Towards Deeper Understanding of the LSA Performance

Preslav Nakov[1], Elena Valchanova[2], Galia Angelova[2]

(1) University of California at Berkeley, EECS, Berkeley CA 94720, USA
(2) Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences,
25 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
nakov@eecs.berkeley.edu, {elenav, galia}@lml.bas.bg

Abstract

The paper presents on-going work towards deeper understanding of the factors influencing the performance of the Latent Semantic Analysis (LSA). Unlike previous attempts that concentrate on problems such as matrix elements weighting, space dimensionality selection, similarity measure etc., we primarily study the impact of another, often neglected, but fundamental element of LSA (and of any text processing technique): the definition of "word". For the purpose, a balanced corpus of Bulgarian newspaper texts was carefully created, to allow for in-depth observations of the LSA performance, and series of experiments were performed in order to understand and compare (with respect to the task of text categorisation) six possible inputs with different level of linguistic quality, including: graphemic form as met in the text, stem, lemma, phrase, lemma&phrase and part-of-speech annotation. In addition to LSA, we made comparisons to the standard vector-space model, without any dimensionality reduction. The results show that while the linguistic processing has a substantial influence on the LSA performance, the traditional factors are even more important, and therefore we did not prove that the linguistic pre-processing substantially improves text categorisation.

1 Introduction

The contemporary statistical text data processing relies almost exclusively on bag-of-words models and individual words counts. Multiword terms are less often used as basic language elements due to the complexity of their recognition. Thus, the definition of *word*/*term* from the point of view of the particular algorithm may turn to be of crucial importance. It can be just the surface *word type* (*form*) or *word token* as seen in the text (possibly converted to lowercase), the *lemma* (the canonical form) after inflexions removal, the *root* or the *stem* (a prefix shared by the different forms of the same word). In the latter case a stem can group together a set of inflected forms only (of the same root), but often includes derivational variants as well. In addition, the homographs can be further disambiguated: this can be limited to part-of-speech (POS) only, or may involve word sense disambiguation (when a supervised algorithm is used) or word sense discrimination (e.g. clustering, when unsupervised). Finally, the *terms* can be multi-word phrases or named entities.

There is a default assumption in the computational linguistics community that a better linguistically motivated definition of *word* ***may*** improve the information retrieval (IR) results, whatever *improvement* means, but this assumption remains to be proven or at least checked in carefully designed experiments. Our present work is focused on LSA in an attempt to better understand the role of the preliminary linguistic processing in the sequence of text transformations in the computation of document similarity scores.

The paper is organised as follows. Section 2 briefly introduces LSA as an IR technique. Section 3 contains related work. Section 4 presents the resources and the text collections used. Section 5 describes our experiments and section 6 discusses the results. Section 7 contains the conclusion and points some future directions of our work.

2 LSA and Text Categorisation

LSA is among the most popular techniques for indexing, retrieval and analysis of textual information during the last decade. While the classic statistical approaches look for dependencies between two words (or documents), LSA concentrates on term-document associations, considering the word-word and document-document ones as their derivatives.

LSA assumes an internal structure of the word usage that cannot be observed directly due to the freedom of lexical choice: a variety of words and word combinations can refer to the same notion (different people use the same words to describe the same object 10-20% of the time only (Furnas et al. 86)). The fundamental assumption is the existence of a set of mutual latent dependencies between the words and the contexts they are used in (phrases, paragraphs or texts). The claim is that their identification and proper treatment permit LSA to deal successfully with synonymy and partially with polysemy, which are among the major problems with the word-based approaches (Landauer et al. 98).

LSA is fully automatic and does not use any linguistic nor conceptual resources. It is a two-stage process including learning and analysis of the indexed data. When learning, LSA performs an

automatic document indexing. The process starts with the construction of a matrix X with columns associated with the *documents*, and rows – with the *terms* (words or key-phrases). The cell (i,j) contains the frequency (possibly weighted) of term i in document j. Given m terms and n documents X is an $m \times n$ matrix. Let the rank of X is r, $r \leq \min(m,n)$. The fundamental idea of LSA is to submit X to singular value decomposition (SVD), which results in three matrices: $T(m \times r)$, $D(r \times n)$ (orthonormal) and $S(r \times r)$ (diagonal), such that $X = TSD^t$. The main diagonal of S contains non-zero elements known as *singular values*, while T and D are the *left* and the *right singular matrices*. Let us rearrange the rows and columns of T, D and S so that the singular values are ordered descendingly. We remove most of the elements of S, while keeping the biggest l ($l << r$) and obtain $S_1(l \times l)$. Similarly, we remove all but the first l columns of T and all but the first l rows of D^t, which gives us $T_1(m \times l)$ and $D_1(l \times n)$. Now T_1, S_1 and D^t_1 have suitable dimensionalities to be multiplied. The matrix $X_1 = T_1 S_1 D^t_1$ is the least squares best-fit approximation of X in the new l-dimensional space.

This projection is supposed to remove the unnecessary noise while revealing the latent factors. It compresses the original space in a much smaller one where only a limited number of singular values are kept (typically between 100 and 400; experiments for English show 300 is optimal in general (Landauer & Dumais 97)). A vector of reduced dimensionality is associated with each term and with each document. It is possible to perform SVD in a way that the truncated matrices T_1, S_1 and D_1 are found directly (Berry et al. 93).

