AUTOMATIC ESTIMATION OF THE LSA DIMENSION

Jorge Fernandes, Andreia Artífice and Manuel J. Fonseca
Department of Computer Science and Engineering, INESC-ID/ IST/ Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal

Keywords: LSA, LSA dimension, Unsupervised text classification, Bootstrapping.

Abstract: Nowadays the size of collections of information achieved considerable sizes, making the finding and exploration of a particular subject hard to achieve. One way to solve this problem is through text classification, where a theme or category is assigned to a text based on the analysis of its content. However, existing approaches to text classification require some effort and a high level of knowledge on this subject by the users, making them inaccessible to the common user. Another problem of current approaches is that they are optimized for a specific problem and can not easily be adapted to another context. In particular, unsupervised methods based on the LSA algorithm require users to define the dimension to use in the algorithm. In this paper we describe an approach to make the use of text classification more accessible to common users, by providing a formula to estimate the dimension of the LSA based on the number of texts used during the bootstrapping process. Experimental results show that our formula for estimation of the LSA dimension allows us to create unsupervised solutions able to achieve results similar to supervised approaches.

1 INTRODUCTION

Text classification consists in assigning a category (from a set of predefined categories) to a text, based on its content. Although this problem dates from the 60s, it still is relevant since the amount of (uncatalogued) information increases everyday. Information such as, RSS feeds, news, scientific papers, e-books, etc., need to be organized and cataloged to make life easier for users.

Nowadays when we want to classify a set of texts we can either resort to supervised, unsupervised or even hybrid approaches. While supervised solutions require the manual classification of a set of texts, unsupervised approaches avoid that by using a bootstrapping method to perform a first classification of unclassified texts. The classified texts (in both approaches) are then used to train a classifier, which will be later used to classify other texts.

Various unsupervised approaches use the Latent Semantic Analysis (LSA) (Landauer et al., 1998) algorithm to perform feature extraction from the texts. Since one of the main characteristics of the LSA algorithm is its dimension, the selection of this value is of crucial importance because it affects the final results of the classification. However, from the literature we did not find any founded choice for its value, being its selection made by skilled people, according to the actual context of the problem and after several iterations to optimize the final results.

In this paper we propose an empirical formula to automatically estimate the best value for the LSA dimension, according to the current context of the problem, namely the number of documents used during the bootstrapping step. We applied this formula to estimate the LSA dimension of an unsupervised system and compare the classification results with a supervised solution. Experimental evaluation show that the results are similar, with the advantage of not requiring any input from the users beside the set of texts to be used by the bootstrapping algorithm.

The remainder of this paper is organized as follows. Section 2 provides a summary of the supervised and unsupervised approaches for text classification. In Section 3 we present our solution for the automatic estimation of the LSA dimension and in Section 4 we present the results of the experimental evaluation. Finally, in Section 5 we conclude the paper.

2 RELATED WORK

Most of the existing text classification techniques can be grouped into two groups: supervised learning and unsupervised learning.

Fernandes J., Artífice A. and J. Fonseca M..
AUTOMATIC ESTIMATION OF THE LSA DIMENSION.
DOI: 10.5220/0003666103010305
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2011), pages 301-305
ISBN: 978-989-8425-79-9

Supervised Learning requires a manual classification of a group of texts into a predefined set of categories. This result is then used to train and build an automatic classifier able to categorize any text into the predefined set of categories.

According to (Huang, 2001), there are two key factors for a successful supervised learning. One is the feature extraction, which should accurately represent the contents of text in a compact and efficient manner, and the other is the classifier design, which should take the maximum advantage of the properties inherent to the texts, to achieve the best possible results. Huang studied several algorithms for both factors, and concluded that the LSA algorithm is the most appropriated for the feature extraction, and the SVM for the classifier.

(Debole and Sebastiani, 2003) and (Ishii et al., 2006) both agree with Huang in using the LSA for feature extraction, but they introduced some changes to the feature extraction process. While the first authors included a number of "supervised variants" of TFIDF weighting, the second authors complemented the LSA by introducing the concept of data grouping.

Although supervised learning can obtain good results, they require a large number of texts (literature values vary between 500 and 1400) and a manual classification to train the final classifier.

Unsupervised Learning tries to overcome the disadvantages of the supervised approaches by replacing the manual classification of a high number of texts with an automatic classification (often called bootstrapping). By doing so we are able to greatly reduce the costs and the need for human intervention.

Unfortunately the automatic classification of texts used by the unsupervised learning can cause various misclassification, introducing noise in the training of the classifier and affecting its final performance, which traditionally is worst than in the supervised learning.

Since most unsupervised approaches requires a list of representative keywords for each category, some authors tried to improve the bootstrapping quality by developing algorithms to help in the selection of the best keywords for each category. (Liu et al., 2004) used a clustering algorithm to identify the most important words for each cluster of texts. Then the user could inspect the ranked list and select a small set of representative keywords for each category.

(Barak et al., 2009) went a step forward by completely automating the process. Their approach attempts to automatically extract possible keywords using only the category name as a starting point. The authors introduced also a novel scheme that models both lexical references, based on certain relations present in WordNet and Wikipedia, and contextual references, using the model of the LSA. From the resulting model they extract the necessary keywords.

(Gliozzo et al., 2005) tried to minimize the number of misclassifications of the bootstrapping by first preprocessing the text to remove all the words that are not nouns, verbs, adjectives and adverbs. The resulting set of words is then represented using LSA. An algorithm based on unsupervised estimation of Gaussian Mixtures is then applied to differentiate between relevant and non-relevant information using statistics from unclassified texts. According to the authors a SVM classifier trained with the results from this bootstrapping algorithm achieved results comparable to a supervised solution.

In summary, although supervised learning approaches present the best results, they require someone (an expert person) to manually classify a large number of texts, which is an arduous and monotonous task, with an enormous cost associated. On the other hand, the unsupervised learning avoids the manual classification by including a bootstrapping technique, but requires specific knowledge about the algorithms in use (e.g. LSA) and of the domain problem. Indeed, when we use an approach that reduce the dimension of the features extracted from the text, like for instance the LSA algorithm the selection of the dimension is very important, since its value affects the final results. From the analysis of the several existing proposals based on the LSA algorithm, we did not find a clear explanation on how to choose the best dimension for the LSA. In most cases its value is chosen after several iterations and taking into account the specific context of the current problem.

To overcome this, to minimize the human intervention, and to offer good results, we propose in this paper a solution to automatically estimate the "optimum" dimension for the LSA algorithm, taking into account only the number of texts.

3 LSA DIMENSION ESTIMATION

The solution that we developed for the bootstrapping step starts by reducing the size of the vocabulary by removing useless words that only introduce noise in the categorization. We remove words contained into a stopwords list and the least frequent words (words that appeared less than three times)(Joachims, 1997). By removing the least frequent words we are able to eliminate typos and reduce the noise of the vocabulary. Additionally, by reducing the size of the vocabulary we will reduce the complexity of the problem and the computational cost of all the following algorithms.

To represent the content of the texts we used the LSA. We built a word-document matrix by grouping the individual document representations and applied a TFIDF (Salton and Buckley, 1988) to the resulting frequency matrix, followed by SVD (Berry, 1992) to obtain the new matrices of reduced dimensionality. The reduced dimension of the resulting space need to be carefully selected (as we will see below) to fit well the problem to be solved. Then, the resulting latent semantic space is used to classify the documents and this classification is used to train the classifier.

To get satisfactory results the dimension of the LSA algorithm must be selected appropriately. Typically this value is selected empirically, based on values used on other similar problems, or through various tests to identify the interval where the optimum value of the dimension is. This represents a problem, because an ordinary person does not have the knowledge to make this selection, and also because we can not use a fixed value. Here, we propose a solution that will allow ordinary people to use the LSA algorithm in various problem contexts, by defining a formula for the estimation of the most appropriated LSA dimension based on the context of the problem. To that end, we analyzed the behavior of the LSA and performed several experimental tests.

From the literature, we found that the LSA's performance increases as the number of dimensions also increases, until reaching a maximum. After that value, any further increase in the number of dimensions will only decrease the performance. Moreover, if we look carefully into the LSA algorithm we can see that the number of dimensions is affected by the size of the vocabulary and by the number of texts used to extract the features. Since the size of the vocabulary is somehow directly related to the number of texts, we can assume that the number of dimensions depends exclusively from the number of texts.

Based on this we performed a set of experimental studies to identify the range of dimension values where the performance has a maximum, and tried to figure out a formula to estimate a value for the dimension within that interval.

To perform the experimental tests we considered various context problems, where we had ten categories (Science and Technology, Cinema and TV, Sport, Economy and Management, Informatics and Internet, Games, Music, Politics, Health, and Motor Vehicles), with distinct characteristics among them, and three number of texts per category (32, 64 and 128). Texts were collected from news sites, and their sizes vary from a few paragraphs to one to two pages.

For each problem context we performed five individual tests using five different sets of texts (in the same conditions) and measured the F1-measure. The final result for a specific context is the average of the F1-measure from the five individual tests.

The values of the dimension used to compute the F1-measure were selected to figure out if the optimum value for the dimension varies linearly or non-linearly. To that end we considered the following values for the dimension: $\sqrt{n_T}$, $2\sqrt{n_T}$, $3\sqrt{n_T}$, $\frac{1}{5}n_T$ and $\frac{1}{3}n_T$, where n_T represents the number of texts.

We first studied the performance for 32 texts per category (320 texts in total), and we achieved the values depicted in Figure 1. As we can see we have a maximum for 106 dimensions. It is important to highlight that this does not mean that the optimum value for the dimension is 106, but that the F1-measure increases and then decreases, with the optimum value inside the interval $]64,160[$. In this particular case, we also computed the F1-measure for a dimension of $\frac{1}{2}n_T$ because the F1-measure was still increasing at $\frac{1}{3}n_T$.

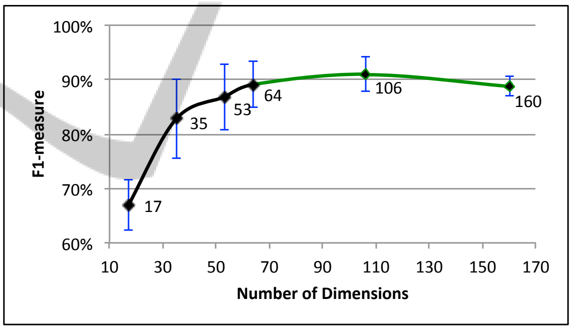

Figure 1: LSA performance for a set of 320 texts.

For the next test we used 64 texts per category (640 texts in total), and we achieved the values depicted in Figure 2. As we can see we achieved a maximum for 128, and the interval for the optimum value is $]75,213[$.

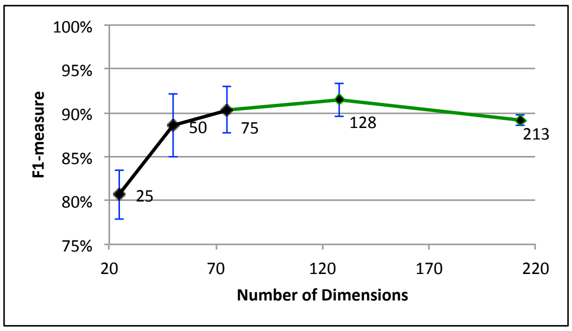

Figure 2: LSA performance for a set of 640 texts.

Finally, we used 128 texts per category (1280 texts in total), producing the results depicted in Figure 3. In this case we have a maximum for 107 dimensions and an interval for the optimum value between $]71,256[$.

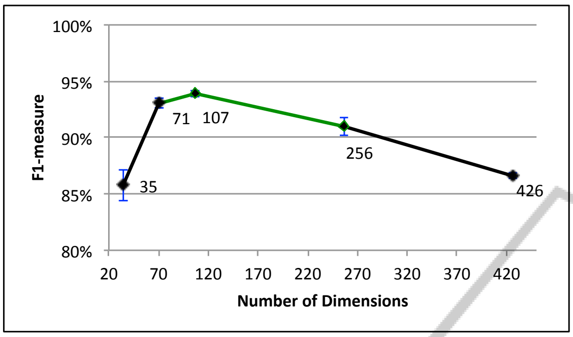

Figure 3: LSA performance for a set of 1280 texts.

By analyzing the obtained data, the first conclusion that we can take is that the number of dimensions does not grow linearly with the number of texts. Indeed, while the number of texts increase the number of dimensions corresponds to a smaller percentage of the total number of texts. In the first case (320 texts) the optimum value is between 20% and 50% of the number of texts, in the second case (640 texts) is between 12% and 33%, and in the last case (1280 texts) is between 6% and 20%.

After looking at various mathematical functions we identified the square root as the one with the most similar behavior. Based on that we studied some possibilities and achieved the following formula for the estimation of the LSA dimension:

$$k = n_T^{\left(\frac{1}{1+\frac{\log(n_T)}{10}}\right)} \qquad (1)$$

where n_T is the number of texts.

If we now apply this formula to the previous three cases we obtain the following values:

Table 1: Intervals for the optimum values of dimension, and the values automatically estimated using equation 1.

# Texts	Expected interval	Estimated Value
320	$]64,160[$	100
640	$]75,213[$	155
1280	$]71,256[$	235

As we can see all the estimated values belong to the interval where the optimum value is. Moreover, the literature mentioned that for problems where we have more than 20000 texts the typical value recommended for the dimension is between 200 and 2000. Though, we can conclude that our formula estimates values inside that interval.

4 EXPERIMENTAL EVALUATION

To evaluate our formula for the estimation of the appropriate dimension for the LSA, we compared the results achieved by a classifier trained with the classification produced by our bootstrapping algorithm and a classifier trained using texts classified manually. Our goal was to check if our solution, where the overall bootstrapping step was automatized with the inclusion of the estimation of the LSA dimension, can compete with a supervised solution, where exist a lot of human intervention.

To perform the tests, we used the same set of texts in both solutions, supervised and unsupervised. Then, we compared the two classifiers using two distinct situations. One where we used 500 texts to train the classifiers and 1000 for evaluation, and another where we used 1000 texts to train and 500 for evaluation. In both situations the two sets were disjoints.

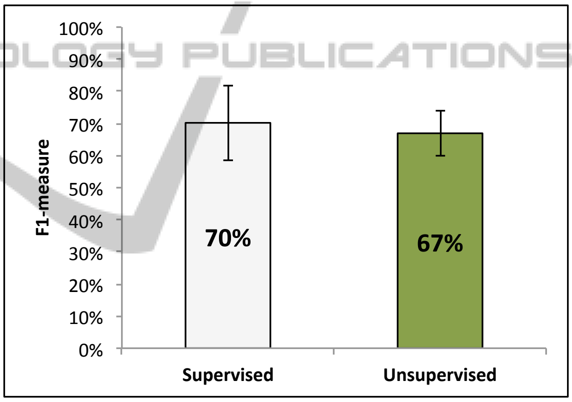

Figure 4: Performance results for supervised and unsupervised solutions, using 500 texts to train and 1000 texts for recognition.

Figure 4 shows the results achieved by both solutions for the first situation, while Figure 5 presents the results for the second case. As we can see in both cases our approach presents results similar to the supervised solution. In the first case we have a F1-measure of 67% against 70% an in the second we have 84% against 85%. Although in both cases the F1-measure is smaller for the unsupervised solution, the standard deviations intersect, meaning that our approach can achieve results comparable to supervised solutions without their costs.

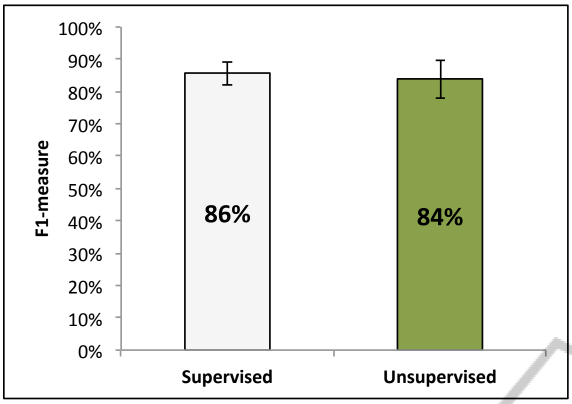

Figure 5: Performance results for supervised and unsupervised solutions, using 1000 texts to train and 500 texts for recognition.

5 CONCLUSIONS

As we have seen unsupervised solutions have a bootstrapping step where, in the majority of the cases, a LSA algorithm is used. However, to properly take advantage of the LSA a good selection of its dimension is crucial. In this paper we presented a solution, based on a set of empirical studies, to automatically estimate the most appropriated dimension for the LSA. By providing this estimation mechanism, we will allow people without specific knowledge about the LSA algorithm to use it parameterized with the correct values to assure a good performance.

Indeed, from the experimental evaluation we can conclude that our formula allows unsupervised solutions based on the LSA to achieve results similar to supervised methods.

ACKNOWLEDGEMENTS

This work was supported by FCT through the PIDDAC Program funds (INESC-ID multiannual funding) and the Crush project, PTDC/EIA-EIA/108077/2008.

REFERENCES

Barak, L., Dagan, I., and Shnarch, E. (2009). Text categorization from category name via lexical reference. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 33–36, Morristown, NJ, USA. Association for Computational Linguistics.

Berry, M. (1992). Large scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6:13–49.

Debole, F. and Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 784–788. ACM Press.

Gliozzo, A., Strapparava, C., and Dagan, I. (2005). Investigating unsupervised learning for text categorization bootstrapping. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 129–136, Morristown, NJ, USA. Association for Computational Linguistics.

Huang, Y. (2001). Support vector machines for text categorization based on latent semantic indexing. Technical report, Electrical and Computer Engineering Department, The Johns Hopkins University.

Ishii, N., Murai, T., Yamada, T., and Bao, Y. (2006). Text classification by combining grouping, lsa and knn. In *ICIS-COMSAR '06: Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse*, pages 148–154, Washington, DC, USA. IEEE Computer Society.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. In *Discourse Processes*, pages 259–284.

Liu, B., Li, X., Lee, W., and Yu, P. (2004). Text classification by labeling words. In *AAAI'04: Proceedings of the 19th national conference on Artifical intelligence*, pages 425–430. AAAI Press.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.