Representation of hypertext documents based on terms, links and text compressibility

Julian Szymański[1] and Włodzisław Duch[2,3]

[1] Department of Computer Systems Architecture, Gdańsk University of Technology, Poland,
julian.szymanski@eti.pg.gda.pl
[2] Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
[3] School of Computer Engineering, Nanyang Technological University, Singapore
Google: Duch W

Abstract. Three methods for representation of hypertext based on links, terms and text compressibility have been compared to check their usefulness in document classification. Documents for classification have been selected from the Wikipedia articles taken from five distinct categories. For each representation dimensionality reduction by Principal Component Analysis has been performed, providing rough visual presentation of the data. Compression-based feature space representation needed about 5 times less PCA vectors than the term or link-based representations to reach 90% cumulative variance, giving comparable results of classification by Support Vector Machines.

1 Introduction

WWW can be seen as a very large repository of documents that changes in time and constantly grows. The challenge is to organize Internet documents automatically. Categorization (supervised or unsupervised) strongly depends on the methods used to represent text and for many hypertext documents not only the words, but also the links between the documents have been found useful to determine a text category. The best example of such organization is given by Wikipedia, which is ideal for testing link and term-based methods of text representation. Successful validation of information retrieval algorithms on the Wikipedia articles should lead to improvements of information retrieval in the Internet, for example by assigning information to categories found in the Wikipedia. Although the current manually-made system of Wikipedia categories is not perfect it can be used for evaluation of methods based on various text representations. An important advantage of Wikipedia comes from the fact that the data is available for download as semi-structured SQL files and XML dumps[4,5]. The experiments presented in this article have been performed on the Wikipedia in simple English version[6], reducing the data to the most popular articles only.

In recent years significant progress in machine learning methods brought a wide spectrum of techniques for data analysis, especially clustering and classification. These

[4] http://en.wikipedia.org/wiki/Wikipedia_database

[5] http://download.wikimedia.org/

[6] http://simple.wikipedia.org/wiki/Main_Page

Lecture Notes in Computer Science Vol. 6443, pp. 282–290, 2010.

algorithms represent objects (such as documents) using feature vectors, or creating kernel features that are based on similarity between the objects. Even the best machine learning algorithms without appropriate representation of objects will fail. The aim of the experiments presented here is to find hypertext representation suitable for automatic categorization. Three methods of text representation have been studied: bag-of-words based on terms, the use of links between documents, and estimation of similarity between documents based on their compressibility. Cumulative percentage of variance captured by the most important PCA components [1] already tells us a lot about the quality of representation, and also allow to made rough view of the data in 2D. SVM classification has been performed in the three feature spaces before and after PCA reduction.

2 Text representation

Humans understand text using a lot of background knowledge; spreading activation processes in the brain invoke additional concepts through automatic (usually shallow) inferences. This process may be partially captured in simple algorithms provided with the help of large ontologies or semantic networks [2]. Here three approaches to text representation that do not use *a priori* knowledge are presented.

In Information Retrieval text is typically represented by the so-called **B**ag **o**f **W**ords (BoW), using frequencies of words as features. The lack of word order and simple grammatical constructions is a sever limitation of such representation. There are several methods that try to deal with that problem. First, features may include collocations and frequent phrases. Second, features may be constructed from statistical analysis of co-occurencess of successive words using n-grams [3]. Disadvantage of n-grams approach is that it produces very high dimensional feature spaces and requires large training sets. Dimensionality reduction based on PCA may automatically discover some phrases important for document categorization. The Latent Semantic Analysis (LSA) [4] and newer spectral methods work in such reduced spaces, automatically discovering useful combinations of words that contributes to document categorization.

The words that appear in a text have different inflections and require pre-processing to avoid redundant features. Stemming maps words that have the same root (stem) but different inflections on their basic forms (ex: living, lives → live). Words that appear frequently in all texts are removed using stop-words list[7].

2.1 Terms

The preprocessed words are called terms and in the BoW text representation they are used as features. The value, or descriptiveness of a term for a given document may be estimated by the strength w of association between the term and the text. Typically for n-th term and k-th document w value is calculated as a product of two factors: term frequency tf and inverse document frequency idf, given by $w_{k,n} = tf_{k,n} \cdot idf_n$. The term frequency is computed as the number of its occurrences in the document and

[7] http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/

is divided by the total number of terms in the document. The frequency of a term in a text determines its importance for document content description. If a term appears in the document frequently, it is considered as more important. The inverse document frequency increase the weight of terms that occur in a small number of documents. The idf_n factor describes the importance of the term for distinguishing documents from each other and is defined as $idf_n = \log(k/k_{term(n)})$, where k is the total number of documents, and $k_{term(n)}$ denotes the number of documents that contain term n.

Features (terms) and weights w that associate them with the collection of documents allows to represent each document by a single point in the Vector Space Model (VSM) [5]. Document similarity is then easily computed using different distance measures such as cosine or euclidean measures [6].

2.2 Links

Representations of texts based on terms lead to high-dimensional feature spaces (compare the size of feature spaces in Table 2). Without preprocessing the number of features would be equal to the total number of the distinct words that appear in all documents. Another more compact way to create numerical representation of texts for evaluation of document similarity is based on references that appear between documents. For articles and books the list of references and bibliographical notes about their authors contain useful information. If hypertext documents are considered their hyperlinks can be used as additional features. This is particularly useful in Wikipedia and in scientific articles, where the number of references is relatively large.

Feature space based on links and shared references may be constructed in several ways. Each link provides a new dimension and the simplest document representation creates a binary vector, where 1 denotes the presence of the link (reference) to another document, and 0 means that there is no link. Documents on similar topics tend to link to similar set of other documents and cite the same references. Possible extensions of this representation involve frequency of references, various forms of weighting, the use of directed links (± 1 for links from or to the document) and personal names that serve as links. These modifications haven't been considered here, only binary representations of articles have been used below.

2.3 Compression

The third approach to the representation of text documents is based on algorithmic information [7]. If two documents are similar their concatenation will not lead to a significant increase of algorithmic complexity. The measure of algorithmic information contained in the text may be estimated using standard file compression techniques. If two text files are quite different compressed concatenated file will have the size approximately equal to the sum of sizes of the two files compressed separately. If the two files are similar compressed concatenated file will be only slightly larger than the size of a single compressed file. To express the complexity-based similarity measure as a fraction by which the sum of the separately compressed files exceeds the size of the jointly compressed file the following formula is used:

$$sim_{A,B} = 2\left(1 - \frac{size(A+B)_p}{size(A)_p + size(B)_p}\right) \tag{1}$$

where A and B denote text files, and the suffix p denotes the compression operation. This is a good measure of similarity that implicitly takes into account strings of letters, collocations and longer phrases that are used to form a dictionary by the compression algorithm. Each document D is thus represented by a vector with components $V(D)_i = sim_{D,D_i}$, therefore the dimensionality is equal to the total number of documents. Preprocessing in this case is restricted to stop list only, as most compression algorithms can handle word morphology themselves. The book on Kolmogorov algorithmic complexity [7] shows many applications of similarity based on such measures.

3 The Data - evaluation dataset

The three ways to generate numerical representation of texts have been compared on a set of articles selected from the Wikipedia. These articles belong to five different subcategories of the Wikipedia supercategory "Science" ↪[8]: Chemistry ↪ **Chemical compounds**, Biology ↪ **Trees**, Mathematics ↪ **Algebra**, Computer science ↪ **MS** (Microsoft) **operating systems**, Geology ↪ **Volcanology**. Detailed information about selected documents is presented in Table 1 and Table 2. A total of 281 articles has been selected. For term-based representation only those terms that appeared in the whole collection of articles more than once (freq. $>$ 1) have been kept. Also for the link-based representation references (features) that appear only one time have been removed. Table 1 explains colors and symbols used in Figure 1 to mark particular classes.

Table 1. Category names and the number of articles used to construct data sets

Category name	Number of the articles	Color and Symbol	
Chemical compounds	115	red	∗
Trees	69	green	+
Algebra	21	blue	□
MS operating systems	19	black	·
Volcanology	57	magneta	◇

Table 2. Size of feature spaces for different representation methods

Features space size				
terms		links		complexity
raw data	freq. $>$ 1	raw data	freq. $>$ 1	
12358	3658	1817	650	281

4 Comparison of text representations

The rough view of the class distribution in different representations can be made using two principal components with the highest variance [1]. This is shown in Figure 1. It

[8] ↪ denotes hierarchical relation

is clear that two PCA components are not sufficient to separate all data, although most documents from the "tree" category may be distinguished quite easily. This is also clear from analysis of eigenvalues showing that the first two eigenvectors capture only a small percentage of variance.

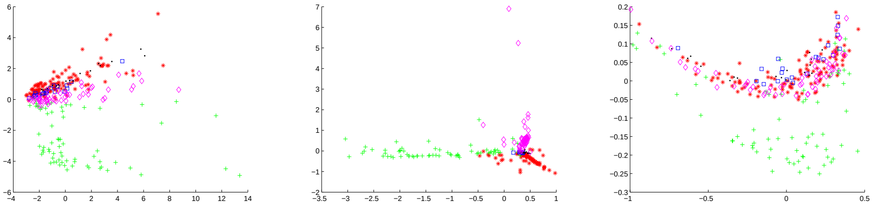

Fig. 1. Projection of dataset on two highest principal components for text representation based on terms, links and compression

Using other combinations of principal components for scatterograms or using multidimensional scaling more structure can be observed, indicating that different methods of representation extract different information from texts. To see how much information is lost by performing PCA dimensionality reduction, in Figure 2 cumulative sums of the percentage of variance captured by the most important components for different representation methods is presented. For terms and links more than 100 components are needed to capture 80% of variance, but for algorithmic complexity very few components are needed, and 90% of variance is accounted for using only 36 components, about 5 times less than for terms or links (Fig. 2 where 163 and 154 components were needed.

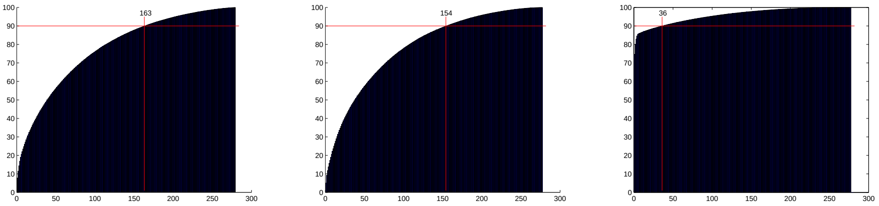

Fig. 2. Cumulative sum of primary components variance for different text representations: terms, links and algorithmic complexity

Information extracted by different text representations may be estimated by comparing classifier errors in various feature spaces. Support Vector Machines classifier [8] have proved to be suitable for text categorization [9], [10]. To perform multiclass classification with SVM one-versus-other class approach has been used with two-fold crossvalidation repeated 50 times for accurate averaging of the results.

The detailed results of calculations are presented in Table 3. Standard deviations in all these calculations did not exceed 3%, and to limit the size of the tables are not reported here. Feature spaces have been generated using term, link and complexity-based document representations, and the results in the first column ("raw") are obtained by using linear SVM directly in these spaces. The second column, marked in Table 3 as $f. > 1$, shows results in reduced feature spaces, after removal of features that appear only with relation to a single document. In almost all cases this leads to improvement, sometimes quite significant. Dimensionality of the original and reduced spaces is given in Table 2.

The next two columns contain results in the kernel spaces obtained by linear SVM. Instead of the original vectors $\mathbf{X}$ kernel features $z_i(\mathbf{X}) = \{K(\mathbf{X}, \mathbf{X}_i)\}$ are generated using Euclidean distance $K(\mathbf{X}, \mathbf{X}_i) = ||\mathbf{X} - \mathbf{X}_i||$ and cosine distance $K(\mathbf{X}, \mathbf{X}_i) = \mathbf{X} \cdot \mathbf{X}_i / ||\mathbf{X}|| ||\mathbf{X}_i||$ as kernels. These kernel spaces have dimension equal to the number of all documents, in our experiments equal to 281, thus much smaller than the original feature spaces. We have used explicit representation of these kernel spaces with linear SVM instead of explicitly kernelized version of SVM because results of both approaches are essentially equivalent [11], but analysis of the discriminant functions is greatly simplified. For large number of documents selection of redundant kernel features by simple filters may reduce dimensionality in a similar manner to selection of support vectors.

Significant improvements of classification accuracy have been obtained in these kernel spaces. For terms and links Euclidean and cosine kernels replace original features by distances to all training data. Although our database is relatively small this is quite beneficial and should lead to even better results for larger sets of documents. Surprisingly, also for representation based on algorithmic complexity clear improvement in accuracy is noted, although this space is already based on similarity estimated using compression techniques. Transforming data in this representation by distance-type kernels amounts to second-order similarity transformation [12].

Table 3. Evaluation of the classification with SVM for different text representations.

Category name	Text representation										
	terms				links				complexity		
	raw data	f.>1	cos	euclid	raw data	f.>1	cos	euclid	raw data	cos	euclid
Chemical compounds	87.2	93.7	98.9	97.2	85.3	86.2	97.7	95.1	91.1	96.8	95.9
Trees	90.4	92.9	98.7	96.1	87.9	92.1	96.8	95.5	92.7	98.5	95.3
Algebra	94.1	98.8	99.3	97.8	88.2	91.9	98.7	97.6	95.9	94.9	93.5
MS operating systems	98.6	98.6	99.9	98.6	97.4	98.6	99.7	99.6	99.3	99.7	98.9
Volcanology	94.5	94.3	98.8	98.2	94.4	95.7	98.8	96.2	94.5	97.8	95.6
Overall	92.9	95.6	99.0	97.5	90.6	92.9	98.3	96.8	94.7	97.5	95.8

In the second set of experiments (Table 4) dimensionality of representation space has been reduced even further by taking only the most important PCA components that cover 90% of the variance in the data. For the term and link representation that leads to some loss of accuracy, while the complexity based representation, with quite small (36) number of dimensions has not been degraded at all. Transforming 163 PCA vectors for term representation using cosine or Euclidean kernel recovered all information in this space, giving an insignificant improvement of the results. However, for links PCA reduction leads to decrease of classification accuracy by 3% for cosine kernel.

Table 4. Evaluation of the classification with SVM for different text representations, scaled with PCA.

Category name	Text representation scaled with PCA								
	terms			links			complexity		
	PCA=163	cos	euclid	PCA=154	cos	euclid	PCA=36	cos	euclid
Chemical compounds	77.4	98.8	97.3	75.3	94.4	96.2	94.1	96.7	95.2
Trees	82.2	98.4	97.6	79.9	88.9	93.3	90.7	97.7	96.6
Algebra	96.8	99.8	97.8	92.8	97.8	97.6	96.7	96.5	95.4
MS operating systems	96.9	99.9	96.9	97.1	98.3	97.8	98.8	99.3	98.9
Volcanology	86.2	98.9	98.7	86.8	97.1	96.8	93.4	98.1	95.6
Overall	87.9	99.1	97.6	86.3	95.3	96.3	94.7	97.6	96.3

5 Discussion and future plans

Reading or listing to words neural activation in the brain spreads invoking additional concepts that support understanding and categorization of documents [2]. One should not expect perfect categorization without approximation of such processes with the help of extensive background knowledge and at least shallow inferences. However, it is important to know what kind of knowledge is most important and how to create useful features that would capture important information allowing for text categorization. In this paper we have compared three approaches to text representation, based on terms, links and similarity of their algorithmic complexity. Complexity measure allowed for much more compact representation, as seen from the cumulative contribution of principal components in Fig. 2 and achieved best accuracy in PCA-reduced space with only 36 dimensions, Tab. 4. However, after using cosine kernel term based representation is slightly more accurate. Explicit representation of kernel spaces and the use of linear SVM classifier allows to find important reference documents for a given category, as well as identify collocations and phrases that are important for characterization of each category.

Experiments presented here should be treated as a test-bed for large scale application of our methods for text categorization. The selection of Wikipedia articles from very different subcategories of articles in the supercategory "Science" used here for computational experiments made classification tasks perhaps too easy, as is evident from very high accuracy obtained by the two-fold crossvalidation. In future we plan to investigate more complex tasks, requiring hierarchical of classification, with articles more similar to each other. We can expect that for such more complex tasks the differences in usage of the text representations would be even larger, with more significant advantages coming from kernelization of feature spaces. We plan to run experiments on a much larger scale, on the whole Wikipedia, but this requires parallelisation of algorithms to run them on a powerful cluster instead of on single PC. We also plan to run unsupervised methods for clustering Wikipedia articles and provide tools to automatically create categories for this largest repository of human knowledge.

Different methods of text representation may be combined before such kernelization, and although we have not shown it here, combining term, link and complexity-based representations, followed by kernelization and aggregation of features using PCA leads to even better results with quite small feature spaces. Another idea to introduce more background knowledge and capture some semantics is to map articles on activations of a semantic network and then calculate distances between them. WordNet dictionary [13] may be used for this purpose with word disambiguation techniques [14] that allow to map words to their proper synsets. We have made some research in this direction and the first results are very promising (in preparation). Representation methods based on neurolinguistic inspirations [2] that use natural concept semantics will also be investigated.

Acknowledgment This work was supported by the Polish Ministry of Higher Education under research grant no. N N519 432 338.

References

1. Jolliffe, I.: Principal component analysis. Springer (2002)
2. Duch, W., Matykiewicz, P., Pestian, J.: Neurolinguistic approach to natural language processing with applications to medical text analysis. Neural Networks **21(10)** (2008) 1500–1510
3. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science **267**(5199) (1995) 843
4. Landauer, T., Dumais, S.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. Psychological Review **104**(2) (1997) 211–240
5. Wong, S.K.M., Ziarko, W., Wong, P.N.: Generalized vector spaces model in information retrieval. In: SIGIR '85, New York, NY, USA, ACM Press (1985) 18–25
6. Korenius, T., Laurikkala, J., Juhola, M.: On principal component analysis, cosine and Euclidean measures in information retrieval. Information Sciences **177**(22) (2007) 4893–4905
7. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications. Springer (3rd ed, 2008)
8. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2001)
9. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Machine Learning: ECML-98 (1998) 137–142
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) **34**(1) (2002) 1–47
11. Maszczyk, T., Duch, W.: Support feature machines: Support vectors are not enough. In: World Congress on Computational Intelligence, IEEE Press (2010) 3852–3859
12. Duch, W.: Towards comprehensive foundations of computational intelligence. In Duch, W., Mandziuk, J., eds.: Challenges for Computational Intelligence. Volume 63. Springer (2007) 261–316
13. Miller, G.A., Beckitch, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. Cognitive Science Laboratory, Princeton University Press (1993)
14. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval. In: 16th ACM SIGIR Conference, ACM (1993) 171–180