The second phase is the analysis. Most often this includes a study of the proximity between a couple of documents, a couple of words or between a word and a document. A simple mathematical transformation using the singular values and vectors from the training phase permits to obtain the vector for a non-indexed term or document. The proximity between two documents (or two terms) can be calculated as the dot product between their normalised LSA vectors. Other measures are also possible: Euclidean and Manhattan distances, Minkowski measures, Pearson's correlation coefficient etc. (see (Deerwester et al. 90; Landauer et al. 98; Nakov 00)).

Although LSA is a comparatively old and well-studied technique, its effective usage requires sophisticated tuning which is viewed as a kind of art. Some of the most important performance factors are:

- Definition of term;
- Matrix elements weighting;
- Space dimensionality choice;
- Similarity measure choice.

The definition of term usually attracts less attention. To the best of our knowledge, it is not studied at all for the highly inflectional Slavonic languages, which motivates our current research. Although LSA is an IR technique, it has been used for a variety of tasks, including text categorisation (Bartell et al. 92; Berry et al. 95; Foltz & Dumais 92). We found the latter more natural for automatic evaluation since it allows relatively easy results comparison and is to some extent more objective than the classic IR. There are several subtasks in text categorisation: topic identification, authorship attribution, text genre classification, language identification etc. The experiments reported below are restricted to topic identification.

We used the k-nearest-neighbour classifier, which is among the best performing text categorisation algorithms (Yang 99). The idea is to calculate a similarity score between the document to be classified and each of the labelled documents in the training set. When $k = 1$ the class of the most similar document is selected. Otherwise, the classes of the k closest documents are used, taking into account their scores. The two most popular approaches are:

1) direct sum;

2) sum after dividing the scores by the rank of the document in the sorted list of the closest k ones.

Consider $k = 6$ and let the 6 closest documents have scores/classes as follows: .98/*cls2*, .76/*cls1*, .65/*cls3*, .53/*cls1*, .47/*cls2* and .33/*cls1*. By just adding the individual scores, we obtain:

cls1: .76+.53+.33=1.62,

cls2: .98+.47=1.45 and

cls3: .65,

so *cls1* wins. If we divide by the rank, we have:

cls1:.76/2+.53/4+.33/6=.5675,

cls2: .98/1+.47/5=1.074 and

cls3: .65/3=.2167, so *cls2* wins.

In our experiments, we built a LSA matrix from the documents in the training set. The new document to be classified is projected in the LSA space and then compared to each one from the training set using cosine as a similarity measure. Then the kNN classifier for a particular value of k is used in order to predict the class. In addition to LSA, we used for comparison purposes the standard vector-space model – without any dimensionality reduction.

3 Related Work

A variety of algorithms have been applied in the past to supervised text categorisation: Naïve Bayes (Domingos & Pazzani 97), k-nearest-neighbour (kNN) (Mitchell 96), Rocchio (Rocchio 71), support

vector machines (Joachims 97), decision trees (Lewis & Ringuette 94), decision lists, neural networks (Wiener et al. 93), maximum entropy, expectation maximisation, linear least squares etc. (see (Yang 99) for an overview). For application of LSA to text categorisation see (Bartell et al. 92; Berry et al. 95; Foltz & Dumais 92).

Normally, the text categorisation algorithms comprise two steps: feature selection and classifier learning over the feature space. The first step is the critical one: once the good features are identified, any standard machine learning classification algorithm would perform (more or less) well (Scott & Matwin 99). In the case of LSA, this is highly dependent on the parameters tuning as mentioned above. See (Nakov 00) for an overview, and (Dumais 91; Spark-Jones 72; Nakov et al. 01) for a deeper study of the weight functions impact on LSA performance.

An important element of our study comprises *stemming*, which is a popular IR technique for wordforms count reduction and has proved to be beneficial for languages like French (Savoy 93), Dutch (Kraaij & Pohlmann 94), Slovene (Popovic & Willett 92), Russian and Ukrainian (Kovalenko 02) (see (Nakov 03) for an overview). Despite the contradictory evidence in the past (Harman 91), nowadays it is largely accepted that stemming improves IR, although not necessarily significantly: while Krovetz reports 30-40% (Krovetz 93) in a later evaluation Hull finds only 1-3% (Hull 96).

4 Linguistic Resources and Text Collections

A key resource for the experiments reported below is the Morphological Dictionary of Bulgarian, created at the Linguistic Modelling Department (CLPP-BAS), which contains approximately 900,000 wordforms (60,000 lemmas). A list of 442 stop-words was derived from it. The texts from the corpus were lemmatised according to this dictionary. Here we discuss in more details the related stemmer.

There are no systematic studies of the impact of stemming on IR for Bulgarian. It is an interesting problem because of the definite and indefinite articles, which appear *augmented* at the very *end* of the words. We used a rule-based inflectional stemmer for Bulgarian (Nakov 03) (the stemming rules are learned automatically from the morphological dictionary above). The stemmer uses three letter contextual rules of the form: remove *-u*, if the preceding three letters are *-ocm-* and there is at least one vowel remaining to the left (e.g. *рад**осм**и→рад**осм***).

We manually built a special collection *Set15* of news articles from Bulgarian online sources, including 702 different documents manually grouped in 15 categories: 693 documents were assigned exactly one category, 9 had two categories. We were unhappy with the latter group since it requires a multi-way classification algorithm, which would lead to unnecessary complications. So, we investigated each of the 9 documents and removed the less suitable category for each one, keeping the more likely one only. The list of categories and the number of documents for each of them is shown in Table 1.

