Noname manuscript No.
(will be inserted by the editor)

Computing Semantic Clusters by Semantic Mirroring and Spectral Graph Partitioning

Lars Eldén · Magnus Merkel · Lars Ahrenberg · Martin Fagerlund

the date of receipt and acceptance should be inserted later

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 semantic relations of words that occur in the translations. Spectral graph theory is used to partition the graph, which leads to a grouping of the words in different clusters. We illustrate the method using a small sample of seed words from a lexicon of Swedish and English adjectives and discuss its application to computational lexical semantics and lexicography.

Keywords semantics, cluster, bilingual, dictionary, mirroring, WordNet, graph, spectral, partitioning, adjacency, graph Laplacian, Fiedler vector, matrix, eigenvector, eigenvalue

Mathematics Subject Classification (2010) 94C15,65F15,68R10

1 Introduction

A great deal of linguistic knowledge is encoded implicitly in bilingual resources such as parallel texts and bilingual dictionaries. Dyvik [7,6] 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. This knowledge can then be applied in the creation or extension of wordnets, thesauri and ontologies. Work in this area is also related to work within

L. Eldén · Martin Fagerlund
Department of Mathematics, Linköping University, Sweden
E-mail: lars.elden@liu.se, marfa229@student.liu.se

Magnus Merkel · Lars Ahrenberg
Department of Computer Science, Linköping University, Sweden
E-mail: magnus.merkel@liu.se, lars.ahrenberg@liu.se

Word Sense Disambiguation (WSD) and the observation that translations are a good source for detecting such distinctions [21,13,5]. A word that has multiple meanings in one language is likely to have different translations in other languages. This means that translations can serve as sense indicators for a particular source word, and make it possible to derive different senses for it.

In this paper we propose a method for the analysis of semantic mirrors (a very preliminary version of the method is described in the conference presentation [9]), with the objective to produce synonym sets. The semantic mirroring procedure can be interpreted as the construction of a graph where each node represents a word, and each edge one or more translations via words in the second language. If the graph consists of disconnected subgraphs then it is very likely that those represent different senses. Similarly, if there are two subgraphs, say, that internally are strongly connected but are only weakly connected to one another (via a small number of edges), then it is likely that the subgraphs represent different senses. The method used to find the weak connections in the graph is *spectral partitioning* [19,4]. This method is widely used e. g. for the partitioning of graphs for load balancing in parallel computing [22]. Here we use the ordering of the nodes in the graph produced by the method and search the graph for weak connections. The algorithm is *hierarchical*: the original connected graph is divided into two subgraphs, which are then subdivided, until a specified number of subgraphs is reached.

The outcome of the procedure is two or more sets of words, which we will refer to as semantic clusters. As there are many parameters to the method, including the bilingual resource, the number of graph divisions and the thresholds set on eigenvalues, there is no guarantee that there will be a one-to-one correspondence between groupings and word senses as recognized by, say, a dictionary or a WordNet. On the other hand, there is no universally recognized notion of word sense either, cf. Section 2.1.

In the following section we provide a linguistic background and present the method of Semantic Mirrors in more detail. In section 3 we show how it is implemented as spectral graph partitioning, and, in section 4, we give illustrations of its application to a bilingual dictionary of English and Swedish adjectives. Related work on semantic mirrors is briefly discussed in Section 5. Finally, in Section 6 we discuss how the proposed method can be used in (explorative) lexical semantics and state our conclusions.

1.1 Notation

Graphs are denoted by calligraphic Roman letters. The corresponding weighted adjacency matrices, which will be real throughout, are denoted by capital Roman letters. The standard basis vector with zeros in all positions except the i'th, which holds a 1, is denoted e_i. The vector e has all entries equal to 1. We also denote sets of words by capital Roman letters. For a vector x, $\mathrm{diag}(x)$ denotes the diagonal matrix with the vector elements on the diagonal.

2 Semantic Mirroring

Helge Dyvik introduced translations ("semantic mirrors") as a semantic knowledge source [6, 7]. In a two-way lexicon with translations of a word in one language into different words in the other, and translations back to the first language, there is much information that is only implicitly given. By using this partially hidden information, we can extract an intricate network of translational correspondences, bringing together the vocabularies of the two languages. This makes it possible for us to treat each language as the semantic mirror of the other language, in a way that will be described further on. To motivate the procedure of extracting the information Dyvik makes the following assumptions [8]:

1. Semantically closely related words tend to have strongly overlapping sets of translations.
2. Words with a wide meaning tend to have a higher number of translations, than words with a more narrow meaning.
3. Contrastive ambiguity, i.e. ambiguity between two unrelated senses of a word, tend to be a historically accidental and idiosyncratic property. Therefore, we don't expect to find instances of the same contrastive ambiguity in other words, or by words in other languages.
4. Words with unrelated meanings will not share translations into another language, except in cases where the shared words are contrastively ambiguous between the two unrelated meanings. By assumption (3) there should then be at most one such shared word.

