

# Computing Word Senses by Semantic Mirroring and Spectral Graph Partitioning

**Martin Fagerlund**  
Linköping University

Linköping, Sweden  
marfa229@student.liu.se

**Magnus Merkel**  
Linköping University

Linköping, Sweden  
magnus.merkel@liu.se

**Lars Eldén**  
Linköping University

Linköping, Sweden  
lars.elden@liu.se

**Lars Ahrenberg**  
Linköping University

Linköping, Sweden  
lars.ahrenberg@liu.se

## Abstract

Using the technique of "semantic mirroring" a graph is obtained that represents words and their translations from a parallel corpus or a bilingual lexicon. The connectedness of the graph holds information about the different meanings of words that occur in the translations. Spectral graph theory is used to partition the graph, which leads to a grouping of the words according to different senses. We also report results from an evaluation using a small sample of seed words from a lexicon of Swedish and English adjectives.

## 1 Introduction

A great deal of linguistic knowledge is encoded implicitly in bilingual resources such as parallel texts and bilingual dictionaries. Dyvik (1998, 2005) has provided a knowledge discovery method based on the semantic relationship between words in a source language and words in a target language, as manifested in parallel texts. His method is called Semantic mirroring and the approach utilizes the way that different languages encode lexical meaning by mirroring source words and target words back and forth, in order to establish semantic relations like synonymy and hyponymy. Work in this area is strongly related to work within Word Sense Disambiguation (WSD) and the observation that translations are a good source for detecting such distinctions (Resnik & Yarowsky 1999, Ide 2000, Diab & Resnik 2002). A word that has multiple meanings in one language is likely to have different translations in other languages. This means that translations serve as sense indicators for a particular source

word, and make it possible to divide a given word into different senses.

In this paper we propose a new graph-based approach to the analysis of semantic mirrors. The objective is to find a viable way to discover synonyms and group them into different senses. The method has been applied to a bilingual dictionary of English and Swedish adjectives.

## 2 Preparations

### 2.1 The Translation Matrix

In these experiments we have worked with a English-Swedish lexicon consisting of 14850 English adjectives, and their corresponding Swedish translations. Out of the lexicon was created a translation matrix  $\mathbf{B}$ , and two lists with all the words, one for English and one for Swedish.  $\mathbf{B}$  is defined as

$$\mathbf{B}(i,j) = \begin{cases} 1, & \text{if } i \sim j, \\ 0, & \text{otherwise.} \end{cases}$$

The relation  $i \sim j$  means that word  $i$  translates to word  $j$ .

### 2.2 Translation

Translation is performed as follows. From the word  $i$  to be translated, we create a vector  $\bar{e}_i$ , with a one in position  $i$ , and zeros everywhere else. Then perform the matrix multiplication  $\mathbf{B}\bar{e}_i$  if it is a Swedish word to be translated, or  $\mathbf{B}^T\bar{e}_i$  if it is an English word to be translated.  $\bar{e}_i$  has the same length as the list in which the word  $i$  can be found.

## 3 Semantic Mirroring

We start with an English word, called *engl*<sup>1</sup>. We look up its Swedish translations. Then we look up

<sup>1</sup>Short for english1. We will use swe for Swedish words.

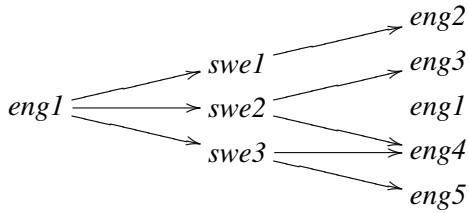
the English translations of each of those Swedish words. We have now performed one "mirror-operation". In mathematical notation:

$$f = \mathbf{B}\mathbf{B}^T\bar{e}_{eng1}.$$

The non-zero elements in the vector  $f$  represent English words that are semantically related to  $eng1$ . Dyvik (1998) calls the set of words that we get after two translations the *inverse t-image*. But there is one problem. The original word should not be here. Therefore, in the last translation, we modify the matrix  $\mathbf{B}$ , by replacing the row in  $\mathbf{B}$  corresponding to  $eng1$ , with an all-zero row. Call this new modified matrix  $\mathbf{B}_{mod1}$ . So instead of the matrix multiplication performed above, we start over with the following one:

$$\mathbf{B}_{mod1}\mathbf{B}^T\bar{e}_{eng1}. \quad (1)$$

To make it clearer from a linguistic perspective, consider the following figure<sup>2</sup>.



