Relatedness Curves for Acquiring Paraphrases

Georgiana Dinu
Saarland University
Saarbruecken, Germany
dinu@coli.uni-sb.de

Grzegorz Chrupała
Saarland University
Saarbruecken, Germany
gchrupala@lsv.uni-saarland.de

Abstract

In this paper we investigate methods for computing similarity of two phrases based on their relatedness scores across *all* ranks k in a SVD approximation of a phrase/term co-occurrence matrix. We confirm the major observations made in previous work and our preliminary experiments indicate that these methods can lead to reliable similarity scores which in turn can be used for the task of paraphrasing.

1 Introduction

Distributional methods for word similarity use large amounts of text to acquire similarity judgments based solely on co-occurrence statistics. Typically each word is assigned a representation as a point in a high dimensional space, where the dimensions represent contextual features; following this, vector similarity measures are used to judge the meaning relatedness of words. One way to make these computations more reliable is to use Singular Value Decomposition (SVD) in order to obtain a lower rank approximation of an original co-occurrence matrix.

SVD is a matrix factorization method which has applications in a large number of fields such as signal processing or statistics. In natural language processing methods such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990) use SVD to obtain a factorization of a (typically) word/document co-occurrence matrix. The underlying idea in these models is that the dimensionality reduction will produce meaningful dimensions which represent concepts rather than just terms, rendering similarity measures on these vectors more accurate. Over the years, it has been shown that these methods can closely match human similarity judgments and that they can be used in various applications such as information retrieval, document classification, essay grading etc. However it has been noted that the success of these methods is drastically determined by the choice of dimension k to which the original space is reduced.

(Bast and Majumdar, 2005) investigates exactly this aspect and proves that no fixed choice of dimension is appropriate. The authors show that two terms can be reliably compared only by investigating the curve of their relatedness scores over all dimensions k. The authors use a term/document matrix and analyze relatedness curves for inducing a hard related/not-related decision and show that their algorithms significantly improve over previous methods for information retrieval.

In this paper we investigate: 1) how the findings of (Bast and Majumdar, 2005) carry over to acquiring paraphrases using SVD on a *phrase/term* co-occurrence matrix and 2) if reliable similarity scores can be obtained from the analysis of relatedness curves.

2 Background

2.1 Singular Value Decomposition

Models such as LSA use Singular Value Decomposition, in order to obtain term representations over a space of *concepts*.

Given a co-occurrence matrix X of size (t, d), we can compute the singular value decomposition: $U\Sigma V^T$ of rank r. Matrices U and V^T of sizes (t, r) and (r, d) are the left and right singular vectors; Σ is the $(r,\ r)$ diagonal matrix of singular values (ordered in descending order)[1]. Similarity between terms i and j is computed as the scalar product between the two vectors associated to the words in the U matrix:

$$sim(u_i, u_j) = \Sigma_{l=1}^{k} u_{il} u_{jl}$$

[1] Any approximation of rank $k < r$ can simply be obtained from an approximation or rank r by deleting rows and columns.

Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics, ACL 2010, pages 27–32,
Uppsala, Sweden, 16 July 2010. ©2010 Association for Computational Linguistics

2.2 Relatedness curves

Finding the optimal dimensionality k has proven to be an extremely important and not trivial step. (Bast and Majumdar, 2005) show that no single cut dimension is appropriate to compute the similarity of two terms but this should be deduced from the curve of similarity scores over all dimensions k. The curve of relatedness for two terms u_i and u_j is given by their scalar product across all dimensions k, k smaller than a rank r:

$$k \rightarrow \Sigma_{l=1}^{k} u_{il}u_{jl},\ for\ k = 1, ..., r$$

They show that a smooth curve indicates closely related terms, while a curve exhibiting many direction changes indicates unrelated terms; the actual values of the similarity scores are often misleading, which explains why a good cut dimension k is so difficult to find.

2.3 Vector space representation of phrases

We choose to apply this to acquiring paraphrases (or inference rules, i.e. entailments which hold in just one direction) in the sense of DIRT (Lin and Pantel, 2001).

In the DIRT algorithm a phrase is a noun-ending path in a dependency graph and the goal is to acquire inference rules such as *(X solve Y, X find solution to Y)*. We will call dependency paths $patterns$. The input data consists of large amounts of parsed text, from which patterns together with X-filler and Y-filler frequency counts are extracted.

