SLOVAK TEXT DOCUMENT CLUSTERING

Daniel ZLACKÝ, Ján STAŠ, Jozef JUHÁR, Anton ČIŽMÁR
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic,
e-mail: daniel.zlacky@tuke.sk, jan.stas@tuke.sk, jozef.juhar@tuke.sk, anton.cizmar@tuke.sk

ABSTRACT

Text document clustering is a task that organizes text documents according to their semantic similarity. This paper focused on clustering Slovak text documents from Wikipedia into specific categories using different clustering algorithms such as agglomerative hierarchical clustering, divisive hierarchical clustering, K-Means, K-Medoids and self-organizing maps. These algorithms were compared according to several term weighting schemes such as TF-IDF (Term Frequency Inverse Document Frequency), residual IDF, Okapi and others. We also used PCA (Principal Component Analysis) to illustrate the document vectors in three-dimensional space. We used purity and entropy to evaluate the clustering results. The best results were obtained by agglomerative hierarchical clustering using TF-IDF as a term weighting scheme.

Keywords: *hierarchical clustering, K-Means, PCA, SOM, TF-IDF, VSM*

1. INTRODUCTION

The number of documents, articles and books on the Internet is growing every day. These have brought several challenges and problems for the effective search and organization of this text data.

Clustering is a useful technique, which organizes object into smaller groups. Text document clustering groups similar documents that to form a coherent cluster, while documents that are different have separated apart into different clusters [1]. This property can be used in many various areas, where we want to organize and sort objects. In digital libraries or online stores are objects sort into several categories. If we are interested in fantasy literature, we can browse books only from this specific domain not from the all categories.

Clustering can also improve a language model for large vocabulary continuous speech recognition system in Slovak language, because we can construct domain-specific corpora from already collected and prepared Slovak text data [2] [3].

This paper is organized as follows. Section 2 will describe how we can represent text documents, several weighting schemes, similarity measures and dimension reduction technique PCA that can be used in this area. Several basic clustering algorithms will be presented in Section 3. Experiment settings, evaluation, results and analysis will be explained in Section 4.

2. TEXT DOCUMENT REPRESENTATION

Text documents are represented in the vector space model (VSM) by term vectors [4]:

$$D = (t_i, t_j, \ldots t_p) \tag{1}$$

where each t_k identifies a content term assigned to the document D. The first step in creating VSM is the text normalization. We need to remove all special mark-up tags, formatting, punctuation and the remaining text is parsing.

We can reduce the dimensionality of VSM by removing the stop words. There are function words (prepositions, conjunctions) and the others words with non-descriptive character for the topic of a document.

Remaining words can be stemmed by removing the different ending in one word. For example words like "*kamerou*", "*kamerovali*", "*kamerovými*" will be mapped to a single word "*kamer*".

Term weighting is the last step in the creating the VSM. There are many weighting scheme, which are suitable in our task. Term frequency (TF) model measures the frequency of occurrence of the terms in the document:

$$tf_{t,d} = \sum_{x \in d} f_t(x), \tag{2}$$

where $f_t(x)$ will be 1, if term t occurs in the document d. A relevant measure cannot only take TF into account, but a new collection-dependent factor must be introduced [4]. Inverse document frequency (IDF) performs this function. IDF is computed as follows [5]:

$$idf_{t,d} = \log\left(\frac{|D|}{df_t}\right), \tag{3}$$

where $|D|$ is the number of documents in our collection and df_t is the number of documents, in which term t appears. The combination of TF and IDF scheme give us one of the most widely used weighting schemes, which is defined as a product between TF and IDF model. The weight of a term j in a document i is defined as [5]:

$$w_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times \log\left(\frac{|D|}{df_t}\right). \tag{4}$$

In the Table 1 are illustrated several others term weighting schemes that can be used in the creating the VSM. ATC uses maximum term frequency in the vector (*max_tf*) to damp the original TF factor. Entropy weighting implied by (1 - ε_i) reflects the fact that two words appearing with the same count in the document d do not necessarily convey the same amount of information about the document [6]. LTU and Okapi utilize the document length (n_d) and average document length (avg_n_d) in their equations. Residual IDF is defined as the difference between the logs of actual IDF and IDF predicted by Poisson distribution [7].