The words to the right in the picture above ( $eng2, \dots, eng5$ ) are the words we want to divide into senses. To do this, we need some kind of relation between the words. Therefore we continue to translate, and perform a second "mirror operation". To keep track of what each word in the inverse t-image translates to, we must first make a small modification. We have so far done the operation (1), which gave us a vector, call it  $e \in \mathbb{R}^{14850 \times 1}$ . The vector  $e$  consists of nonzero integers in the positions corresponding to the words in the inverse t-image, and zeros everywhere else. We make a new matrix  $\mathbf{E}$ , with the same number of rows as  $e$ , and the same number of columns as there are nonzeros in  $e$ . Now go through every element in  $e$ , and when finding a nonzero element in row  $i$ , and if it is the  $j$ :th nonzero element, then put a one in position  $(i, j)$  in  $\mathbf{E}$ . The procedure is illustrated in (2).

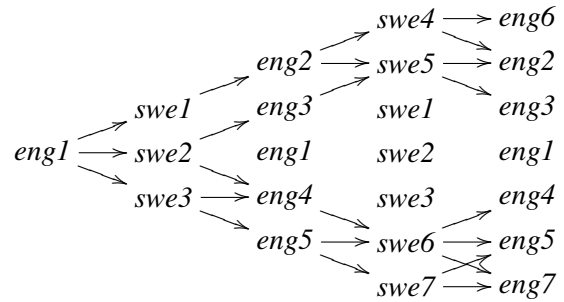
<sup>2</sup>The arrows indicate translation.

$$\begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \\ 0 \\ 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

When doing our second "mirror operation", we do not want to translate through the Swedish words  $swe1, \dots, swe3$ . We once again modify the matrix  $\mathbf{B}$ , this time replacing the columns of  $\mathbf{B}$  corresponding to the Swedish words  $swe1, \dots, swe3$ , with zeros. Call this second modified matrix  $\mathbf{B}_{mod2}$ . With the matrix  $\mathbf{E}$  from (2), we now get:

$$\mathbf{B}_{mod2}\mathbf{B}_{mod2}^T\mathbf{E} \quad (3)$$

We illustrate the operation (3):



Now we have got the desired relation between  $eng2, \dots, eng5$ . In (3) we keep only the rows corresponding to  $eng2, \dots, eng5$ , and get a symmetric matrix  $\mathbf{A}$ , which can be considered as the adjacency matrix of a graph. The adjacency matrix and the graph of our example are illustrated below.

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \quad (4)$$

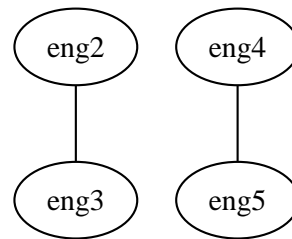


Figure 1: The graph to the matrix in (4).

The adjacency matrix should be interpreted in the following way. The rows and the columns correspond to the words in the inverse t-image. Following our example,  $eng2$  corresponds to row 1 and

column 1, *eng3* corresponds to row 2 and column 2, and so on. The elements on position  $(i, i)$  in  $\mathbf{A}$  are the vertex weights. The vertex weight associated with a word, describes how many translations that word has in the other language, e.g. *eng2* translates to *swe4* and *swe5* that is translated back to *eng2*. So the vertex weight for *eng2* is 2, as also can be seen in position  $(1, 1)$  in (4). A high vertex weight tells us that the word has a high number of translations, and therefore probably a wide meaning.