In this setting, a pattern receives two vector representation, one in a **X**-filler space and one in the **Y**-filler space. In order to compute the similarity between two patterns, these are compared in the X space and in the Y space, and the two resulting scores are multiplied. (The DIRT algorithm uses Lin measure for computing similarity, which is given in Section 4). Obtaining these vectors from the frequency counts is straightforward and it is exemplified in Table 1 which shows a fragment of a Y-filler DIRT-like vector space.

	..	*case*	*problem*	..
(*X solve Y*, ***Y***)	..	6.1	4.4	..
(*X settle Y*, ***Y***)	..	5.2	5.9	..

Table 1: DIRT-like vector representation in the **Y**-filler space. The values represent mutual information.

3 Relatedness curves for acquiring paraphrases

3.1 Setup

We parsed the XIE fragment of GigaWord (approx. 100 mil. tokens) with Stanford dependency parser. From this we built a pattern/word matrix of size (85000, 3000) containing co-occurrence data of the most frequent patterns with the most frequent words[2]. We perform SVD factorization on this matrix of rank $k = 800$. For each pair of patterns, we can associate two relatedness curves: a X curve and Y curve given by the scalar products of their vectors in the U matrix, across dimensions $k : 1, ..., 800$.

3.2 Evaluating smoothness of the relatedness curves

In Figure 1 we plotted the X and Y curves of comparing the pattern $X \xleftarrow{subj} win \xrightarrow{dobj} Y$ with itself.

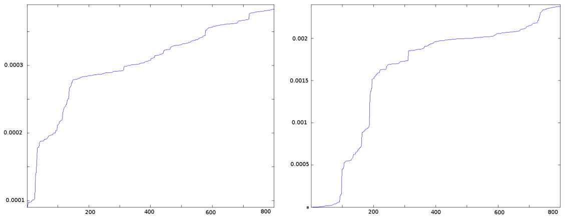

Figure 1: **X-filler** and **Y-filler** relatedness curves for the identity pair $(X \xleftarrow{subj} win \xrightarrow{dobj} Y, X \xleftarrow{subj} win \xrightarrow{dobj} Y)$

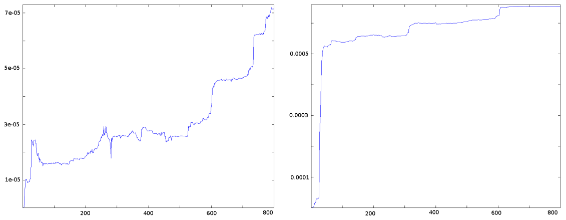

Figure 2: **X-filler** and **Y-filler** relatedness curves for $(X \xleftarrow{subj} leader \xrightarrow{prp} of \xrightarrow{pobj} Y, X \xleftarrow{pobj} by \xleftarrow{prp} lead \xrightarrow{subj} Y)$

Normally, the X and Y curves for the identical pair are monotonically increasing. However what can be noticed is that the actual values of these functions differ by one order of magnitude in the X and in the Y curves of identical patterns, showing that in themselves they are not a good indica-

[2] Even if conceptually we have two semantic spaces (given by X-fillers and Y-fillers), in reality we can work with a single matrix, containing for each pattern also its reverse, both represented solely in a X-filler space

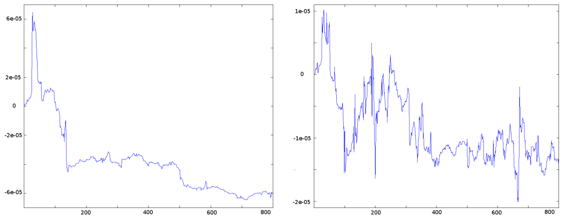

Figure 3: **X-filler** and **Y-filler** relatedness curves for $(X \xleftarrow{subj} win \xrightarrow{dobj} Y, X \xleftarrow{subj} murder \xrightarrow{dobj} Y)$

tor of similarity. In Figure 2 we investigate a pair of closely related patterns: $(X \xleftarrow{subj} leader \xrightarrow{prp} of \xrightarrow{pobj} Y, X \xleftarrow{pobj} by \xleftarrow{prp} lead \xrightarrow{subj} Y)$. It can be noticed that while still not comparable to those of the identical pair, these curves are much smoother than the ones associated to the pair of unrelated patterns in Figure 3[3].

However, unlike in the information retrieval scenario in (Bast and Majumdar, 2005), for which a hard related/not-related assignment works best, for acquiring paraphrases we need to quantify the smoothness of the curves. We describe two functions for evaluating curve smoothness which we will use to compute scores in X-filler and Y-filler semantic spaces.

Smooth function 1 This function simply computes the number of changes in the direction of the curve, as the percentage of times the scalar product increases or remains equal from step l to step $l+1$:

$$CurveS1(u_i, u_j) = \frac{\Sigma_{u_{il}u_{jl} \geq 0} 1}{k},\ l = 1, ..., k$$

An increasing curve will be assigned the maximal value 1, while for a curve that is monotonically decreasing the score will be 0.

Smooth function 2 (Bast and Majumdar, 2005) The second smooth function is given by:

$$CurveS2(u_i, u_j) = \frac{max - min}{\Sigma_{l=1}^{k} abs(u_{il}u_{jl})}$$

where max and min are the largest and smallest values in the curves. A curve which is always increasing or always decreasing will get a score of 1. Unlike the previous method this function is sensitive to the absolute values in the drops of a curve. A curve with large drops, irrelevant of their cardinality, will be penalized by being assigned a low score.

4 Experimental results

In order to compute the similarity score between two phrases, we follow (Lin and Pantel, 2001) and compute two similarity scores, corresponding to the X-fillers and Y-fillers, and multiply them. Given a similarity function, any pattern encountered in the corpus can be paraphrased by returning its most similar patterns.

We implement five similarity functions on the data we have described in the previous section. The first one is the **DIRT** algorithm and it is the only method using the original co-occurrence matrix in which raw counts are replaced by point-wise mutual information scores.

DIRT method The similarity function for two vectors p_i and p_j is:

$$\mathrm{sim}_{\mathrm{Lin}}(p_i, p_j) = \frac{\sum_{l \in I(p_i) \cap I(p_j)} (p_{il} + p_{jl})}{\sum_{l \in I(p_i)} p_{il} + \sum_{l \in I(p_j)} p_{jl}}$$

where values in p_i and p_j are point-wise mutual information, and $I(\cdot)$ gives the indices of non-negative values in a vector.

Methods on SVD factorization All these methods perform computations the (85000, 800) U matrix in the SVD factorization. On this we implement two methods which do an arbitrary dimension cut of $k = 600$: 1) **SP-600** (scalar product) and 2) **COS-600** (cosine similarity). The other two algorithms: **CurveS1** and **CurveS2** use the two curve smoothness functions in Section 3.2; the curves plot the scalar product corresponding to the two patterns, from dimension 1 to 800.

Data In these preliminary experiments we limit ourselves to paraphrasing a set of patterns extracted from a subset of the TREC02-TREC06 question answering tracks. From these questions we extracted and paraphrased the most frequently occurring 20 patterns. Since judging the correctness of these paraphrases ”out-of-context” is rather difficult we limit ourselves to giving examples and analyzing errors made on this data; important observations can be clearly made this way, however in future work we plan to build a proper evaluation setting (e.g. task-based or instance-based in the sense of (Szpektor et al., 2007)) for

[3] The drop out dimension discussed in (Bast and Majumdar, 2005) Section 3, does not seem to exist for our data. This is to be expected since this result stems from a definition of perfectly related terms which is adapted to the particularities of term/document matrices, and not of term/term matrices.

a more detailed analysis of the performance on the methods discussed.

4.1 Results

We list the paraphrases obtained with the different methods for the pattern $X \xleftarrow{subj} show \xrightarrow{dobj} Y$. This pattern has been chosen out of the total set due to its medium difficulty in terms of paraphrasing; some of the patterns in our list are relatively accurately paraphrased by all methods, such as win, while others such as $marry$ are almost impossible to paraphrase, for all methods. In Table 2 we list the top 10 expansions returned by the four methods using the SVD factorization. In bold we mark correct patterns, which we consider to be patterns for which there is a context in which the entailment holds in at least one direction.

As it is clearly reflected in this example the SP-600 is much worse than any of the curve analysis methods; however using cosine as similarity measure at the same arbitrarily chosen dimension (COS-600) brings major improvements.

The two curve smoothness methods exhibit a systematic difference between them. In this example, and also across all 20 instances we have considered, CurveS1 ranks as most similar, a large variety of patterns with the same lexical root (in which, of course, syntax is often incorrect). Only following this we can find patterns expressing lexical variations; these again will be present in many syntactic variations. This sets CurveS1 apart from both CurveS2 and from COS-600 methods. These latter two methods, although conceptually different seem to exhibit surprisingly similar behavior. The behavior of CurveS1 smoothing method is difficult to judge without a proper evaluation; it can be the case that the errors (mostly in syntactic relations) are indeed errors of the algorithm or that the parser introduces them already in our input data.

Table 3 shows the top 10 paraphrases returned by the DIRT algorithm. The DIRT paraphrases are rather accurate, however it is interesting to observe that DIRT and SVD methods can extract different paraphrases. Table 4 gives examples of correct paraphrases which are identified by DIRT but not CurveS2 and the other way around. This seems to indicate that these algorithms do capture different aspects of the data and can be combined for better results. An important aspect here is the fact that obtaining highly accurate paraphrases at the cost of losing coverage is not particularly difficult[4] however not very useful. Previous work such as (Dinu and Wang, 2009) has shown that for these resources, the coverage is a rather important aspect, since they have to capture the great variety of ways in which a meaning can be expressed in different contexts.

DIRT
$\xleftarrow{\mathbf{subj}} \mathbf{reflect} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{\mathbf{subj}} \mathbf{indicate} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{\mathbf{subj}} \mathbf{demonstrate} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{\mathbf{pobj}} \mathbf{in} \xleftarrow{\mathbf{prp}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{pobj} to \xleftarrow{prp} show \xrightarrow{dobj}$
$\xleftarrow{\mathbf{subj}} \mathbf{represent} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{prp}} \mathbf{in} \xrightarrow{\mathbf{pobj}}$
$\xleftarrow{\mathbf{subj}} \mathbf{display} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{subj} bring \xrightarrow{dobj}$
$\xleftarrow{\mathbf{pobj}} \mathbf{with} \xleftarrow{\mathbf{prp}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$

Table 3: Top 10 paraphrases for $X \xleftarrow{subj} show \xrightarrow{dobj} Y$

CurveS2	**DIRT**
$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$	
$\xleftarrow{pobj} in \xleftarrow{prp} indicate \xrightarrow{dobj}$ $\xleftarrow{pobj} in \xleftarrow{prp} reflect \xrightarrow{dobj}$ $\xleftarrow{dobj} interpret \xrightarrow{prp} as \xrightarrow{pobj}$	$\xleftarrow{subj} display \xrightarrow{dobj}$ $\xleftarrow{subj} confirm \xrightarrow{dobj}$ $\xleftarrow{subj} point \xrightarrow{prp} to \xrightarrow{pobj}$
$\xleftarrow{\mathbf{subj}} \mathbf{win} \xrightarrow{\mathbf{dobj}}$	
$\xleftarrow{subj} vie \xrightarrow{prp} for \xrightarrow{pobj}$ $\xleftarrow{subj} compete \xrightarrow{prp} for \xrightarrow{pobj}$ $\xleftarrow{subj} secure \xrightarrow{dobj}$	$\xleftarrow{pos} victory \xrightarrow{prp} in \xrightarrow{pobj}$ $\xleftarrow{subj} win \xrightarrow{dobj} title \xrightarrow{nn}$ $\xrightarrow{appos} winner \xrightarrow{nn}$
$\xleftarrow{\mathbf{subj}} \mathbf{enter} \xrightarrow{\mathbf{dobj}}$	
$\xleftarrow{subj} march \xrightarrow{prp} into \xrightarrow{pobj}$ $\xleftarrow{subj} advance \xrightarrow{prp} into \xrightarrow{pobj}$ $\xleftarrow{pos} entry \xrightarrow{prp} to \xrightarrow{pobj}$	$\xleftarrow{subj} start \xrightarrow{prp} in \xrightarrow{pobj}$ $\xleftarrow{subj} play \xrightarrow{prp} in \xrightarrow{pobj}$ $\xleftarrow{subj} join \xrightarrow{prp} in \xrightarrow{pobj}$

Table 4: Example of paraphrases (i.e. ranked in the top 30) identified by one method and not the other

4.2 Discussion

In this section we attempt to get more insight into the way the relatedness curves relate to the intuitive notion of similarity, by examining curves of incorrect paraphrases extracted by our methods.

The first error we consider, is the pattern $X \xleftarrow{pos} confidence \xleftarrow{pobj} of \xleftarrow{prp} Y$ which is judged as being very similar to $show$ by SP-600, COS-600 as well as CurveS2. Figure 4 shows the relatedness curves. As it can be noticed, both the X and Y similarities grow dramatically around dimension

[4] High precision can be very easily achieved simply by intersecting the sets of paraphrases returned by two or more of the methods implemented

SP-600	**COS-600**	**CurveS1**	**CurveS2**
$\xleftarrow{pos} \mathit{confidence} \xleftarrow{pobj} \mathit{of} \xleftarrow{prp}$	$\xleftarrow{\mathbf{subj}} \mathbf{indicate} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{prp}} \mathbf{in} \xrightarrow{\mathbf{pobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{indicate} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{subj} \mathit{boost} \xrightarrow{dobj} \mathit{rate} \xrightarrow{nn}$	$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{prp}} \mathbf{of} \xrightarrow{\mathbf{pobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{indicate} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{reflect} \xrightarrow{\mathbf{dobj}}$
$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{prp}} \mathbf{of} \xrightarrow{\mathbf{pobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{represent} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{subj} \mathit{show} \xrightarrow{prp} \mathit{with} \xrightarrow{pobj}$	$\xleftarrow{\mathbf{subj}} \mathbf{represent} \xrightarrow{\mathbf{dobj}}$
$\xrightarrow{prp} \mathit{to} \xrightarrow{pobj} \mathit{percent} \xrightarrow{nn}$	$\xleftarrow{\mathbf{pobj}} \mathbf{by} \xleftarrow{\mathbf{prp}} \mathbf{show} \xleftarrow{\mathbf{partmod}}$	$\xleftarrow{\mathbf{pobj}} \mathbf{with} \xleftarrow{\mathbf{prp}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{subj} \mathit{bring} \xrightarrow{dobj} \mathit{rate} \xrightarrow{nn}$
$\xleftarrow{subj} \mathit{total} \xrightarrow{dobj} \mathit{yuan} \xrightarrow{appos}$	$\xleftarrow{\mathbf{pobj}} \mathbf{in} \xleftarrow{\mathbf{prp}} \mathbf{reflect} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{subj} \mathit{show} \xrightarrow{tmod}$	$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{prp}} \mathbf{of} \xrightarrow{\mathbf{pobj}}$
$\xleftarrow{subj} \mathit{hit} \xrightarrow{dobj} \mathit{dollar} \xrightarrow{appos}$	$\xleftarrow{pos} \mathit{confidence} \xleftarrow{pobj} \mathit{of} \xleftarrow{prp}$	$\xleftarrow{subj} \mathit{show} \xrightarrow{prp} \mathit{despite} \xrightarrow{pobj}$	$\xleftarrow{\mathbf{dobj}} \mathbf{interpret} \xrightarrow{\mathbf{prp}} \mathbf{as} \xrightarrow{\mathbf{pobj}}$
$\xleftarrow{subj} \mathit{reach} \xrightarrow{dobj} \mathit{dollar} \xrightarrow{appos}$	$\xleftarrow{\mathbf{pobj}} \mathbf{by} \xleftarrow{\mathbf{prp}} \mathbf{reflect} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{pobj} \mathit{during} \xleftarrow{prp} \mathit{show} \xrightarrow{dobj}$	$\xleftarrow{pos} \mathit{confidence} \xleftarrow{pobj} \mathit{of} \xleftarrow{prp}$
$\xleftarrow{subj} \mathit{slash} \xrightarrow{dobj} \mathit{rate} \xrightarrow{nn}$	$\xleftarrow{\mathbf{pobj}} \mathbf{in} \xleftarrow{\mathbf{prp}} \mathbf{indicate} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{pobj}} \mathbf{in} \xleftarrow{\mathbf{prp}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{subj}} \mathbf{show} \xrightarrow{\mathbf{dobj}} \mathbf{rate} \xrightarrow{\mathbf{nn}}$
$\xleftarrow{nn} \mathit{confidence} \xleftarrow{pobj} \mathit{of} \xleftarrow{prp}$	$\xleftarrow{\mathbf{subj}} \mathbf{reflect} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{pobj}} \mathbf{by} \xleftarrow{\mathbf{prp}} \mathbf{show} \xleftarrow{\mathbf{partmod}}$	$\xleftarrow{subj} \mathit{put} \xrightarrow{dobj} \mathit{rate} \xrightarrow{nn}$
$\xleftarrow{subj} \mathit{raise} \xrightarrow{dobj} \mathit{rate} \xrightarrow{nn}$	$\xleftarrow{subj} \mathit{interpret} \xrightarrow{prp} \mathit{as} \xrightarrow{pobj}$	$\xleftarrow{\mathbf{pobj}} \mathbf{on} \xleftarrow{\mathbf{prp}} \mathbf{show} \xrightarrow{\mathbf{dobj}}$	$\xleftarrow{\mathbf{pobj}} \mathbf{by} \xleftarrow{\mathbf{prp}} \mathbf{show} \xleftarrow{\mathbf{partmod}}$