Table 1 Term weighting schemes

ATC	$\frac{\left(0.5+0.5\frac{tf_{t,d}}{\max tf}\right)\log\left(\frac{N}{df_t}\right)}{\sqrt{\sum_{t=1}^{N}\left[\left(0.5+0.5\frac{tf_{t,d}}{\max tf}\right)\log\left(\frac{N}{df_t}\right)\right]^2}}$
ENTROPY	$(1-\varepsilon_t)\frac{tf_{t,d}}{n_d}$ $\varepsilon_t = -\frac{1}{\log N}\sum_{d=1}^{N}\frac{tf_{t,d}}{\sum_d tf_{t,d}}\log\frac{tf_{t,d}}{\sum_d tf_{t,d}}$
LTU	$\frac{(\log(tf_{t,d})+1)\log\left(\frac{N}{df_t}\right)}{0.8+0.2\frac{n_d}{avg\_n_d}}$
OKAPI	$\left(\frac{tf_{t,d}}{0.5+1.5\frac{n_d}{avg\_n_d}+tf_{t,d}}\right)\log\left(\frac{N-df_t+0.5}{tf_{t,d}}\right)$
RIDF	$IDF-\log\frac{1}{1-p(k,\lambda_i)}$

2.1. Principal component analysis

PCA is a linear orthogonal transformation and dimension reduction technique, which maps a large number of variables into a several principal components. PCA can be used as a transformation technique in speech processing, which is clearly described in [8]. It can also help us to analyze the text document clustering results.

2.2. Similarity measure in VSM

There exist a lot of similarity measures, which can be used in computing the similarity between the documents such as Euclidean distance, Cosine Similarity or Jaccard coefficient. In [1] author compared several similarity measures in text document clustering. Pearson correlation coefficient and the averaged Kullback-Leibler divergence were better than others. Pearson correlation coefficient is defined:

$$SIM_P(\vec{t}_a,\vec{t}_b)=\frac{m\sum_{t=1}^{m}w_{t,a}\times w_{t,b}-TF_a\times TF_b}{\sqrt{\left[m\sum_{t=1}^{m}w_{t,a}^2-TF_a^2\right]\left[m\sum_{t=1}^{m}w_{t,b}^2-TF_b^2\right]}} \quad (5)$$

where $TF_a=\sum_{t=1}^{m}w_{t,a}$ and $TF_b=\sum_{t=1}^{m}w_{t,b}$.

3. CLUSTERING ALGORITHMS

This section focuses on clustering methods and explains several basic algorithms. Clustering is a technique, which classify a set of objects into groups or clusters based on their similarity. The goal is to create clusters that are coherent internally, but clearly different from each other. Documents in the same cluster should be as similar as possible to each other and the cluster should be as dissimilar as possible to another cluster [7].

3.1. Hierarchical clustering

Hierarchical clustering creates a hierarchy of clusters and it is based on the tree diagram (dendogram). It can be also classified into a two basic types. Agglomerative clustering provides a "*bottom up*" clustering. Each object is in its own cluster at the beginning. It merged the most similar pairs of clusters and ends, when all objects are in the one cluster. Divisive technique provides a "*top down*" clustering. It starts with each object in the one cluster and splits them to smaller clusters based on their similarity. In the Fig. 1 is illustrated an example of the dendogram, where 150 objects were assigned to the 3 clusters.

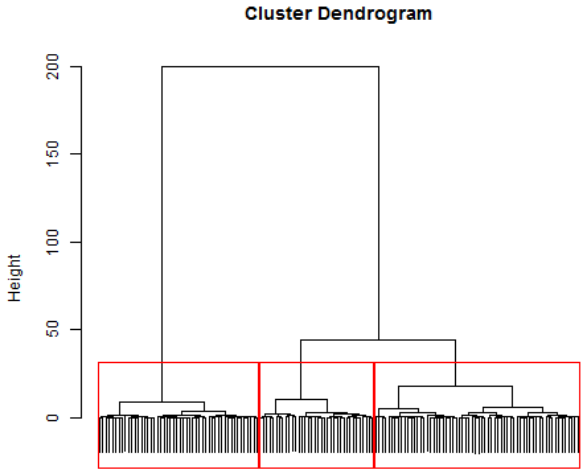

Fig. 1 The results of the hierarchical clustering

3.2. Flat clustering

Non-hierarchical clustering algorithms consist of a certain number of clusters and the relation between them is often undetermined. Most of these algorithms are iterative. They provide a reallocation operation that reassigns objects [7].

