [5]. The problem of finding the sensible part of word is a long-standing problem in the field of computer science. The first publication on that dates back to 1968. Stemming is used by search engines to increase accuracy of the user's search query and it is a part of the process of text normalization. Nowadays, there are a lot of various implementations of Stemming algorithms. They are used for various tasks of intelligent processing of text information.

Stemmer—it is a method that implements Stemming algorithm [5]. It can extract a sensible part of a word (stem). However, the stemmer can create issues that classified next way.

A. Stemming issues of the 1st class

Stem gives too much generalization, and this way one part of a word can be related to more that. This is the largest group Stemming issues. For example, the Stemming of a word *vami* (by you) gives a stem *vam* (to you), but it can be happen it a word *vampir* (vampire). In Russian, it can be very difficult to completely avoid such errors. For example, a modification of a word *past'* (fall) gives forms *pad* (fall) and *pal* (fell). The result of Stemming gives stems—short forms of words, and this is a great extension in the field of the search. To fix such kind of issues, it should be completing removing of *stop words* from text using Ziph's algorithm and *stop-words dictionary*.

B. Stemming issues of the 2st class

A truncated form gives too long tems that are not matched with certain grammatical forms of the same word. Such issues lead stemmer's developer desire to find a compromise with the issues of the 1st kind in the case, when word's form changing. In English, there are exists the same thing. For example, a group of irregular verbs. In Russian, this means that, different forms of a word change the first stem, and this phenomenon is very often. As an example, which is usually born many implementations stemmer. For example, words *koshka* (cat) and *pachka* (pack) have forms *koshek* (cats) and *pachek* (packs). Usually stemmer perform in these cases truncation to *koshk* (cat) and *pachk* (pack) that are not comparable with forms of the genitive and accusative plural.

C. Stemming issues of the 3st class

Stem can not be built because of the changes in the root of the word, which leaves a single letter in tem. Either model inflection involves the use of prefixes. The second event occurs within the grammar dictionary for comparative degree of adjectives and adverbs in Russian. For example, a word *pokrasivee* (more beautiful) as a form of an adjective *krasivyj* (beautiful) or an adverb *pomedlennee* (slower) as a form *medlenno* (slow).

Among all implementations stemmer can be represented two types:

1) Which use a dictionary to extract words;
2) Which use a heuristic model [4].

To isolate the word's root, there was developed a software module that implements Stemmer algorithm. Stemmer uses a heuristic model.

III. Stemming algorithm

To create the a heuristic stemmer, there are be should used dictionaries of endings, gerunds and participles forms, prefixes and suffixes. Stemmer will heuristically determine the part of speech based on dictionaries content. The essence of the algorithm is reduced to the determination of parts of speech to words by them endings, using dictionaries endings. For example, the ending of the sacraments can not be confused with anything else, so, this way, Stemming begins with them.

Stemming will use the following algorithm.

1) Complete a search for participles and gerunds endings in the word. If them found, them should be removed and algorithm goes to step 3.
2) Search the endings of adjectives, verbs or nouns. If they were found, they should be removed.
3) If a word ends in *i*, it should be removed.
4) From the beginning of word, there is seeking the sequence kind *vowel-consonant*. All letters after this combination will be called block *n*. If it not exists or *n* is empty, the process should go to step 7.
5) There are looking for a block *m* in the block *n*. This is a sequence of *vowel-consonant* too. If it is not exists, or it is empty, the process should go to step 7.
6) There are looking for word parts *ost* and *ost'* in the block *m*. If they are found, they should be removed.
7) If a word ends with endings *ejsh* or *ejshe*, they should be removed.
8) If at the end of word a double letter *n* situated, the second letter *n* should be deleted.
9) If at the end of the word is letter *'*, it should be removed.

Stemmer allows you to search the text in Russian more accurately. The issue is the complexity of the module.

IV. Text parser module

Next past is a text parser module. To apply Stemmer, text should be split by words. It can be done by representing the text like a line and parse it by non-word symbols.

Not all words are significant for classification. They are called stop words. There are removing prepositions and erroneous words.

Another thing is help the system to select keywords—is the Ziph's Law. It allows to exclude words that has no sense for processing. There are words that misspelled, word-parts and so on.

There is an algorithm of keyword finding:

1) Split text *t* by words by ". , ! ? *space tab lineend*".
2) Remove non-valuable words by stop word dictionary multiplying (just vector multiplication).
3) Select keywords by Ziph's law.

V. Classification module

The next step is to create a classification module.

The classification method will be based on the Stemmer.

