Porter advanced method for the Universal Decimal Classification

Fiodor Tretyakov, Liya Serebryanaya

***Abstract*—**Text classification is one the most valuable domain of Data Mining. It can helps identify a category of a text. Universal Decimal Classification (UDC) is wide-spread system at scientific world. 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. For now, there is doesn't exist analogue of automated decimal classification.

***Keywords*—** Porter stemming, TF/IDF, Data 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.

Dewey #	10 Main Classes	Kinds of Books
000-099	General Works	encyclopedias, almanacs, record books, such as Guinness
100-199	Philosophy and Psychology	paranormal phenomena,such as ghosts, ethics, how we think
200-299	Religion	mythology, religions
300-399	Social Science	government, holidays, folklore, fairy tales, education, community
400-499	Language	English and foreign languages, sign language, dictionaries
500-599	Natural Science	math, chemistry, biology, weather, rocks, plants, animals in nature
600-699	Applied Science	inventions, health, drugs, transportation, cooking, pets
700-799	Fine Arts and Recreation	crafts, art, drawing, painting, music, games, TV, movies, sports
800-899	Literature	short stories, poetry, plays, jokes, riddles (fiction could be here)
900-999	History and Geography	countries, flags, historical events, biographies (92 or 920)

Fig. 1 UDC table

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

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).

information in the world of science, literature and art, periodicals, different kinds of documents and scientific articles [1].

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].

III. 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 Zipf'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 line end".

2)Remove non-valuable words by stop word dictionary multiplying (just vector multiplication).

3)Select keywords by Zipf's law.

IV. 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 too 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.

$$T = \sum_{i=0,j=0}^{n,m} A_i \times B_i$$

V. TF/IDF ALGORITHM

Tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model [2].

tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist.

In the case of the **term frequency** tf(t,d), the simplest choice is to use the *raw frequency* of a term in a document, i.e. the number of times that term t occurs in document d. If we denote the raw frequency of t by $f_{t,d}$, then the simple tf scheme is tf(t,d) = $f_{t,d}$.

- Boolean "frequencies": tf(t,d) = 1 if t occurs in d and 0 otherwise;
- logarithmically scaled frequency: tf(t,d) = 1 + log $f_{t,d}$, or zero if$f_{t,d}$ is zero;
- augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document:

$$tf(t,d) = 0.5 + 0.5 \times \frac{f_{t,d}}{max\{f_{t',d} : t' \epsilon d\}}$$

The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient [3].

$$idf(t, D) = log \frac{N}{|\{d \epsilon D : t \epsilon d\}|}$$

with

- N: total number of documents in the corpus N = |D|
- $|\{d \epsilon D : t \epsilon d\}|$: number of documents where the term *t* appears (i.e., $tf(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \epsilon D : t \epsilon d\}|$.

Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then tf–idf is calculated as

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0 [1].

VI. 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).

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%.

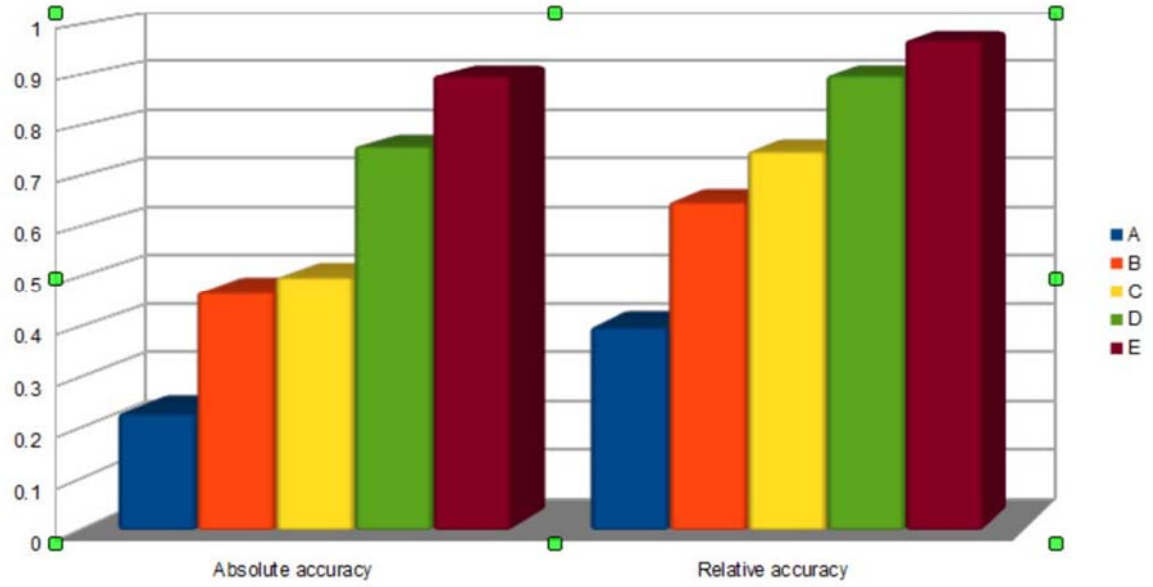

Fig. 2 Results

VII. 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 Radioelectronics— 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 sourcebook, 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 sourcebook, 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.