The elements in the adjacency matrix on position  $(i, j), i \neq j$  are the edge weights. These weights are associated with two words, and describe how many words in the other language that both word  $i$  and  $j$  are translated to. E.g. *eng5* and *eng4* are both translated to *swe6*, and it follows that the weight,  $w(\text{eng4}, \text{eng5}) = 1$ . If we instead would take *eng5* and *eng7*, we see that they both translate to *swe6* and *swe7*, so the weight between those words,  $w(\text{eng5}, \text{eng7}) = 2$ . (But this is not shown in the adjacency matrix, since *eng7* is not a word in the inverse t-image). A high edge weight between two words tells us that they share a high number of translations, and therefore probably have the same meanings.

#### 4 Graph Partitioning

The example illustrated in Figure 1 gave as a result two graphs that are not connected. Dyvik argues that in such a case the graphs represent two groups of words of different senses. In a larger and more realistic example one is likely to obtain a graph that is connected, but which can be partitioned into two subgraphs without breaking more than a small number of edges. Then it is reasonable to ask whether such a partitioning has a similar effect in that it represents a partitioning of the words into different senses.

We describe the mathematical procedure of partitioning a graph into subgraphs, using spectral graph theory (Chung, 1997). First, define the degree  $d(i)$  of a vertex  $i$  to be

$$d(i) = \sum_j A(i, j).$$

Let  $D$  be the diagonal matrix defined by

$$D(i, j) = \begin{cases} d(i), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

The Laplacian  $L$  is defined as

$$L = D - A.$$

We define the normalised Laplacian  $\mathcal{L}$  to be

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}.$$

Now calculate the eigenvalues  $\lambda_0, \dots, \lambda_{n-1}$ , and the eigenvectors of  $\mathcal{L}$ . The smallest eigenvalue,  $\lambda_0$ , is always equal to zero, as shown by Chung (1997). The multiplicity of zero among the eigenvalues is equal to the number of connected components in the graph, as shown by Spielman (2009). We will look at the eigenvector belonging to the second smallest eigenvalue,  $\lambda_1$ . This eigenpair is often referred to as the *Fiedler value* and the *Fiedler vector*. The entries in the Fiedler vector corresponds to the vertices in the graph. (We will assume that there is only one component in the graph. If not, chose the component with the largest number of vertices). Sort the Fiedler vector, and thus sorting the vertices in the graph. Then make  $n - 1$  cuts along the Fiedler vector, dividing the elements of the vector into two sets, and for each cut compute the conductance,  $\phi(S)$ , defined as

$$\phi(S) = d(V) \frac{|\partial(S, \bar{S})|}{d(S)d(\bar{S})}, \quad (5)$$

where  $d(S) = \sum_{i \in S} d(i)$ .  $|\partial(S, \bar{S})|$  is the total weight of the edges with one end in  $S$  and one end in  $\bar{S}$ , and  $V = S + \bar{S}$  is the set of all vertices in the graph. Another measure used is the sparsity,  $sp(S)$ , defined as

$$sp(S) = \frac{|\partial(S, \bar{S})|}{\min(d(S), d(\bar{S}))} \quad (6)$$

For details, see (Spielman, 2009). Choose the cut with the smallest conductance, and in the graph, delete the edges with one end in  $S$  and the other end in  $\bar{S}$ . The procedure is then carried out until the conductance,  $\phi(S)$ , reaches a tolerance. The tolerance is decided by human evaluators, performing experiments on test data.

#### 5 Example

We start with the word *slithery*, and after the mirroring operation (3) we get three groups of words in the inverse t-image, shown in Table 1. After two partitionings of the graph to *slithery*, using the method described in section 4, we get five sense groups, shown in Table 2.

smooth	slimy	saponaceous
slick	smooth-faced	
lubricious	oleaginous	slippy
slippery	oily	
glib	greasy	
sleek		

Table 1: The three groups of words after the mirroring operation.

slimy	glib	oleaginous
smooth-faced	slippery	oily
smooth	lubricious	greasy
sleek	slick	
saponaceous	slippy	

Table 2: The five sense groups of *slithery* after two partitionings.

