TOPIC-BASED APPROACH TO DOCUMENT CLASSIFICATION

Nikita Voronkov

The Department of Applied Mathematics and Computer Science, Belarusian State University e-mail: *voronkov@scnsoft.com*

Abstract. The article addresses the problem of document classification. A technology for automatic topic extraction from documents and application of the topics as document representation in classification task are described. The topics are understood as noun phrase-based main themes of documents. The suggested algorithm can also be used to improve the quality of automatic document clusterization.

Introduction

Very often people have to deal not with one or dozens of documents, but with hundreds and thousands of documents of different types. And these documents are not sometimes classified according to certain characteristics. Apparently, it becomes very difficult to find any information within large amount of documents without their classifications. In such situation users commonly employ different automatic classification and clusterization systems which aim at reducing information quantity to be processed.

Generally, the task is to classify the input documents into several parts (classes) in accordance with some predefine properties of these classes.

We will consider a classification task when a user has a set of documents for each class that represent these classes and the system itself will build properties of the classes. We define topics (main document themes) as class properties. To automatically extract topics from the text we use summarization system which builds topic-oriented summary [6].

1. General scheme of topic-based classification algorithm

A process of topic-based classification consists of the following main steps:

1. Topics extraction;
2. Topics tree building and main topics selection;
3. Classification that comprises:
   a) Building classes images using topics extracted automatically from a user’s training corpora;
   b) Document classification based on topics extracted automatically from a document and comparing them with classes images.

A general scheme of topics extraction can be described by the following steps:

1. Pre-formatting. There are a lot of document formats, so it is necessary to convert them in one standartizated format to process. The input of pre-formatter module is documents in different formats (txt, rtf, doc, pdf, html etc.). The output of this module is text, which is split on sections, paragraphs, titles, etc. Besides, pre-formatter filters any auxiliary information such as buttons’ text, scripts, menus and so on.
2. Part of speech sentence tagging, extraction of syntactical and semantic relations from sentences (SAO: subject-action-object). On this step sentence splitting, part of speech tagging and different types of relation extraction take place [1,3].
3. Topics and main topics extraction itself:
   a) Statistics collection.
   b) Topic tree building.
   c) Main topics extraction.

Topics are the most important simple noun phrases extracted from the document. The process of topics extraction is described below in more detail.

2. Topics extraction

A process of topics extraction is divided into statistics collection and several filtration stages.On the stage of statistics collection system accumulates information about words frequency.

To make our statistical algorithm more effective, a set of informative word tags is extracted. Informative words are nouns, adjectives, verbs, adverbs, proper nouns, etc. Uninformative words have tags such as prepositions, articles, numeral and some other tags. Our system takes into account only statistics of informative words.

We also use a set of additional coefficients to improve quality of statistical algorithm. During the processing of each sentence words get some weight taking into consideration which of the following conditions is met:

- word comes across the sentence;
- word is a part of noun phrase;
- word is a part of main semantic/linguistic word of noun phrase;
- word is a part of SAO;
- word is a part of S and O of SAO and main semantic/linguistic word of S and O;
- word is a part of text title.

Every document section has some coefficient of importance of this section in document. So, we define the most important section in the document where the word came across and then we scale word frequency according to coefficient of this section.

After all, we normalize the weights of all the words so that the weight of the most important word becomes equal to 1.

Each simple noun phrase extracted from sentences is a candidate to become a topic [4, 5]. Preliminarily, system transforms each of them according to the following rules:

- split noun phrase by “or” and “and”;
- filter uninformative words from noun phrase by tags (articles, etc.);
- filter uninformative words from noun phrase by dictionary (“said”, etc.);
- noun phrase transformation (producers of hazardous materials -> hazardous materials producer);
- canonization of main linguistic word (if it is plural, the system makes it singular);

Then, whole noun phrase filtration takes place. Our system makes it in 2 stages:

1. The system filters uninformative noun phrases by tags. For example, if some noun phrase consists of words with special tags or there are some words with non-letter symbols and there is no informative word in this phrase, so this noun phrase is considered as uninformative.
2. On the second stage system filters noun phrases, which consist of special predefined words. For example, “step + letter”. In this case noun phrases such as “step A”, “step B” etc. will be filtered.

If some noun phrase was not filtered during these 2 stages, we add it to a list of informative noun phrases. We also store location of this noun phrase in the sentence and in the document.

When all sentences of document are processed, we get the list of all informative phrases extracted from text. Then, each noun phrase from this list gets some statistical weight based on word weights, which are counted by statistical algorithm by this time. Noun phrase weight is equal to the arithmetic mean weight of all the words it contains.

Usually, there are a lot of noun phrases in each document and there are a lot of unique noun phrases, so it makes sense to cut off some noun phrases with low weight. For this purpose, all noun phrases are sorted in decreasing order of the weights assigned. Than, it is possible to apply one of the following algorithms to cut off noun phrases:

- leave not more than a predefined number of the heaviest topics;
- leave noun phrases with the weights greater than some predefined value;
- leave noun phrases with the weights greater than the weight of the heaviest noun phrase multiplied by some predefined coefficient;
- among sorted informative noun phrases such a number of the most difficult ones is being chosen, which does not allow their total weight fall below 80% of the total weight of all noun phrases.

For our system we have chosen the last type of noun phrase set cut off with coverage of 80%. All noun phrases remaining will be called topics. Then, from the topics obtained a hierarchical tree is built.

3. Topics tree building

To accomplish this, all topics are sorted in increasing order of their lengths (considering informative words only) and within the group of words of the same length topics are sorted in increasing order of their weights. Then, moving from the last list element towards the beginning of the list, the most suitable parent is sought for every topic. The criteria for selection the best parent is the following:

- it is contained in the child;
- it length and weight are the highest.