Category	Size	%
Agriculture&Foresty	11	1.57%
Culture	33	4.70%
Defence	13	1.85%
Economy	130	18.52%
Education&Science	5	0.71%
Energy&EnergyResources	21	2.99%
Finance	209	29.77%
ForeignAffairs	60	8.55%
Health	13	1.85%
Interior	71	10.11%
Justice	25	3.56%
Labor&SocialPolicy	21	2.99%
RegDevelop&PublicWorks	8	1.14%
Sport	67	9.54%
Transport&Communications	15	2.14%
TOTAL	***702***	***100.00%***

Table 1: *Set15* – the 15 categories and their sizes.

Although the stemmer targets the inflectional morphology only, the resulting stems sometimes conflate different derivational variants as well, e.g. both *здраве* (*health*) and *здравен* (*healthy*) stem to *здрав*. This means, sometimes it is potentially more powerful than lemmatisation (although it is conservative and not as aggressive as the traditional stemmers for English, e.g. the Porter stemmer (Porter 80)). On the other hand, from a linguistic perspective, it is less correct than lemmatisation, so we wanted to compare them. In order to improve the coverage we used all the possible 93,066 rules for the three-letter left contexts (see (Nakov 03)).

Category	Size	%
Agriculture&Foresty	*12*	9.45%
Culture	33	25.98%
Defence	*15*	11.81%
Sport	67	52.76%
TOTAL	***127***	***100.00%***

Table 2: *Set4* – the 4 categories and their sizes.

In addition, we built another collection *Set4*, a subset of the original one, containing the documents from 4 categories only (see Table 2). Note that the sizes differ a little bit. This is because when only 4 categories are considered some of the two-category documents lose the redundant category, and so there is no need to remove them from the other one. Thus,

Agriculture&Foresty has one and *Defence* has two additional documents as compared to Table 1.

Set15 contains 19,429 word types and 406,783 word tokens (the numbers and the non-Cyrillic symbols are excluded). We excluded the stop-words from a predefined list containing 442 wordforms: the Bulgarian closed-class words as found in the morphological dictionary above, namely (note that some forms belong to more than one POS class) – conjunctions (31), interjunctions (17), particles (38), prepositions (68), pronouns (288) and auxiliary verbs (26). Further, we filtered out all single letter words as well as the ones met in a single document (as they cannot contribute to the similarity between two documents). As a result, the word types count dropped to 19,301 and the word tokens count – to 259,000. Similarly, when the filtering was applied to *Set4* the word types/tokens dropped from 15,483/80,016 to 5,530/36,527.

A potential problem with this stop-words removal is that many of the stop-words can be regular ones depending on their POS: e.g.

под/preposition (*under*), cf. *под*/noun (*floor*),

тези/pronoun (*these*) cf. *тези*/noun (*theses*),

бил/auxiliary (*has been)* cf. *Бил*/person (*Bill*).

There are 53 such stop-words in our dictionary. So, for the POS disambiguation, lemma and lemma&phrase experiments described below we checked the POS before filtering. For the stemming experiments no checking was performed.

In sum, *Set4* contains (types/tokens): 15,438/80,016 words, 4,487/71,879 stems extracted by the above-mentioned stemmer, 4,558/73,018 lemmas, 951/1455 phrasal terms and 190/458 named entities. Semantic processing concerned named entities only, as synonymy of institution names was relevantly marked.

5 Experiments and Evaluation

For the evaluation we used a 10-fold stratified cross-validation. For the purpose, *Set15* was split into 10 sets of almost equal size such that the class distribution in each set follows as much as possible the class distribution in the original set as given in Table 1. We ran 10 tests, each time training on 9 of them and testing on the remaining one. We then calculated the classification accuracy on the test set for each run and took the average over the 10 runs in order to obtain the cross-validation accuracy.

Set4 is smaller and we did not want to lose too much data for training, so we performed a stratified *20-fold* cross-validation. An alternative would be to follow a *leave-one-out* strategy: train on 126 documents and test on the remaining one. But since we remove a document from one class only, this class will suffer, unlike the rest, and thus the results will not be a good approximation of the real performance. We decided that 20 is a good balance between the need to model to some extent the original distribution and to waste as less documents as possible during testing.

As we will see below, the choice of weighting functions applied prior to SVD can have a dramatic impact on the further performance. The weighting can be expressed as a product of two numbers *Local* and *Global Weight Functions* (LWF and GWF). The LWF $L(i,j)$ represents the weight of term i in document j, while GWF $G(i)$ expresses the weight of term i across the entire document set. Some of the most popular weighting schemes for LSA, together with their numerical codes used in our tables, follow (Nakov et al. 01):

LWF = 0: $L(i,j) = X_{ij}$ — *term frequency*

LWF = 1: $L(i,j) = \log (1+X_{ij})$ — *logarithm*

GWF = 0: $G(i)=1$ — *trivial*

GWF = 1: $G(i) = 1/\sqrt{\sum_j L(i,j)^2}$ — *normalised*

GWF = 2: $G(i) = g(i) / d(i)$ — *GfIdf*

GWF = 3: $G(i) = 1+ \log (N / d(i))$ — *Idf*

GWF = 4: $G(i) = -\Sigma_j\ p(i,j) \log p(i,j)$

GWF=5: $G(i) = 1 + \{\Sigma_j\ p(i,j) \log p(i,j)\} / \log N$

where

N is the training documents count;

$g(i)$ the frequency of term i across all documents;

$d(i)$ is the number of documents containing i;

$p(i,j)$ is the probability (normalised frequency) of observing term i in document j.

Both GWF=4 and GWF=5 represent some kind of entropy. In our experiments, GWF=4 performed slightly worse than GWF=5 but otherwise exhibited the same behaviour across the table columns, so we removed it from the tables in order to save space.