## 6 Evaluation

A small evaluation was performed using a random sample of 10 Swedish adjectives. We generated sets under four different conditions. For the first, using conductance (5). For the second, using sparsity (6). For the third and fourth, we set the diagonal entries in the adjacency matrix to zero. These entries tell us very little of how the words are connected to each other, but they may effect how the partitioning is made. So for the third, we used conductance and no vertex weights, and for the fourth we used sparsity and no vertex weights. There were only small differences in results due to the conditions, so we report results only for one of them, the one using vertex weights and sparsity.

Generated sets, with singletons removed, were evaluated from two perspectives: consistency and synonymy with the seed word. For consistency a three-valued scheme was used: (i) the set forms a single synset, (ii) at least two thirds of the words form a single synset, and (iii) none of these. Synonymy with the seed word was judged as either yes or no.

Two evaluators first judged all sets independently and then coordinated their judgements. The criterion for consistency was that at least one domain, such as personality, taste, manner, can be found where all adjectives in the set are interchangeable. Results are shown in Table 3.

Depending on how we count partially consistent groups this gives a precision in the range 0.57 to 0.78. We have made no attempt to measure recall.

	Count	Average	Percentage
All groups	58	5.8	100
Consistent groups	33	3.3	57
2/3 consistency	12	1.2	21
Synonymy with seed word	14	1.4	24

Table 3: Classified output with frequencies from one type of partition

It may be noted that group size varies. There are often several small groups with just 2 or 3 words, but sometimes as many as 10-15 words make up a group. For large groups, even though they are not fully consistent, the words tend to be drawn from two or three synsets.

## 7 Conclusion

So far we have performed a relatively limited number of tests of the method. Those tests indicate that semantic mirroring coupled with spectral graph partitioning is a useful method for computing word senses, which can be developed further using refined graph theoretic and linguistic techniques in conjunction.

## 8 Future work

There is room for many more investigations of the approach outlined in this paper. We would like to explore the possibility to have a vertex (word) belong to multiple synsets, instead of having discrete cuts between synsets. In the present solution a vertex belongs to only one partition of a graph, making it impossible to having the same word belong to several synsets. We would also like to investigate the properties of graphs to see whether it is possible to automatically measure how close a seed word is to a particular synset. Furthermore, more thorough evaluations of larger data sets would give us more information on how to combine similar synsets which were generated from distinct seed words and explore more complex semantic fields. In our future research we will test the method also on other lexica, and perform experiments with the different tolerances involved. We will also perform extensive tests assessing the results using a panel of human evaluators.

## References

- Daniel A. Spielman. 2009. *Spectral Graph theory*. Lecture notes.
- Daniel A. Spielman, S. -H. Teng. 2006. *Spectral partitioning works: Planar graphs and finite element meshes*. Elsevier Inc.
- Diab, M. Resnik, P. 2002. *An Unsupervised Method for Word Sense Tagging using Parallel Corpora*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 255-262.
- Fan R. K. Chung. 1997. *Spectral Graph Theory*. American Mathematical Society, Providence, Rhode Island.
- H. Dyvik. 1998. *A Translational Basis for Semantics*. In: Stig Johansson and Signe Oksefjell (eds.): *Corpora and Crosslinguistic Research: Theory, Method and Case Studies*, pp. 51-86. Rodopi.
- H. Dyvik. 2005. *Translations as a Semantic Knowledge Source*. Proceedings of the Second Baltic Conference on Human Language Technologies, Tallinn.
- Nancy Ide. 2000. *Cross-lingual sense determination: Can it work?* Computers and the Humanities: Special issue on SENSEVAL, 34:223–234.
- Philip Resnik , David Yarowsky. *Distinguishing systems and distinguishing senses: new evaluation methods for Word Sense Disambiguation* Natural Language Engineering, v.5 n.2, p.113-133, June 1999