Checking whether a topic is contained in another topic requires O(l1 + l2) operations, where l1 and l2 – are lengths of the topics in informative words. Finding the best parents requires O(n*n) operations, where n - is a quantity of topics.

However, we can propose some modifications to this algorithm, which show the same complexity, but in practice works 5 to 10 times faster in average.

1. All topics are sorted in increasing order of their lengths (considering informative words only) and within each group of words of the same length topics are sorted in increasing order of their weights;
2. Taking into account that all words in text have unique identifiers from 0 to M - 1, where M is the number of unique canonized wordforms in the text, for each word there is built a list of IDs of noun phrases where the word was found;
3. Moving from the last list element towards the beginning of the list, there is built a list of noun phrase IDs where each ID corresponds to a noun phrase where all informative words of each topic occurred;
4. The list of noun phrase IDs is sorted in decreasing order;
5. The best parent is sought in the order of the above list of possible candidates among those of them that have the ID smaller than the ID of the current noun phrase.

Then, root vertices and all children of each vertex are sorted in decreasing order of the weights. Thus, a hierarchic tree of topics is obtained.

Then, from every tree branch the main topic is selected which is the one that shows the major number of occurrences in the text.

4. Classification

A process of document classification consists of so-called training stage – building classes images by pre-processing users set of documents describing classes he needs and classification itself – relegating any new document to one of the classes.

4.1. Building classes images

We define classes and documents image as a set of weighted topics extracted by summarization system [6].

The following steps describe automatic classes’ image construction based on user’s corpora of documents describing each class.

1. Extract topics with their weights from all documents of each class.
2. Collect topics of each class. Sum the weights of the same topics within each class and unique topics.
3. Filter topics with weight lower than some threshold.
4. Recount topics weight within each class.

$$T_i = \frac{t_i}{\sum t_k} \tag{1}$$

where t_i - the weight of *i-th* topic in class, $\sum t_k$ - total topics weight of a class.

So the weight of each topic is equal to the percent of the topic weight in total topics weight in its class and $\sum T_i = 1$.

4.2. Document classification

Each document gets estimation to each class by topics and by words. The total estimation is a combination of these two with some coefficients c1 and c2. During classification, for each document 2 vectors are built:

$$\begin{gathered} Dt = (Dt_1,...,Dt_N) \\ Dw = (Dw_1,...,Dw_N) \end{gathered} \tag{2}$$

where Dt_i - document estimation by topic with respect to class *i*, Dw_i - document estimation by words with respect to class *i*, *N* – classes quantity.

$$Dt_i = \sum_k (T_{ki} * \Delta_{ki} - Ft * (1 - \Delta_{ki})) \tag{3}$$

where T_{ki} – the weight of *k-th* document topic in class *i*,
Ft – constant, fine if topic is not found in this class,

$$\Delta_{ki} = \begin{cases} 1, \ if \ \ T_k \ \ exists \ in \ class \ i \\ 0, otherwise \end{cases} \tag{4}$$

$$Dw_i = \sum_k (W_{ki} * \Delta_{ki} - Fw * (1 - \Delta_{ki})) \tag{5}$$

where W_{ki} – the weight of *k-th* document word in class *i*, *Fw* – constant, fine if word is not found in this class.

$$\Delta_{ki} = \begin{cases} 1, \textit{ if } W_k \textit{ exists in class } i \\ 0, \textit{otherwise} \end{cases} \tag{6}$$

W_k is counted as the highest topic weight in the class the word W_k is appeared in.

At last, D-vector is counted:

$$D = (D_1 = c1 * Dt_1 + c2 * Dw_1, ..., D_N = c1 * Dt_N + c2 * Dw_N) \tag{7}$$

We relegate the document to class C, if $D_C = \max_k D_k$.

The testing training corpora consisted of 15 classes. Each class was represented by from 1700 to 2000 US patents. Each class was represented by from 6500 to 12500 topics. The training corpora reproduction (classifying training corpora itself) is equal to 95.5%. 34000 web documents were classified with precision of 70% and recall of 98%.

Conclusions

Proposed classification algorithm is not memory and speed efficient but shows rather high quality of class images building and subsequent document classification. Our method does not require manually assigning class properties by the user but extracts such properties automatically. Moreover, there is a possibility to automatically assign a name to each class in form of a set of topics with the highest weight. Proposed classification algorithm can also be used in multidocument summarization and clusterization if the number of documents is relatively high in which prior grouping of the documents by their relevance to one class provides for better quality of resulting summary and clusterization results.

References

[1]. Advances in Automatic Text Summarization. *The MIT Press*, 1999.

[2]. Sovpel I.V. "Issues on the implementation of natural language text analysis" *doctoral thesis*, Minsk, 1980.

[3]. Sovpel I.V. "Linguistical and technical principles, methods and algorithms of automatic text analysis", *Vishejshaja Shkola*, Minsk, 1991.

[4]. Salton, Gerard, Amit Singhal, Chris Buckley and Mandar Mitra, “Automatic Text Decomposition Using Text Segments And Text Themes”, *Proceedings of the Seventh ACM Conference on Hypertext*, Washington D.C., 1996.

[5]. Salton, Gerard and Amit Singhal, “Automatic Text Theme Generation and the Analysis of Text Structure”, *Cornell Computer Science Technical Report* 94-1438, July 1994.

[6]. Voronkov N.V., Sovpel I.V., “Automatic topic-oriented summarization”, *Text Processing and Cognitive Technologies. Paper Collection.* **№7**. Kazan, 2002, 94-102.