Table 2: Top 10 paraphrases for $X \xleftarrow{subj} show \xrightarrow{dobj} Y$

500. Therefore the scalar product will be very high at cut point 600, leading to methods' SP-600 and COS-600 error. However the two curve methods are sensitive to the shape of the relatedness curves. Since CurveS2 is sensitive to actual drop values in these curves, this pair will still be ranked very similar. The curves do decrease by small amounts in *many* points which is why method CurveS1 does score these two patterns as very similar.

An interesting point to be made here is that, this pair is ranked similar by three methods out of four because of the dramatic increase in relatedness at around dimension 500. However, intuitively, such an increase should be more relevant at earlier dimensions, which correspond to the larger eigenvalues, and therefore to the most relevant concepts. Indeed, in the data we have analyzed, highly similar patterns exhibit large increases at earlier (first 100-200) dimensions, similarly to the examples given in Figure 1 and Figure 2. This leads us to a particular aspect that we would like to investigate in future work, which is to analyze the behavior of a relatedness curve in relation to relevance weights obtained from the eigenvalues of the matrix factorization.

In Figure 5 we plot a second error, the relatedness curves of *show* with $X \xleftarrow{subj} boost \xrightarrow{dobj} rate \xrightarrow{nn} Y$ which is as error made only by the SP-600 method. The similarity reflected in curve Y is relatively high (given by the large overlap of Y-filler $interest$), however we obtain a very high X similarity only due to the peak of the scalar product exactly around the cut dimension 600.

5 Conclusion

In this paper we have investigated the relevance of judging similarity of two phrases across all ranks k in a SVD approximation of a phrase/term co-occurrence matrix. We confirm the major observations made in previous work and our preliminary experiments indicate that reliable similarity scores for paraphrasing can be obtained from the analysis of relatedness scores across all dimensions.

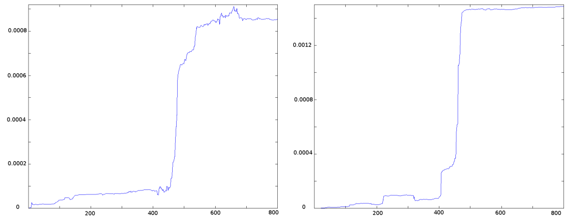

Figure 4: **X-filler** and **Y-filler** relatedness curves for $(X \xleftarrow{subj} show \xrightarrow{dobj} Y, X \xleftarrow{pos} confidence \xleftarrow{pobj} of \xleftarrow{prp} Y)$

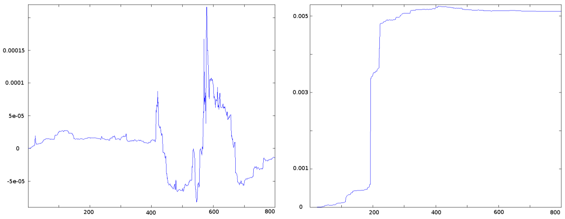

Figure 5: **X-filler** and **Y-filler** relatedness curves for $(X \xleftarrow{subj} show \xrightarrow{dobj} Y, X \xleftarrow{subj} boost \xrightarrow{dobj} rate \xrightarrow{nn} Y)$

In the future we plan to 1) use the observations we have made in Section 4.2 to focus on identifying good curve-smoothness functions and 2) build an appropriate evaluation setting in order to be able to accurately judge the performance of the methods we propose.

Finally, in this paper we have investigated these aspects for the task of paraphrasing in a particular setting, however our findings can be applied to any vector space method for semantic similarity.

References

Scott C. Deerwester and Susan T. Dumais and Thomas K. Landauer and George W. Furnas and Richard A. Harshman 1990. *Indexing by Latent Semantic Analysis In JASIS.*

Bast, Holger and Majumdar, Debapriyo. 2005. Why spectral retrieval works. *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. *In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Georgiana Dinu and Rui Wang. 2009. Inference rules and their application to recognizing textual entailment. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*.

Idan Szpektor and Eyal Shnarch and Ido Dagan 2007. Instance-based Evaluation of Entailment Rule Acquisition. *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.