K-means is one of the simplest iterative clustering methods. It works in these steps [9]:

- Choose k objects as the initial cluster centers
- Repeat
  - (Re)assign each object to the cluster based on the given similarity function
  - Update the centroid
- Until no change

It works well, when we are processing large data sets and it is very easy to implement, but it is sensitive to outliers.

K-medoids clustering algorithm uses medoids to represent the clusters. A medoid is the most centrally located object in the cluster [9]. K-medoids algorithm is more robust than K-means, it is less sensitive to outliers, but the implementation is much more complicated and is suitable, when we are processing smaller data sets.

3.3. Self-organizing map

Self-organizing maps (SOM) or Kohonen maps are neural networks that can be used in text document clustering. It is unsupervised technique, which maps multidimensional space into lower dimensional space – map and it is described in [10]. There exists several extended version of SOM such as GH-SOM (Growing Hierarchical Self-Organizing Map) or LabelSOM, which are described in [11] and [12].

3.4. Clustering evaluation

Purity and entropy are basic evaluation measures, which are used to evaluate the quality of different clustering techniques. Purity is defined as the proportion between the numbers of documents from a single category to the total number of documents in the given cluster. The purity of a cluster C_r is defined [7]:

$$P(C_r) = \frac{1}{n_r} \max_i n_r^i, \tag{6}$$

where $\max_i(n_r^i)$ is the number of documents from the dominant category and n_r is the size of the particular cluster. The purity of the entire clustering solution is the weighted sum of the individual cluster purities [7]:

$$purity = \sum_{r=1}^{k} \frac{n_r}{n} P(C_r). \tag{7}$$

A perfect clustering solution is, when the cluster contains documents only from the single category with purity close to 1.

Entropy evaluates how different documents are distributed within each cluster. The entropy of cluster C_r is defined [7]:

$$E(C_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r}, \tag{8}$$

where q is the number of categories in the corpora and n_r^i is the number of documents from i-th class that were assigned to the r-th cluster. The entropy of the entire clustering solution is the weighted sum of the individual cluster entropies [7]:

$$entropy = \sum_{r=1}^{k} \frac{n_r}{n} E(C_r). \tag{9}$$

The clustering solution is better, when the entropy values are close or equal to 0.

4. RESULTS

Our database consists from 935 Slovak articles gathered from Wikipedia, which were manually sorted into 8 categories. These documents were stemmed using unsupervised morpheme segmentation with Morfessor [13]. The list of 648 Slovak stop words were created, which were removed from these documents. The statistics of the created database are illustrated in the Table 2. As we can see, the database consists from more than 64 thousand unique words. The particular VSMs were not created from all of these features. Words with the collection frequency less than 4 were not used in the creating VSMs. We used Pearson correlation coefficient as a similarity measure. The most documents were from Music category and the least from Cars category.

Table 2 Clustering results for K-medoids and ATC

Category	Number of documents	Number of unique words
Car	59	4 069
Film	81	6 889
Hockey	119	7 529
Music	76	3 085
Medicine	100	7 109
PC	113	10 572
Law	280	16 690
Space	107	8 579
Total	935	64 522 (23 666 with $df_t > 3$)

The graphical illustration of 935 vectors with 23 666 features is very complicated. We used PCA to transform this high dimensional space into few PCA components. In the Fig. 2, 935 documents are projected into three-dimensional space using the first three PCA components. We can suppose that categories such as films (red dots in Fig. 2), space (grey) and hockey (green) will be different enough. On the other hand, in categories such as films and music, (dark blue) and cars (black), medicine (light blue), computers (purple) and law (yellow) can be several mistakes in categorization, because their distribution in