Example 1 To give a first illustration of the ideas of semantic mirroring we consider a small, contrived and somewhat simplified, "manually performed" example. Note that our description differs from that in the papers by Dyvik: In order to keep the example small we only perform one and a half mirroring[1], while Dyvik normally does two (as we will do in the rest of the paper).

Using a publicly available dictionary[2], and a Swedish seed word **rätt**, we get the following translation,

rätt $\longrightarrow$ {course, dish, a lot, meal, proper, right, justice, rightly, somewhat, correct, directly, fair and square, full, law, court, quite, fairly} $=: U_0$.

These are the words that we want to cluster, hoping that the clustering will reveal different senses. We translate each word in U_0 to Swedish and then back to English and note which words in U_0 are returned. For instance, `course` is translated

course $\longrightarrow$ {lopp, kurs, lärokurs, flöde, fat, gång, stråt, väg}.

Out of those Swedish words, only **kurs** and **fat** have a translation in U_0:

kurs $\longrightarrow$ {tack, **course**, class},
fat $\longrightarrow$ {bowl, saucer, plate, barrel, **course, dish**, platter}.

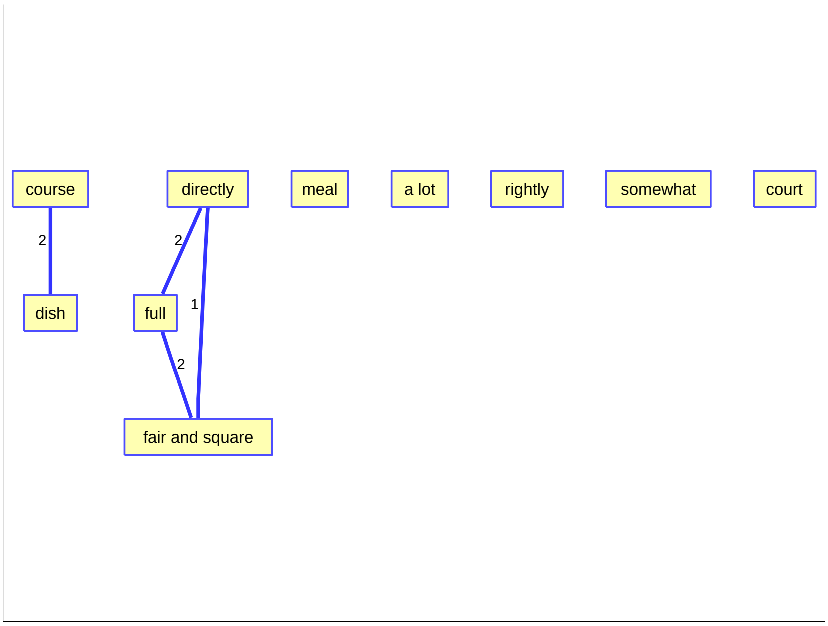
Analogously,

dish $\longrightarrow$ {bunke, fat, behållare, bytta, kärl},

and here again **fat** translates back to **dish, course**.

If we perform all the translations we get the graphs in Figure 1. The edge

[1] Translation Swedish $\rightarrow$ English $\rightarrow$ Swedish $\rightarrow$ English.
[2] `www.gratisordbok.se`.