The results of the evaluation for *Set4* using *k*NN with $k = 1$ are summarized in Table 3, which shows the classifier's micro-average accuracy over the 20 cross-validation runs. The first two columns contain the LWF and GWF numerical codes, as described above. Column 3 shows the LSA space dimensionality (*orig.* means: no dimensionality reduction, i.e. the original vector-space, which is 15,438 for raw words; 4,487 for stems etc. See the end of section 4.). In fact, each row in Table 3 corresponds to a particular combination of LWF*GWF and LSA dimensionality. The original space without any LSA

reduction is shown in bold.

Columns 4-8 and 10-14 of Table 3 correspond to experiments with stop-words *kept* and *removed* and contain the following:

- *raw words:* the words as met in the text;
- *stem:* stemmed words;
- *lemma:* lemmatised words;
- *lemma&phrase: lemma* and *phrase* together;
- *POS disam.:* disambiguation in terms of POS: e.g. distinguish between *свят/adjective* (holy) and *свят/noun* (world).

Column 9, entitled *phrase only*, contains the results when only multi-word phrases and named entities are used as features.

LWF	GWF	LSA dim.	STOP-WORDS KEPT						STOP-WORDS REMOVED				
			raw words	stem	lemma	lemma& phrase	POS disam.	phrase only	raw words	stem	lemma	lemma & phrase	POS disam.
0	0	10	78.74%	88.98%	84.25%	85.04%	69.29%	83.46%	92.13%	92.13%	96.85%	96.85%	86.61%
0	0	20	82.68%	85.83%	85.83%	85.83%	81.10%	82.68%	92.91%	98.43%	99.21%	99.21%	89.76%
0	0	40	85.04%	86.61%	86.61%	86.61%	85.83%	83.46%	96.06%	100.00%	98.43%	99.21%	92.13%
0	**0**	**orig.**	**74.80%**	**89.76%**	**85.83%**	**85.04%**	**72.44%**	**37.80%**	***96.06%***	***96.06%***	***98.43%***	***99.21%***	***91.34%***
0	1	10	76.38%	89.76%	81.10%	84.25%	70.08%	83.46%	96.85%	98.43%	96.85%	97.64%	89.76%
0	1	20	85.04%	87.40%	86.61%	86.61%	80.31%	83.46%	95.28%	99.21%	98.43%	98.43%	88.19%
0	1	40	81.10%	89.76%	85.04%	84.25%	77.17%	83.46%	95.28%	98.43%	98.43%	98.43%	93.70%
0	**1**	**orig.**	**61.42%**	**87.40%**	**85.04%**	**84.25%**	**66.93%**	**34.65%**	***96.06%***	***96.06%***	***98.43%***	***99.21%***	***91.34%***
0	2	10	55.91%	61.42%	65.35%	64.57%	56.69%	81.89%	92.13%	94.49%	93.70%	95.28%	82.68%
0	2	20	56.69%	66.14%	70.08%	70.87%	59.06%	83.46%	92.91%	95.28%	93.70%	95.28%	84.25%
0	2	40	58.27%	67.72%	70.87%	71.65%	59.84%	84.25%	92.13%	98.43%	98.43%	98.43%	86.61%
0	**2**	**orig.**	**57.48%**	**68.50%**	**72.44%**	**72.44%**	**59.84%**	**37.01%**	***93.70%***	***98.43%***	***98.43%***	***98.43%***	***86.61%***
0	3	10	95.28%	98.43%	99.21%	99.21%	89.76%	83.46%	97.64%	98.43%	99.21%	99.21%	94.49%
0	3	20	96.85%	100.00%	100.00%	100.00%	96.06%	83.46%	99.21%	99.21%	99.21%	99.21%	96.85%
0	3	40	92.91%	99.21%	99.21%	100.00%	94.49%	84.25%	99.21%	100.00%	100.00%	100.00%	97.64%
0	**3**	**orig.**	**92.13%**	**98.43%**	**96.85%**	**97.64%**	**91.34%**	**37.80%**	***99.21%***	***100.00%***	***100.00%***	***100.00%***	***98.43%***
0	5	10	97.64%	98.43%	99.21%	99.21%	92.13%	84.25%	96.06%	99.21%	99.21%	99.21%	92.91%
0	5	20	98.43%	100.00%	99.21%	99.21%	96.06%	83.46%	98.43%	100.00%	100.00%	100.00%	96.06%
0	5	40	98.43%	100.00%	100.00%	99.21%	96.85%	84.25%	98.43%	100.00%	100.00%	100.00%	96.85%
0	**5**	**orig.**	**96.85%**	**100.00%**	**99.21%**	**100.00%**	**94.49%**	**35.43%**	***99.21%***	***100.00%***	***100.00%***	***100.00%***	***96.85%***
1	0	10	96.85%	95.28%	96.85%	96.85%	93.70%	82.68%	94.49%	96.85%	97.64%	97.64%	95.28%
1	0	20	90.55%	97.64%	97.64%	97.64%	92.13%	83.46%	98.43%	100.00%	100.00%	100.00%	96.85%
1	0	40	92.13%	96.06%	96.85%	96.85%	88.98%	84.25%	96.85%	98.43%	99.21%	99.21%	96.06%
1	**0**	**orig.**	**90.55%**	**94.49%**	**95.28%**	**95.28%**	**90.55%**	**33.86%**	***96.06%***	***98.43%***	***99.21%***	***99.21%***	***96.85%***
1	1	10	92.91%	96.85%	96.85%	97.64%	90.55%	83.46%	96.85%	98.43%	97.64%	97.64%	94.49%
1	1	20	89.76%	94.49%	93.70%	94.49%	88.19%	83.46%	93.70%	96.85%	99.21%	99.21%	96.06%
1	1	40	81.89%	89.76%	88.98%	88.98%	83.46%	83.46%	95.28%	94.49%	95.28%	96.85%	96.06%
1	**1**	**orig.**	**62.99%**	**85.04%**	**90.55%**	**90.55%**	**80.31%**	**35.43%**	***95.28%***	***92.91%***	***96.85%***	***97.64%***	***88.98%***
1	2	10	84.25%	89.76%	88.19%	89.76%	74.02%	81.89%	93.70%	96.06%	96.06%	96.85%	89.76%
1	2	20	87.40%	88.19%	88.19%	89.76%	76.38%	83.46%	94.49%	98.43%	96.06%	96.06%	88.19%
1	2	40	82.68%	88.19%	88.98%	88.19%	77.95%	84.25%	93.70%	98.43%	97.64%	97.64%	88.19%
1	**2**	**orig.**	**82.68%**	**93.70%**	**92.13%**	**92.13%**	**82.68%**	**40.16%**	***96.85%***	***98.43%***	***98.43%***	***98.43%***	***91.34%***
1	3	10	97.64%	99.21%	99.21%	100.00%	98.43%	83.46%	99.21%	99.21%	99.21%	99.21%	99.21%
1	3	20	99.21%	99.21%	99.21%	99.21%	98.43%	83.46%	98.43%	99.21%	99.21%	99.21%	99.21%
1	3	40	99.21%	99.21%	99.21%	99.21%	99.21%	84.25%	99.21%	100.00%	99.21%	99.21%	99.21%
1	**3**	**orig.**	**98.43%**	**100.00%**	**99.21%**	**100.00%**	**97.64%**	**35.43%**	***99.21%***	***100.00%***	***100.00%***	***100.00%***	***99.21%***
1	5	10	98.43%	99.21%	99.21%	99.21%	99.21%	83.46%	99.21%	99.21%	99.21%	99.21%	99.21%
1	5	20	99.21%	99.21%	99.21%	99.21%	99.21%	83.46%	98.43%	99.21%	99.21%	99.21%	99.21%
1	5	40	99.21%	99.21%	99.21%	99.21%	99.21%	84.25%	99.21%	100.00%	99.21%	99.21%	98.43%
1	**5**	**orig.**	**98.43%**	**100.00%**	**100.00%**	**100.00%**	**99.21%**	**37.01%**	***99.21%***	***100.00%***	***100.00%***	***100.00%***	***100.00%***