The UDC represented like a tree and consists of 126 441 categories. This is to much to start a classification directly. This method will has to low performance and to avoid it, it is a great improvement to use embedded feature of UDC—the hierarchy. The searching of a category reduces to passing the tree. But it require the special tree extension of tree—the siblings. So the algorithm will be built as following:

1) There is the processing of descriptions of all categories with the module by extracting the roots of words and put the result in the *word dictionary*. Every line in this dictionary is a key, which is the root of the word, and the value in a row—the number of word forms by a key from the description of the category.
2) Complete the Step 1 for all texts, applying them to these abstracts.
3) Select the initial list of categories. Let it be the children of the invisible parent category (the root UDC category).
4) Select begging category set. Its number is determined by the value of the variable *T*, calculated by formula 1.
5) There is a variety of categories for the text, where *T* is the maximum. If such *T* more than *1*, then the system will process all trees parallel the result is the UDC-code connected by + a list of categories.
6) When this category has a subcategory, next searching set is a category's children plus the category, because the text could refer to it. This case method goes to step *7*. If subcategories do not exist, the algorithm ends.
7) Look *T* for selected categories. If the largest *T*-value for the parent, then select it and finishes the algorithm. If subcategory "wins" (*T*-value for it is maximal), the system goes to step *6*.

$$T = \sum_{i=0, j=0}^{n,m} Ai \times Bi$$

VI. Artificial neural network

The result obtained with this classification can be improved if it will be introduced machine learning. All words selected form texts can be divided by two groups: existed in the text or not. And the words that are not included in the text's dictionary are markers of a category. Hence, they should be marked as belonging to the category. The ideal solution for this case is artificial neural network (ANN).

Artificial neural network is machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown.

Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature.

A neural network is convenient to build on the basis of the perceptron. It will be a single-layer linear activation function, since there is no need to introduce some complex logic. There is a representation of the artificial neural network in the form of a tree.

To start the system need to train a neural network. First selection is 10000 texts.

VII. Results

Now it is necessary to compare the methods in the field of accuracy. In the case of absolute precision: the method of frequency classification based on the full-text search (A), the method of frequency classification using Stemming (B), the frequency method without the Stemming but with the ANN (C), frequency method using both the Stemming and the ANN (D), a semantic reference expert system (pseudo-reference confidence 90%) (E) (Fig. 2).

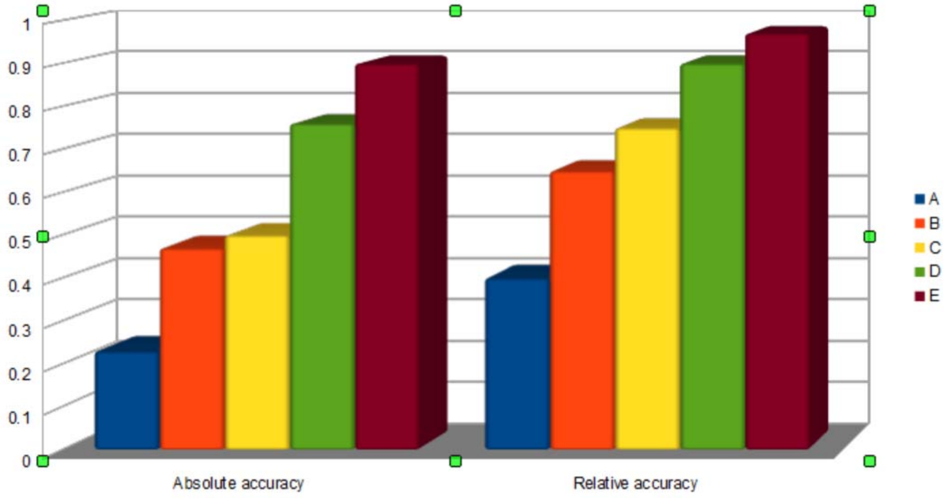

Fig. 2 Results

The graph shows that D (76%) is very close to the format of E. At the same time as the difference between B (47%) and C (50%) is not so much. Full-text frequency method gives the accuracy of only 23%. This chart reflects the confidence of the classifier with the choice.

If there are using relative accuracy, then it is E 97%, D show excellent results—90%. The difference between C and B increases—75% and 65% respectively. For A is—40%.

VIII. Conclusion

With stemmer, texts can be classified with high accuracy. The minus is the complexity of the architecture of the module.

To demonstrate the performance of this algorithm must be integrated into a software package.

The package has the following specification:

1) Receive an input text.
2) Classify.
3) Push the text in the database with the appropriate index to account for this result in the following classifications.
4) Print the results to the user on the screen.

The result is a system that allows user to assign UDC to text with high speed, accuracy and it is automated.

ACKNOWLEDGMENT

The authors would like to thank to the Head of Belorussian State University of Informatics and Radio-electronics—Mikhail Batura, to the Head of Computer Systems and Networks Department, and to the Head of Software of Information Technologies Natalia Lapickaya.

REFERENCES

[1] V. Tolstoy, “Deep analysis of the text. From the series of lectures "Modern Internet-technologies" for students of the 5th grade of the Department of Computer Technology Faculty of Physics”, Donetsk, Ukraine: Department of CS, 2005.

[2] F. Tretyakov and L. Serebryanaya, “Russian texts' classification method”, Minsk, Belarus: International scientific and technical conference dedicated to the 50th anniversary of the MRTI-BSUIR source book, 2014.

[3] F. Tretyakov and L. Serebryanaya, “Automated method of universal decimal classification in the field of distance learning”, Minsk, Belarus: Distance Learning - Educational environment XXI century source book, 2014.

[4] P. Braslavsky, “Favorite applied problems of computer science”, Moscow, Russia: Williams, 2005.

[5] R. Duda, P. Hart and D. Stork, “Pattern classification”, New-York, USA: John Wiley & Sons, 2001.

This publication is the result of the project implementation:
TEMPUS CERES: Centers of Excellence for young RESearchers.
Reg.no.544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES