The Application of Singular Value Decomposition to Dutch Noun-Adjective Matrices

Tim Van de Cruys
CLCG, Rijksuniversiteit Groningen
t.van.de.cruys@rug.nl

Résumé L'apprentissage automatique de la sémantique est un sujet assez populaire dans le domaine du traitement automatique du langage. Beaucoup de recherches ont été éffectuées en comparant des contextes syntaxiques similaires. On peut, par example, trouver des substantifs d'un champ sémantique similaire en examinant les adjectifs avec lesquels ils sont souvent relatés. Si on opte pour cette méthode, il y a néanmoins deux problèmes qui se posent, à savoir la complexité computationnelle et l'insuffisance des donneés. Cet article décrit l'application d'une technique mathématique, la décomposition en valeurs singulières. Cette technique a été appliquée au domaine de Recherche d'Information avec des résultats favorables. On pose la question s'il est possible de trouver, grâce à la technique, des dimensions sémantiques latentes à l'espace d'adjectifs reduit avec lesquelles on peut faire un groupement qui est aussi bon ou meilleur que le groupement original.

Abstract Automatically acquiring semantics from text is a subject that has gathered a lot of attention in natural language processing. A lot of research has been done by looking at syntactically similar contexts. For example, semantically related nouns can be clustered by looking at the collocating adjectives. There are, however, two major problems with this approach : computational complexity and data sparseness. This paper describes the application of a mathematical technique called singular value decomposition, which has been succesfully applied in Information Retrieval to counter these problems. It is investigated whether this technique is also able to cluster nouns according to latent semantic dimensions in a reduced adjective space.

Mots-clefs : analyse sémantique, clustering sémantique, LSA

Keywords: semantic analysis, semantic clustering, LSA

1 Introduction

Automatically acquiring semantics from text is one of the most difficult tasks in natural language processing. Most work that has been done in this field relies on the notion of *semantic similarity*. Automatically acquiring a relative measure of how similar a word is to known words is much easier than determining what the actual meaning is. Practically, semantic similarity is mostly calculated by using *vector space measures*. Such kind of approaches (Pantel & Lin, 2002), (van der Plas & Bouma, 2005), (van de Cruys, 2005) appear quite fruitful in the search for semantically related words.

There are, however, two major problems with this approach. The first problem is that computations will quickly get quite expensive when a large number of dimensions is used. A second problem is that this approach suffers from data sparseness : low frequent words do not appear frequently enough in corpora to be able to decently calculate the semantic similarity. In this paper, a mathematical technique called singular value decomposition (SVD) will be explored. This technique is known to counter both these problems in Information Retrieval, where it is applied to term-documents matrices (Landauer *et al.*, 1998).

2 Methodology

2.1 Calculating Semantic Similarity

Most work on semantic similarity relies on the Distributional Hypothesis (Harris, 1985). This hypothesis states that words that occur in similar contexts tend to be similar. Take for example the invented word *banban*, used in a number of contexts :

- un banban frais
- un banban salé
- un banban délicieux
- un banban sucré
- un banban dur

A speaker of French who is not familiar with the word *banban* can easily infer from the context that it is some kind of food. In the same way, a computer might be able to extract similar words from similar contexts, and group them into clusters.

The actual semantic similarity of words is determined by means of vector space measures. The two words, for which the semantic similarity is to be calculated, are represented as vectors in a multi-dimensional space. In this research, the vector space consists of the adjectives (modifiers) of the nouns. Figure 1 gives an example of four nouns represented as vectors in *modifier space*.

	rouge	délicieux	rapide	loué
pomme	2	1	0	0
vin	2	2	0	0
voiture	1	0	1	2
camion	1	0	1	1

FIG. 1 – A noun-by-adjective matrix

The matrix shows that the modifier *rouge* collocates with all four nouns, while *délicieux* only collocates with *pomme* and *vin*. On the other hand, *rapide* and *loué* only collocate with *voiture* and *camion*.

A computer needs to be able to compute this similarity. Several similarity measures are available to calculate the similarity among various patterns. A few possibilities are *Dice coefficient*, *Jaccard coefficient*, and *Overlap coefficient*. For an extensive overview of various similarity metrics, see (Weeds, 2003). In these experiments, the *cosine measure* has been used. The cosine measures the angle between two vectors.

2.2 Reducing the Dimensions

To reduce the number of dimensions, a singular value decomposition is applied to the noun-adjective matrix. Singular value decomposition is a mathematical technique that is closely akin to statistical methods such as factor analysis, correspondence analysis and principal components analysis. The rectangular noun-adjective matrix is decomposed into three other matrices. The first matrix contains a number of rows equal to the original matrix, but has m columns, corresponding to new, specially derived variables such that there is no correlation between any two columns : they are linearly independent. These derived variables are called the principal components. The second matrix has columns corresponding to the original columns, but m rows composed of derived singular vectors. The third matrix is a diagonal matrix : it is a square m by m matrix with non-zero entries along the diagonal. This matrix contains derived constants called *singular values,* which denote the variance explained by each derived component. If one or more singular values are omitted, then the reconstructed matrix will be the best possible approximation of the original matrix.

2.3 Clustering the Data

There are various clustering methods. An extensive overview of clustering is given in (Jain *et al.*, 1999). In general, a distinction can be made between :

- partitional clustering algorithms : algorithms that produce ‘stand-alone’ clusters which are not embedded in a structure ;
- agglomerative (hierarchical) clustering algorithms : algorithms that assign a complete branching structure to the various clusters, up to the root node.

Both algorithms have been explored. As a partitional algorithm, K-means has been used, and for hierarchical clustering group-average agglomerative clustering has been used. In the evaluation framework, K-means has been used, as this algorithm tends to score slightly better on noun clustering. K-means is non-deterministic, i.e. its outcome depends on the starting conditions. Because this is a problem for a proper evaluation, the algorithm has been iterated several times, and the best clustering has been chosen according to an optimization function.

2.4 Experimental Design

The noun-adjective collocations have been extracted from the **Twente News Corpus**, a 300M words corpus of Dutch newspaper texts. Lemma’s have been used to get a better generalization, and the frequencies of the adjectives have been logarithmically smoothed. For the *n* most frequent nouns, vectors have been created that contain the frequency of the *m* most frequent adjectives. In most experiments, this boils down to 5.000 nouns being clustered with 20.000 adjectives.

The singular value decompositions have been calculated by SVDPACK (Berry, 1992), a package that is able to handle sparse matrices efficiently and quickly (depending on the number of singular values one wants to retain). The 5.000 by 20.000 matrix can easily be decomposed into 300 principal components within a quarter of an hour on a high-end UNIX workstation.

3 Results

3.1 Evaluation Framework

For the evaluation, two different evaluation frameworks have been used. In the first evaluation framework, the relations that exist in each cluster are compared to the close semantic relationships that exist in Eurowordnet. The close semantic relationships that are used are *synonyms*, *hyponyms*, *hypernyms* and *co-hyponyms*. For the central word of a cluster, those relations are extracted from Wordnet, and it is checked how many of those are also present in the cluster. The precision is then the number of words in the cluster that are also present in the Eurowordnet relations.

The second evaluation uses the Wu & Palmer similarity metric[1] (Wu & Palmer, 1994). Instead of calculating precision over a fixed group of closely related words, the general similarity according to Eurowordnet is calculated. In this evaluation, the similarity for every pair of words in a cluster is calculated, and the average is taken. The global similarity is then the global average of all average cluster similarities.

3.2 Comparison of Original Clustering and Dimensionality Reduction

Table 1 shows per number of components retained, the amount of variance explained by that number of components, and the precision and similarity reached. Figure 2 shows the effects of dimension reduction graphically.

# comp.	var. (%)	prec. (%)	sim. (%)	# comp.	var. (%)	prec. (%)	sim. (%)
10	9.5	15.9	39.1	2000	86.8	40.8	57.5
50	23.1	28.0	48.4	3000	93.9	40.8	57.2
100	31.5	31.6	50.9	4000	98.0	40.8	56.9
500	58.6	38.6	57.1	5000	100.0	41.2	57.0
1000	72.7	39.5	57.4	orig	100.0	43.3	57.7
1300	78.2	40.3	57.9	baseline	–	4.0	9.0

TAB. 1 – Evaluation : explained variance, precision and Wu & Palmer similarity per # components

Clearly, clustering with a dimensionality reduction still yields results that are much better than the baseline, but the results are never better than the results yielded by the clustering with the original data (the similarity at 1300 components being a non-significant exception). What is striking, is the steep rise in both evaluation measures, even when a considerably low number of components is used.

The similarity measure levels out at 500 components. Afterwards, there is practically no increase. The precision measure shows the same movement, though still slightly increasing after 500 components.

[1] This evaluation has been implemented in Prolog by Gosse Bouma.

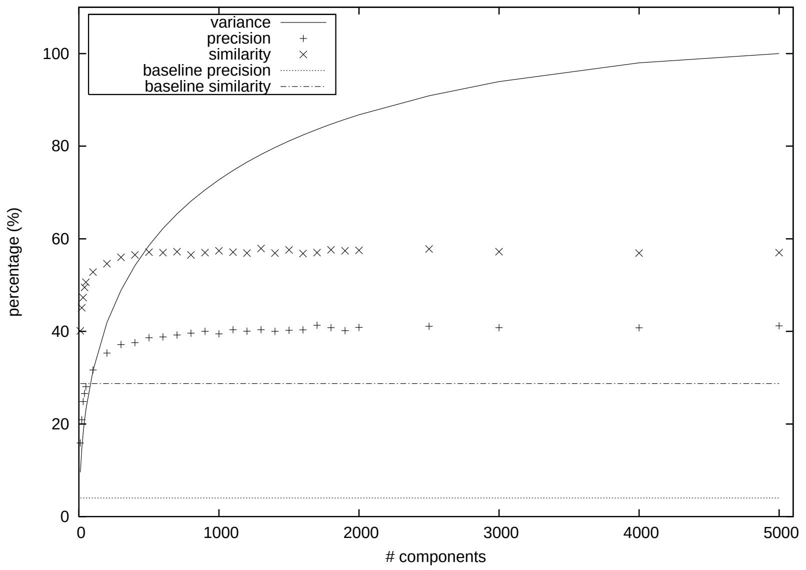

FIG. 2 – Evaluation : explained variance, precision and Wu & Palmer similarity per # components

3.3 Interpretation

The results of a clustering with dimensionality reduction are never better than the original clustering. The singular value decomposition is not able to apply any noise reduction, which is the case in latent semantic analysis (where the dimensionality reduction gives much better results, (Landauer *et al.*, 1998)). On the other hand, the dimension reduction quickly reaches a level of precision and similarity that is comparable to the original clustering. The steep rise shows that the first components capture sound generalizations, that explain a lot of intrinsic data variance.

But the algorithm is not able to find 'real' latent semantic dimensions. This is probably due to the characteristics of adjectives : adjectives mostly do not have a narrow, well-defined meaning. Rather, they have a qualifying semantic meaning. Adjectives are mostly used to qualify nouns, and they can combine with many different nouns. Accordingly, a dimension reduction in adjective space is not able to capture latent semantic dimensions of the kind produced by LSA.

4 Conclusion and Future Work

The results of a dimensionality reduction in adjective space are double-edged. On the one hand, the dimensionality reduction does not bring about any improvement with regard to noun clustering. On the other hand, the algorithm is able to find sound generalizations even with only a few components, which makes it possible to group together semantically related nouns with a reasonable amount of precision and similarity.

But the generalizations that are captured by the algorithm are in no way comparable to the 'latent semantic dimensions' found in LSA. This is probably due to the intrinsic characteristics of adjectives : their qualifying character makes them hard to group together in specific latent semantic dimensions; therefore, the algorithm finds more general adjective dimensions that are only able to capture general semantic distinctions among nouns.

These preliminary results seem to indicate that singular value decomposition should be applied to carefully chosen data : (specific) latent semantic dimensions need to be present in the data. Generalizations in adjective space seem too general. Therefore, clustering with the original data yields better results.

However, the fact that the SVD algorithm is able to capture certain generalizations, is quite promising. It seems that the data that is used (namely adjectives collocating with nouns) is on its own not fit enough to undergo a dimensionality reduction. Taking into account other syntactic relations (such as subject-verb and verb-object relations) might yield more specific latent semantic dimensions, and therefore better results. This approach remains to be investigated.

Although noun-adjective matrices are not perfectly suited to apply SVD to, the SVD algorithm do is able to capture sound generalizations. When the adjective space is extended with other syntactic relations, and even combined with LSA, latent semantic dimensions might emerge that are able to make semantic generalizations beyond the results yielded by the original data.

Références

BERRY M. (1992). Large scale singular value computations. *International Journal of Supercomputer Applications*, **6**(3), 13–49.

HARRIS Z. (1985). Distributional structure. In J. J. KATZ, Ed., *The Philosophy of Linguistics*, p. 26–47. Oxford University Press.

JAIN A. K., MURTY M. N. & FLYNN P. (1999). Data clustering : a review. *ACM Computing Surveys*, **31**(3), 264–323.

LANDAUER T., FOLTZ P. & LAHAM D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Processes*, **25**, 295–284.

MANNING C. & SCHÜTZE H. (2000). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachussets : MIT Press.

PANTEL P. & LIN D. (2002). Discovering word senses from text. *Proceedings of ACM SIGKDD02*.

SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.

VAN DE CRUYS T. (2005). Semantic Clustering in Dutch. An inquiry into the possibilities of using machine learning techniques to automatically acquire semantic classes for nouns in Dutch. Internship report.

VAN DER PLAS L. & BOUMA G. (2005). Syntactic contexts for finding semantically similar words. *Computational Linguistics in the Netherlands 2004. Selected Papers from the Fifteenth CLIN Meeting*, p. 173–184.

VOSSEN P. *et al.* Eurowordnet, building a multilingual database with wordnets for several european languages.

WEEDS J. (2003). *Measures and Applications of Lexical Distributional Similarity*. University of Sussex : PhD Thesis.

WU Z. & PALMER M. (1994). Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, p. 133 –138, New Mexico State University, Las Cruces, New Mexico.