Table 3: *Set4* – micro-averaging categorisation accuracy for 1-NN.

6 Discussion

The most interesting observations are:

(1) The choice of weighting scheme is among the most important factors. When the appropriate combination of LWF*GWF is used (1*3, 1*5, 0*3, 0*5), other factors such as *stop-words removal* and *LSA dimensionality reduction*, become almost irrelevant.

(2) Stemming and lemmatisation are almost equally good for the highly inflectional Bulgarian.

(3) "The statistics dominates the linguistics": for the best performing combination of LWF*GWF (1*3 and 1*5) *the definition of word becomes irrelevant*. Note however that this might be due to the task selected – text categorisation, and may not hold for IR in general. Thus, more tests with respect to a variety of IR related tasks are needed. However, if we look at the worse weighting schemes in Table 3, e.g. GWF∈{0,1,2} and *stop-words kept*, we can see that stemming delivers up to 26% improvement. This suggests that stemming could still be important but its contribution could just be obscured by the impact of weighting. This hypothesis though fails for *Set15*: as Table 5 shows, the impact of stemming is less

than 1% in case we use 1*3 and 1*5 values for LWF*GWF (see the original space and the LSA dimensionality of 100; 100 is among the most popular dimensionalities and proved to be among the best performing on *Set15*).

(4) The stop-words removal has a really dramatic impact but only when GWF∈{0,1,2}. When the best weighting schemes are used (1*3 or 1*5), its impact is up to 1.5% (see Tables 3, 4 and 5).

(5) With the exception of 1*3 and 1*5 the POS disambiguation performs consistently worse than the other experiments. This means that even though words with different POS have different meanings, these might be close enough (polysemy) so that it is better to keep them together. This is consistent with the observations of Krovetz that while resolving homography is beneficial, disambiguating polysemy damages the IR performance (Krovetz 93).

(6) Using phrases only is consistently far worse than stop-words removal.

(7) Combining phrases and lemmatisation gives slightly improved results over lemmatisation only, but these are still almost indistinguishable from the ones obtained using stemming.

(8) LWF = 1 leads to substantial benefits but only when GWF∈{0,1,2} and the stop-words are kept. When the stop-words are removed, the impact is insignificant. This is easy to explain: the stop-words are the most frequent ones and the logarithmic weighting (LWF=1) makes a big difference mostly for them.

