Materials Physics and Mechanics 9 (2010) 246-250 Received: March 17, 2010

DOCUMENT CLASSIFICATION USING WEIGHTED ONTOLOGY

Asta Bevainytė* and Linas Būtėnas

Department of Computer Science, Faculty of Mathematics and Informatics

Vilnius University, Naugarduko st. 24, LT-03225, Vilnius, Lithuania

* e-mail: asta.bevainyte@mif.stud.vu.lt

Abstract. This paper presents document comparison and classification model for Lithuanian language texts based on weighed ontology. The tests have been performed to measure several aspects: i) quality of comparison of documents; ii) optimal size of ontology; iii) type of part of speech words used to create ontology. Final results indicate 96% of correct classification cases and suggest that all the main part of speech terms should be used from the text. The proposed model can be used to classify texts more efficiently than keyword based systems.

1. Introduction

The standard process of searching for information is very limited. Usually we have an idea what we want to find. If we put it in written form, a short paragraph can represent our ideas quite well. But if we start searching for this idea in the Internet, we will have to compress it into 2-5 keywords. This step becomes a main problem as we loose a lot of meaning and other people might use different keywords in their texts. Usually it takes a lot of time to find information or documents we need from WWW.

We propose an idea of using weighted ontology for document comparison. This ontology is created automatically using entire document text, but contains only a fixed number of terms. Later we put weights on each connection between two words according to the distance between them. This weighted ontology of one document is compared to the ontology of some category and result is the similarity value. The main idea is to classify one document into one of the categories.

The method has several advantages:

- the difference of frequencies of terms is measured between documents,
- the distance between terms is taken to the account,
- combination of both features ensure that we have more precise document comparison only.

The main task was to check the quality of proposed weighted ontology system and find out the optimal size of it.

Ontology is an organized set of concepts having relations between them [2, 3]. In a general case, relation connects two terms and can define the type of connection like ‘synonym’ or ‘is a’ and etc. In our approach, the relation represents connection without any label. We try to represent the strength of relation rather than a type of it. A weight of relation shows the distance between terms. Bigger weight means stronger connection and thus allows us to judge if two texts having almost same words have the same structure.

Another issue, we have encountered, is the specific grammar of Lithuanian language. Almost every part of speech (nouns, verbs, adjectives) has different endings according to the situation they are used. To minimize the effect of it, we use word stems instead of words.

Document classification using different algorithms is described in many articles. One of the classic document comparison algorithms is tf–idf (term frequency–inverse document frequency). It is based on measuring statistical frequencies of terms [1]. The difference of frequencies represents the difference of ‘meaning’ between documents. Research in document classification field has many possibilities: some authors write articles about automatic ontology construction [4, 5], others about theme extraction of a document [6, 7] or classification of documents [8]. We have chosen to use automatic ontology construction together with automatic weight calculation between terms and use it for document classification problem. In following sections we describe document classification model and results of tuning and testing its performance.

2. Model

Document preparation and construction of weighted ontology. Every document goes through preparation and transformation to weighted ontology steps before it can be compared to another ontology. The preparation step clears the stop words from the text and cuts the endings of the words leaving only word stems. Stop words are defined as a set of words that are either used very often in every text, or don’t carry any essential meaning (like: and, or, thus, etc.). Afterwards all endings from nouns and adjectives are cut off – this step we call ‘stemming’. It is not 100% accurate because some endings of nouns, adjectives and other parts of speech are the same. As an example, we also cut endings from some verbs. As we show later, such inaccuracy is acceptable and does not reduce the quality of the algorithm. After stemming is done, all stems shorter than 3 symbols are removed. This is needed as stems of two or one symbol are meaningless. Later the frequency of terms in document is counted and only the most frequent ones are selected. The threshold of maximum number of stems was made variable to see which the best is. We need to take only the most important words, as it would increase the speed of document processing and will reduce the quality of comparison very little. Technically weighted ontology is stored as matrix of size N^2, where N is the number of terms taken from the document. It is obvious that increase in a document size would result in exponential increase in computation time. Let M_n be the matrix of document D_n, where $n \in N$ and N is the set of all document numbers. Let m_{ij} be the element of the matrix M_n holding weight between terms (stems) t_i and t_j in the document D_n. The element m_{ij} is calculated in such way:

$$m_{i,j} = \sum_{\forall s} \sum_{\forall x} \frac{1}{\left| p_i - p_j^x \right|} \tag{1}$$

for all $i < j$, where p_i is term t_i position in the sentence s, p_j^x is term t_j position in the same sentence. Counting weight for all x (all positions of term t_i in the same sentence) gives us the representation how often two words are used together; m_{ij} represents it. Calculations are performed according to several rules:

- t_i is unique term in the whole document;
- $p_i \neq p_j$ means that we do not count relationships between the same terms, and thus $m_{ii} = 0$;
- relation between two terms positions is counted only once (according to the rule $i < j$);
- relations are calculated only inside one sentence, thus we need to count the sum from all sentences ($\forall s$).

The main issues are as follows: if the threshold is too high (we take many words into ontology) the comparison step takes too much time, if the threshold is too low (ontology is small) – the quality of comparison is lost. One of the aims of the paper is to determine the right size of ontology for Lithuanian language texts.

Construction of weighted ontology for the category. A set of related documents can define some category. To be able to find documents related to this category, we need to create a weighted ontology for all documents which we call ontology of one category (OOC as shortcut). For every document we use the method of ontology creation described in section 2.1. Afterwards, all ontology matrixes are incorporated into one. Every element in OOC matrix is a triplet holding three values: minimum, maximum and average weight of two terms. Minimum and maximum values are needed as some of the documents might have very low weights of some term pairs resulting in big difference values and thus distorting the real results. An element average $m'_{i,j}$ is calculated like this:

$$m'_{i,j} = \frac{\sum_{\forall n} m^{n}_{i,j}}{N}, \tag{2}$$

where $n \in N$. Maximum is always the biggest value from all $m_{i,j}$ elements, and minimum is either the smallest element, or equal to zero, if only one not empty $m_{i,j}$ element exists in all M matrixes.

Classification of a document. A document can be classified by comparing its own weighted ontology with a weighted ontology for the desired category. The comparison algorithm:

$$similarity = \frac{\sum_{\forall i,j} \left(\left| m'_{i,j} - m_{i,j} \right| * if\,(m'_{i,j}, m_{i,j}) \right)}{count\ (\ f\,(m'_{i,j}, m_{i,j}))} \times 100\ \%\ . \tag{3}$$

The part $if\ (m'_{i,j}, m_{i,j})$ results in:

- 1, if $m_{i,j} \in [m^{\min}_{i,j}, m^{\max}_{i,j}]$
- 1, if $m_{i,j} \in [m^{avg}_{i,j}, m^{\max}_{i,j}]$, then ($m^{\min}_{i,j} = null$) or ($m^{\min}_{i,j} = m^{\max}_{i,j}$ and $m^{avg}_{i,j} < m^{\min}_{i,j}$)
- 0, otherwise.

Similarity value shows document similarity to the compared category in range of 0–100%, where 100% means identical document and is a theoretical value rather than practically reachable one. It is because a document never contains exact amount of all term pairs with exactly the same weight values.

3. Results

In this part of article we describe the results of tests and conclusions derived from them. The main objectives are:

- Determining the quality of document classification system using Lithuanian language texts;
- Finding the optimal size of ontology;
- Clearing out what type of part of speech words should be used to create ontology.

The data used for testing was gathered from the internet. All documents were placed in three categories:

- S – relative texts, the main group for creating OOC matrix, including documents on medicaments;
- D – partially relative texts group; it consists of texts about medicine, illness and so on;
- N – non relative texts; it consists of documents about meals.

30 documents of S group were used to create OOC matrix. Afterwards 15 different texts from each group were used to establish similarity intervals. Later 10 texts of each group

were used for testing phase. During the testing if a document similarity value is clearly in one interval – the document is assigned to that class, if value falls in two or three intervals – we call it “unclassified”.