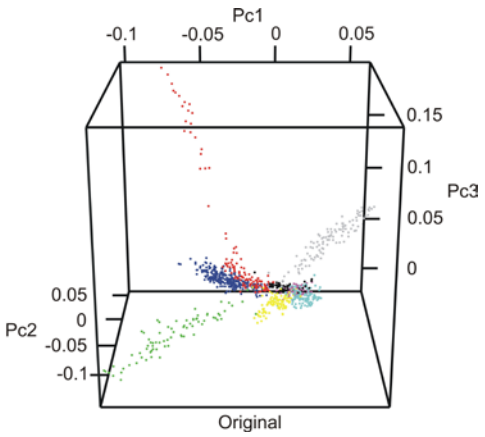

Fig. 2 Document distribution in the three-dimensional space

Fig. 2 is very close to each other and several points are overlapping.

In the Table 3 are projected clustering results for K-medoids clustering algorithms and ATC weighting scheme. For example Cluster 4 contains 18 documents from Film category, 2 from Hockey, 246 from Music and one from Medicine. As we supposed, the most mistakes in categorization are in Cluster 2, where 75 documents is from Film and 15 from Music or in Cluster 6, where 79 documents is from PC and 8 from Medicine category.

In the Table 4 are illustrated purity and entropy results for all tested clustering algorithms and weighting schemes. As we can see, the most consistently results were obtained by K-means clustering with entropy from 0.043 to 0.061. The worst results were gathered by SOM with entropy from 0.354 to 0.768. The best results were

obtained by agglomerative hierarchical clustering with TF-IDF weighting scheme with entropy 0.017.

Table 3 Clustering results for K-medoids and ATC

Cl.	Car	Film	Hockey	Music	Med.	PC	Law	Space
1	57	6	0	6	0	1	0	1
2	0	75	0	15	1	0	1	0
3	0	7	74	6	0	0	0	1
4	0	18	2	246	1	0	0	0
5	0	4	0	1	87	0	2	0
6	0	4	0	1	8	79	2	2
7	0	4	0	5	8	0	95	0
8	2	1	0	0	8	1	0	103

Table 4 Purity and entropy results

Purity					
Method	*k-means*	*k-medoids*	*hc_aglo*	*hc_div*	*som*
ATC	**0.979**	0.872	0.920	0.894	0.542
Entropy	**0.981**	0.810	0.870	0.961	0.626
LTU	**0.979**	0.902	0.932	0.956	0.568
Okapi	**0.972**	0.875	0.419	0.707	0.299
TF-IDF	0.982	0.817	**0.994**	0.969	0.600
TF-RIDF	0.982	0.817	**0.994**	0.966	0.630
Entropy					
ATC	**0.053**	0.233	0.088	0.162	0.481
Entropy	**0.047**	0.292	0.139	0.079	0.354
LTU	**0.046**	0.190	0.063	0.091	0.439
Okapi	**0.061**	0.217	0.615	0.303	0.768
TF-IDF	0.043	0.239	**0.017**	0.064	0.403
TF-RIDF	0.043	0.247	**0.018**	0.073	0.387

5. CONCLUSION

This paper focused on analysis of several basic clustering algorithms and weightings schemes in clustering Slovak text documents from Wikipedia. The best clustering results were achieved by the agglomerative clustering algorithm with TF-IDF weighting scheme. Our future work will focus on clustering large text corpora in Slovak. We want to improve a language model for large vocabulary continuous speech recognition system in Slovak language by constructing domain-specific corpora.

ACKNOWLEDGMENTS

The research presented in this paper was partially supported by the Research and Development Operational Program funded by the ERDF under the projects ITMS-26220220141 (50%) and ITMS-26220220155 (50%).

REFERENCES

[1] HUANG, A.: Similarity measures for text document clustering. In: *Proc. of the 6th New Zealand Computer Science Research Student Conference*, Christchurch, New Zealand (2008), 49-56.

[2] JUHÁR, J. – STAŠ, J. – HLÁDEK, D.: Recent progress in development language model for Slovak large vocabulary continuous speech recognition. *New Technologies – Trends, Innovations and Research*, C. Volosencu (Ed.), InTech Open Access, Rijeka, Croatia (2012), 261-276.