LWF	GWF	LSA dim.	STOP-WORDS KEPT						STOP-WORDS REMOVED				
			raw words	stem	lemma	lemma& phrase	POS disam.	phrase only	raw words	stem	lemma	lemma & phrase	POS disam.
0	**0**	**40**	87.40%	91.34%	89.76%	89.76%	85.83%	83.46%	96.06%	**100.00%**	98.43%	98.43%	92.91%
0	**0**	**orig.**	81.89%	92.13%	91.34%	90.55%	81.10%	42.52%	96.06%	96.06%	98.43%	99.21%	92.91%
0	**1**	**40**	85.83%	90.55%	88.98%	88.98%	84.25%	83.46%	94.49%	97.64%	98.43%	98.43%	92.13%
0	**1**	**orig.**	76.38%	90.55%	92.13%	90.55%	76.38%	34.65%	96.06%	96.06%	98.43%	99.21%	92.91%
0	**2**	**40**	61.42%	71.65%	70.08%	70.08%	59.84%	83.46%	92.91%	98.43%	97.64%	97.64%	86.61%
0	**2**	**orig.**	59.84%	70.08%	67.72%	67.72%	59.84%	40.94%	95.28%	97.64%	98.43%	98.43%	87.40%
0	**3**	**40**	93.70%	98.43%	98.43%	99.21%	95.28%	84.25%	99.21%	**100.00%**	**100.00%**	**100.00%**	98.43%
0	**3**	**orig.**	95.28%	98.43%	97.64%	98.43%	92.91%	37.80%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	99.21%
0	**5**	**40**	99.21%	99.21%	**100.00%**	99.21%	97.64%	84.25%	98.43%	**100.00%**	**100.00%**	**100.00%**	97.64%
0	5	**orig**.	96.85%	**100.00%**	**100.00%**	**100.00%**	93.70%	38.58%	99.21%	**100.00%**	**100.00%**	**100.00%**	97.64%
1	**0**	**40**	93.70%	97.64%	96.85%	96.85%	90.55%	83.46%	97.64%	98.43%	99.21%	99.21%	96.85%
1	**0**	**orig.**	92.91%	96.06%	95.28%	95.28%	92.13%	33.86%	95.28%	97.64%	98.43%	98.43%	96.85%
1	**1**	**40**	92.91%	95.28%	95.28%	94.49%	94.49%	83.46%	95.28%	96.06%	96.85%	96.85%	93.70%
1	**1**	**orig.**	83.46%	89.76%	90.55%	90.55%	87.40%	39.37%	95.28%	93.70%	96.85%	96.85%	92.91%
1	**2**	**40**	84.25%	93.70%	88.19%	88.98%	77.17%	84.25%	96.06%	98.43%	98.43%	98.43%	90.55%
1	**2**	**orig.**	83.46%	93.70%	92.13%	91.34%	82.68%	40.94%	96.85%	98.43%	98.43%	98.43%	93.70%
1	**3**	**40**	99.21%	**100.00%**	99.21%	99.21%	99.21%	84.25%	99.21%	**100.00%**	99.21%	99.21%	99.21%
1	**3**	**orig.**	98.43%	99.21%	**100.00%**	**100.00%**	97.64%	36.22%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
1	**5**	**40**	99.21%	**100.00%**	99.21%	99.21%	99.21%	84.25%	99.21%	**100.00%**	99.21%	99.21%	99.21%
1	**5**	**orig.**	98.43%	**100.00%**	**100.00%**	**100.00%**	99.21%	40.16%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**

Table 4: *Set4* – micro-averaging categorisation accuracy for 10-NN (dividing by the rank).

(9) It is interesting to compare the original vector space and the reduced LSA one. These are often comparable (almost equal for e.g. 1*3 and 1*5) but sometimes there are bigger differences, e.g. for 0*1, stop-words kept, raw words: 61.42% vs. 85.04% (LSA dim. 20). The impact is even more dramatic for *phrase only* experiments: for any combination of LWF*GWF the performance of LSA (for all dimensionality reductions listed) is more than twice the performance without dimensionality reduction: e.g. 33.86% vs. 84.25%. This is explained by the fact that the average number of multi-word phrases per document is very limited, and thus the documents share very few of them, which is not enough to judge similarity in a reliable way. When using LSA though, the projected space contains a summary of the co-occurrences as well as some transitive implications, observed globally across documents. So, even though the LSA vector dimensions are much fewer, they contain much less zero components and thus discriminate better. A similar effect (with a similar explanation) is observed for the raw text without stop-words removal, where LSA proves consistently better (especially for 0*1 and 1*1) for all dimensionality reductions.

(10) Changing the number of neighbours *k* considered in the *k*NN (e.g. *k* = 10, see Table 4) leads to improvement but only unless an appropriate GWF is used (e.g. 3 or 5). In the latter case the results are already fairly good for 1-NN, so there is no much space for improvement left. While *k* = 10 was among the best performing values for *Set4*, for the bigger *Set15* we obtain a consistent improvement as *k* grows from 1 up to 40.

(11) The categories count is another important parameter: going from 4 to 15 categories lowers the best accuracy for *raw words* and *stemming* from 100.00% (Table 3) to 75.50% and 76.35% (Table 5).

(12) For *Set15*, stemming is beneficial, but the stop-words removal has an even bigger impact. The latter is not that obvious from tables 3 and 4 since the values are too close to 100.00%. However, the most important parameter remains the weighting scheme: when GWF∈{3,5} the impact of stemming, stop-words removal and even of LWF is limited.

LWF	GWF	LSA dim.	stop-words kept		*no stop words*	
			raw	stem	*raw*	*stem*
0	**0**	**20**	41.45%	50.14%	*60.83%*	*62.68%*
0	**0**	**100**	50.00%	59.54%	*66.67%*	*70.66%*
0	**0**	**orig.**	45.73%	58.55%	*69.94%*	*72.79%*
0	**1**	**20**	40.17%	51.99%	*61.54%*	*62.25%*
0	**1**	**100**	46.87%	57.12%	*65.95%*	*71.08%*
0	**1**	**orig.**	45.58%	58.12%	*69.94%*	*72.93%*
0	**2**	**20**	26.07%	27.64%	*49.86%*	*58.26%*
0	**2**	**100**	26.78%	28.63%	*60.97%*	*67.09%*
0	**2**	**orig.**	27.49%	29.91%	*66.10%*	*70.94%*
0	**3**	**20**	62.39%	64.67%	*68.09%*	*69.94%*
0	**3**	**100**	**69.09%**	**73.08%**	***73.22%***	***75.21%***
0	**3**	**orig.**	**71.37%**	**74.36%**	***75.21%***	***75.78%***
0	**5**	**20**	65.53%	69.23%	*66.67%*	*70.23%*
0	**5**	**100**	**72.36%**	**75.93%**	***74.07%***	***76.35%***
0	**5**	**orig.**	**73.79%**	**75.50%**	***75.50%***	***75.93%***
1	**0**	**20**	60.68%	65.81%	*63.96%*	*66.95%*
1	**0**	**100**	64.81%	69.37%	*71.37%*	*71.79%*
1	**0**	**orig.**	69.23%	71.79%	*71.37%*	*72.93%*
1	**1**	**20**	59.12%	64.10%	*64.81%*	*65.95%*
1	**1**	**100**	60.40%	66.24%	*54.27%*	*67.38%*
1	**1**	**orig.**	66.67%	71.79%	*71.23%*	*72.79%*
1	**2**	**20**	38.46%	50.14%	*62.54%*	*65.67%*
1	**2**	**100**	47.15%	58.12%	*66.81%*	*71.51%*
1	**2**	**orig.**	50.28%	60.83%	*70.94%*	*73.22%*
1	**3**	**20**	68.52%	69.23%	*68.66%*	*69.52%*
1	**3**	**100**	**72.93%**	**74.07%**	***74.36%***	***73.79%***
1	**3**	**orig.**	**72.51%**	**72.93%**	***72.22%***	***73.22%***
1	**5**	**20**	68.66%	70.66%	*68.80%*	*71.08%*
1	**5**	**100**	**73.36%**	**75.21%**	***74.36%***	***74.93%***
1	**5**	**orig.**	**72.79%**	**73.08%**	***72.51%***	***73.08%***

Table 5: *Set15* – Micro-average accuracy, 10-NN (dividing by the rank).

7 Conclusion and Future Work

Some earlier research on the impact of the linguistically motivated text indexing on IR performance shows that a better word pre-processing is not necessarily needed for effective retrieval (Spark-Jones 99; Strzalkowski 99). It is not clear whether these doubts are due to the weakness of the linguistic analysis, which still does not produce a semantically motivated concept-based representation of the text, or the linguistic analysis as such is not relevant to IR and to text classification in particular (Peng et al. 03). But the current work allows us to gain important insights regarding the need and the role of the linguistic pre-processing.

The experiments above show that while the definition of word has a substantial influence on the LSA performance, the traditional statistical IR factors are even more important: it is enough to look at the major impact of the stop-words removal for some weighting schemes. It is worth mentioning though, that the feature engineering is both language- and task-dependent: e.g. the stop-words may be among the best features for other text categorisation tasks, e.g. language identification or authorship attribution. In the latter case, other features, such as linguistic style markers, are usually even more important.

The experiments above show that the word definition becomes almost irrelevant once we are stuck to topic identification, limit ourselves to the bag-of-words model, have only few well-separated categories (*Set4*) and use the best weighting schemes (e.g. 1*3 and 1*5). In the case of more and less distinguished classes though (*Set15*), both stop-words removal and stemming become important (although still far less than the weighting scheme). More experiments are needed in order to study the impact of lemmatisation, phrases and POS disambiguation in the latter case.

Our future plans include evaluation of the impact of lemmatisation, phrases and POS disambiguation on *Set15* and bigger sets, which would make our experiments and results more complete. It would be also interesting to try using word senses (e.g. with respect to some semantic network) instead of the words themselves, but the bad performance of the POS disambiguation above leaves us sceptical. Another reason for scepticism are the negative results already obtained for English: an extensive study shows that using WordNet senses does not lead to any significant improvement on the Annotated Brown Corpus (Kehagias et al. 01). The problem with the latter work (as well as with the POS disambiguation above) though, is that, in addition to homonymy, the polysemy has also been addressed. This is just the contrary to what stemming does.

Two other important parameters are missing from our study: choice of similarity measure and feature selection. Both have been found important for text categorisation and are interesting to study. It is worth trying other classifiers as well, e.g. Rocchio, Naive Bayes, decision trees etc.

Finally, we would like to consider other IR tasks in an attempt to check whether the appropriate word definition is more important for the classic IR than for text topic identification. This would also allow us to perform a comparison with a recent work (Peng et al. 03), which shows that using simple language- and task-independent character-level (as opposed to word-level) text compression techniques achieves a comparable or even better text categorisation results than the traditional techniques do.

8 Acknowledgements

We would like to thank the anonymous reviewers for the useful comments and suggestions.

References

(Bartell et al. 92) B. Bartell, G. Cottrell, R. Belew. *Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling*. Proc. 15th ACM SIGIR Conf. on R&D in IR. pp. 161-167, 1992.

(Berry et al. 95) M. Berry, S. Dumais, G. O'Brien. *Using linear algebra for intelligent information retrieval.* SIAM: Review, vol. 37(4), pp. 573-595, 1995.

(Berry et al. 93) M. Berry, T. Do, G. O'Brien, V. Krishna, S. Varadhan, *SVDPACKC (Version 1.0) User's Guide*. April 1993.

(Deerwester et al. 90) S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman. *Indexing by Latent Semantic Analysis.* Journal of the Am. Soc. for Information Sciences, vol. 41, pp. 391-447, 1990.