The amount of documents is not very large, but it is enough to check the model as the main idea of it, to be able to find relevance of documents given to several documents which a person already has. Usually people have several documents from which they can start the search, but not a huge electronic library.

Table 1. Similarity intervals depending on OOC matrix size.

Similarity	Ontology matrix size								
	20 terms			40 terms			80 terms		
	N	D	S	N	D	S	N	D	S
Minimal %	0	0	9.36	0	0.27	4.38	0	0.14	1.59
Maximal %	0.37	7.88	24.71	0.14	5.34	15.87	0.24	2.46	9.,90

There is a clear decrease of maximal similarity values and size of intervals in D and S categories in Table 1. This means that having a bigger OOC matrix decreases the preciseness of our algorithm. Intervals of all (N, D and S) categories overlap having 80 terms in matrix and interval of N is totally covered by D having 20 terms in a matrix.

In Table 2 the test results are presented. Looking at both testing results the best ontology matrix size is: i) 20 terms if we look for very similar documents; ii) 40 terms if we need to categorize documents in all three classes. Theoretically if we combine both methods (first doing categorization with 20 terms, later with 40 terms) we can obtain 96% correct answers, as only 10% of D documents will be incorrectly classified.

Table 2. Quality tests using similarity using OOC matrixes from table 1.

Classification	Ontology matrix size								
	20 terms			40 terms			80 terms		
	N	D	S	N	D	S	N	D	S
Correct	0%	90%	100%	100%	90%	70%	90%	90%	70%
Incorrect	0%	0%	0%	0%	10%	0%	10%	0%	0%
Belongs to 2 categories	100%	10%	0%	0%	0%	30%	0%	10%	30%

Table 3. Similarity intervals depending on OOC matrix size (only nouns)

Classification	Ontology matrix size								
	20 terms			40 terms			80 terms		
	N	D	S	N	D	S	N	D	S
Correct	0%	60%	100%	100%	70%	40%	100%	60%	40%
Incorrect	0%	30%	0%	0%	20%	0%	0%	30%	0%
Belongs to 2 categories	100%	10%	0%	0%	10%	60%	0%	10%	60%

Another task, we had, was to check if using all part of speech words is better then using only nouns. The main concern here is that while cutting the endings of the nouns we also cut some endings from the other part of speech words and incorporate them in OOC matrix. It was not clear if such thing is acceptable or is lowering the quality. To clear out this issue, we created OOC matrix completely from nouns. Table 3 shows the test results performed with new matrixes. It is obvious that results are worse. Hence we can make a conclusion that some of the frequent meaningful terms are needed even they are used not in basic forms.

4. Conclusions

The main task of this paper was to present a new approach in document classification techniques than documents are in Lithuanian language. This approach constructs weighted ontology for a document and later compares it to the ontology of some theme. We have shown that it is possible to achieve 96% of correct classification cases, using the combination of two different methods. We also demonstrated that using all part of speech terms increases the quality of classification.

While the word order in sentences of English is strictly defined, Lithuanian language has flexible word order system. Words can be in different places of the sentence, but the meaning will be the same. Such issue limits the use of some algorithms possible in comparison of strict sentence order texts, but gives more space for future research.

References

[1] G. Salton, C. Buckley // *Information Processing & Management* **24** (1988) 513.

[2] N.F. Noy, C.D. Hafner // *AI Magazine* **18** (1997).

[3] T. Gruber // *Knowledge Acquisition* **5** (1993) 199.

[4] J.-U. Kietz, R. Volz, A. Maedche // *International Conference on Grammar Inference (ICGI-2000), Lecture Notes in Artificial Intelligence*, LNAI, 2000.

[5] M.Kavalec, A. Maedche, V. Svatek *// SOFSEM 2004,* p. 249-256.

[6] S. Tiun, R. Abdullah, T.E. Kong // *Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing* (CICLing'01), February 2001, p. 444-453.

[7] M. Montes-y-Gómez, A. Gelbukh , A. López-López, // DEXA 2001: 491-500

[8] Z. Li, W. Keong Ng, A. Sun // *Knowledge Information Systems* **8** (2005) 438.