[3] HLÁDEK, D. – STAŠ, J.: Text mining and processing for corpora creation in Slovak language. *Journal of Computer Science and Control Systems*, Vol. 3, No. 1 (2010), 65-68.

[4] SALTON, G. – BURCKLEY, Ch.: Term weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol. 24, No. 5 (1988), 513-523.

[5] LEE, D. L. – CHUANG, H. – SEAMONS K. E.: Document ranking and the vector-space model. *Software, IEEE*, vol. 14, no. 2 (1997), 67-75.

[6] BELLEGARDA, J.: Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, vol. 88, no. 8 (2002), 1279–1296.

[7] MANNING, Ch. D. – SCHUTZE, H.: *Foundations of statistical natural language processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[8] JUHÁR, J. – VISZLAY, P.: Linear feature transformations in Slovak phoneme-based continuous speech recognition. *Modern Speech Recognition Approaches with Case Studies*. InTech, Rijeka, Croatia (2012), 131-154.

[9] SINGH, S. S. – CHAUHAN, N. C.: K-means v/s K-medoids: A comparative study. *National Conference on Recent Trends in Engineering & Technology*, Gujarat, India, (2011).

[10] KOHONEN, T.: The self-organizing map. In: *Neurocomputing*, vol. 21, issues 1-3., (1988), 1-6.

[11] MERKL, D. – RAUBER, A. – DIESSNER, A.: Uncovering the hierarchical structure of text archives by using an unsupervised neural network with adaptive architecture. *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, London (2000), 384-395.

[12] RAUBER, A.: LabelSOM: On the labeling of self-organizing maps. In *Proc. International Joint Conference on Neural Networks*, Washington (1999), 1-6.

[13] STAŠ, J. – HLÁDEK, D. – JUHÁR, J. – ZLACKÝ, D.: Analysis of morph-based language modeling and speech recognition in Slovak. *Advances in Electrical and Electronic Engineering*, vol. 10, no. 4, (2012), 291 – 296.

Received June 2, 2013, accepted June 23, 2013

BIOGRAPHIES

Daniel ZLACKÝ was born in Poprad, Slovakia in 1988. He received his M.Sc. (Ing.) degree in the field of Telecommunications in 2012 at the Department of electronics and multimedia communications of the Faculty of electrical engineering and informatics at the Technical university of Košice. He is currently PhD student at the Department of electronics and multimedia communications at the Technical university of Košice. His research interests include automatic word segmentation and language modeling in large vocabulary continuous speech recognition (LVCSR) systems.

Ján STAŠ was born in Bardejov, Slovakia in 1984. In 2007 he graduated M.Sc. (Ing.) at the Department of electronics and multimedia communications of the Faculty of electrical engineering and informatics at the Technical university of Košice. He received his Ph.D. degree in 2011 at the same department in the field of Telecommunications. He is currently working as a post-doctoral researcher at the Department of electronics and multimedia communications at the Technical university of Košice. His research interests include natural language processing, computational linguistics and statistical language modeling in large vocabulary continuous speech recognition (LVCSR) systems.

Jozef JUHÁR was born in Poproč, Slovakia in 1956. He graduated from the Technical university of Košice in 1980. He received Ph.D. degree in Radioelectronics from Technical university of Košice in 1991, where he works as a full professor at the Department of electronics and multimedia communications. He is author and co-author of more than 200 scientific papers. His research interest includes digital speech and audio processing, speech/speaker identification, speech synthesis, development in spoken dialogue and speech recognition systems in telecommunication networks.

Anton ČIŽMÁR was born in Michalovce, Slovakia in 1956. He graduated from the Slovak technical university in Bratislava in 1980, at the Department of telecommunications. He received his Ph.D. degree in Radioelectronics from the Technical university of Kosice in 1986, where he works as a full professor at the Department of electronics and multimedia communications and a rector of the Technical university of Košice. He is author and co-author of more than 170 scientific papers. His scientific research areas are broadband information and telecommunication technologies, multimedia systems, telecommunication networks and services, NGN mobile communication systems and localization algorithms.

ISSN 1335-8243 (print) © 2013 FEI TUKE
www.aei.tuke.sk

ISSN 1338-3957 (online)
www.versita.com/aei