(Domingos & Pazzani 97) P. Domingos, M. Pazzani. *On the optimality of the simple Bayesian classifier under zero-one loss.* Machine Learning, vol. 29, pp.103-130, 1997.

(Dumais 91) S. Dumais. *Improving the retrieval of information from external sources.* Behavior Research Methods, Instruments & Computers. vol. 23(2), pp. 229-236, 1991.

(Foltz & Dumais 92) P. Foltz, S. Dumais. *Personalized information delivery: An analysis of information filtering methods*. In CACM, vol. 35(12), pp.51-60, 1992.

(Furnas et al. 86) G. Furnas, T. Landauer, L. Gomez, T. Dumais. *Statistical semantics: Analysis of the Potential Performance of Keyword Information Systems.* Bell Syst.Tech. Jornal, vol. 62(6), pp. 1753-1806, 1986.

(Harman 91) D. Harman. *How effective is suffixing?* Journal of the American Society of Information Science. vol. 42, No 1. pp. 7-15, 1991.

(Hull 96) D. Hull. *Stemming Algorithms: A Case study for detailed evaluation.* In Journal Am. Soc. of Information Science, vol. 47(1), pp. 70-84, 1996.

(Joachims 97) T. Joachims. *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.* In Machine Learning: Proc. of 14th Int. Conference (ICML '97), pp. 143-151, 1997.

(Kehagias et al. 01) A. Kehagias, V. Petridis, A. Kaburlasos, P. Fragkou. *A Comparison of Word- and Sense-based Text Categorization Using Several Classification Algorithms.* J. Int. Inf. Systems (accepted), 2001.

(Kovalenko 02) A. Kovalenko. *Stemka: Morphological analyzer for small search systems.* In System Administrator Magazine. Moscow, October 2002.

(Kraaij & Pohlmann 94) W. Kraaij, R. Pohlmann. *Porter's stemming algorithm for Dutch.* Noordman, de Vroomen, (Eds). Informatiewetenschap 1994: STINFON Conferentie, Tilburg, pp. 167-180, 1994.

(Krovetz 93) R. Krovetz. *Viewing Morphology as an Inference Process.* Proc. 16th ACM SIGIR Conf. on R&D in IR. pp. 191-202. ACM. New York. 1993.

(Landauer & Dumais 97) T. Landauer, S. Dumais. *A Solution to Plato's Problem: The LSA Theory of Acquisition, Induction and Representation of Knowledge.* Psychological Review. vol. 104, pp. 211-240, 1997.

(Landauer et al. 98) T. Landauer, P. Foltz, D. Laham. *Introduction to LSA.* Discourse Processes, vol. 25, pp. 259-284, 1998.

(Lewis & Ringuette 94) D. Lewis, M. Ringuette. *A comparison of two learning algorithms for text categorization.* Proc. 3rd Ann. Symposium on Document Analysis and IR, pp. 81-93, 1994.

(Mitchell 96) T. Mitchell. *Machine Learning.* McGraw Hill, 1996.

(Nakov 03) P. Nakov. *Building an Inflectional Stemmer for Bulgarian.* Proc. CompSysTech, Sofia, 2003 (to appear).

(Nakov et al. 01) P. Nakov, A. Popova, P. Mateev. *Weight functions impact on LSA performance.* Proc. RANLP'2001, pp. 187-193, 2001.

(Nakov 00) P. Nakov. *Getting Better Results with Latent Semantic Indexing.* Proc. of the Students Presentations at ESSLLI-2000, pp. 156-166, Birmingham, 2000.

(Peng et al. 03) F. Peng, D. Schuurmans, S. Wang. *Language and task independent text categorization with simple language models.* HLT Conference (HLT-NAACL-03), pp. 110-117, 2003.

(Popovic & Willett 92) M. Popovic, Willett P. *The Effectiveness of Stemming for NL access to Slovene Textual Data.* J. Am. Soc. of Inform. Science. vol. 43(5), pp.384-390, 1992.

(Porter 80) M. Porter. *An algorithm for suffix stripping.* Program vol. 14(3), pp. 130–137, 1980.

(Rocchio 71) J. Rocchio. *Relevance feedback in information retrieval.* In Salton G. (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall, 1971.

(Savoy 93) J. Savoy. *Stemming of French words based on grammatical categories.* In J. of the Am. Society for Information Science. vol. 44(1), pp. 1-9, 1993.

(Scott & Matwin 99) S. Scott, S. Matwin. *Feature Engineering for Text Classification.* In Bratko, Dzeroski (Eds.) Machine Learning: Proc. 16th Int. Conf., pp. 379-388, 1999.

(Spark-Jones 72) K. Spark-Jones. *A statistical interpretation of term specificity and its applications in retrieval*. J. Documentation, vol. 28, pp. 11-21, 1972.

(Spark-Jones 99) K. Spark-Jones. *What is the role of NLP in text retrieval?* In T. Strzalkowski (ed.) Natural Language Information Retrieval. Kluwer Academic Publishers, vol. 7, pp. 1-24, 1999.

(Strzalkowski 99) T. Strzalkowski (ed.) *Natural Language Information Retrieval.* Kluwer Academic Publishers, Series "Text, Speech and Language Technology" Vol. 7, 1999.

(Wiener et al. 93) E. Wiener, J. Pedersen, A. Weigend. *A neural network approach to topic spotting.* Proc. of the 4th An. Symp. on Document Analysis and IR, pp. 22-34, 1993.

(Yang 99) Y. Yang. *An evaluation of statistical approaches to text categorization.* Journal of IR, vol. 1, nos. 1/2, pp. 67-88, 1999.