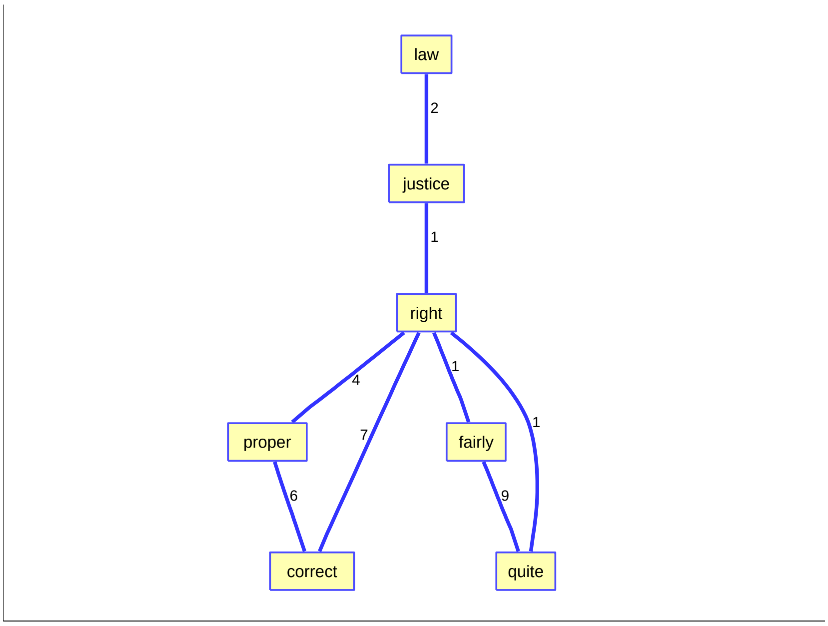


Fig. 1 Translations of the words in U_0: disconnected graphs. The edge weights are the number of Swedish words in the translation between the two node words.

labels denote the number of translations between the English words via Swedish words. It is clear that the disconnected graphs are groups of words with different meanings. We also get the more interesting larger subgraph, where parts of the graphs are rather tight with weaker couplings to the rest. It is apparent that we have three different meanings, or three groups of synonyms.

In the rest of the paper we will describe how by spectral partitioning methods we can partition graphs like the bottom one in Figure 1, and thereby compute groups of words, where each group contains semantically related words.

2.1 Word Senses and Synonym Sets

A word sense is in principle one of the various meanings a word can have. But to actually pinpoint what constitutes a word sense is very difficult, both to determine the exact characteristics of the word sense and what words actually belong to that word sense. As Kilgarriff [14] writes: *Identifying a word's senses is an analytic task for which there are no straightforward answers.* When constructing thesauruses, the objective is primarily to provide consistent meaning clusters, and when compiling dictionaries the objective is to list all the possible word senses for a word entry. Kilgarriff [14] points out that dictionaries often disagree. If we look at the Merriam Webster on-line English dictionary (`www.merriam-webster.com`), there are, for example, 13 different word senses for the adjective **blue** (including subsenses). In WordNet (`wordnetweb.princeton.edu`) the adjective **blue** is found in 8 so called synsets (synonymy sets), each of which represents a word sense, especially since all 8 of them have definitions. The disagreement in the choice of word senses between the two resources is dependent on granularity (if something is divided into subsenses or not) and in coverage (missing sense in one of the resources). For example, the word sense of **blue** depicting “aristocratic” or “blue-blooded” is present in WordNet but not in Merriam Webster. On the other hand there are three word senses present in Merriam Webster that cannot be found in WordNet, for example, “blue states”, describing states that support Democrats in general elections, “blue music” in the music genre blues, and “blue” used about learned, intellectual women.

The disagreement between different lexical resources, such as WordNet and Merriam Webster, could be seen as unfortunate because it points to a lack of consensus and thereby making it difficult to compare new, derived lexical data with any gold standards. A complete list of all word senses or synonym sets is probably not feasible to create, as the potential usage of words is virtually unlimited [17]. Some word senses are of course more or less recognized (with the process known as lexicalization), but others are more dynamic and are created in certain contexts.

In this study a bilingual lexicon of English and Swedish adjectives is used. There are no explicit synonym sets or word senses listed in the dictionary, but the procedure described in this paper has the goal to group words into clusters of words that have identical or related meanings, as far as possible.

2.2 The Mirroring Method Explained

While the method often produces neat word clusters, word meanings do not always correspond one-to-one in two languages, so the mirroring method can impose a semantic structure on the source language that is not inherent to it. Let us see how this can happen.

Denote the words in the source language by $\sigma_1, \sigma_2, \ldots, \sigma_m$, and those in the target language by $\tau_1, \tau_2, \ldots, \tau_n$. For concreteness we will give examples from English and Swedish. The lexical resource that we use, whether a man-made bilingual dictionary or the output from a word alignment system, is assumed to be reasonably complete.

The seed word σ is a word with possibly several senses, $\Xi_1, \Xi_2, \ldots, \Xi_s$. This set is referred to as a *sense space*. We assume that for each sense there exists a synonym set of words expressing that sense. Let $S_i = \{\sigma_1^{(i)}, \sigma_2^{(i)}, \ldots, \sigma_{s_i}^{(i)}\}$ be the synonym set for the i'th sense.

In the general case the mirroring method is applied in four steps:

1. First image $t(\sigma) = \{\tau_1, \tau_2, \ldots, \tau_r\} =: T_0$.
2. First inverse image $t^{-1}(T_0) = \{\sigma, \sigma_1, \sigma_2, \ldots\}$. Put $\Sigma_0 = t^{-1}(T_0) \setminus \sigma$.
3. Second image $t(\Sigma_0)$. Put $T = t(\Sigma_0) \backslash T_0$.
4. Second inverse image $t^{-1}(T)$. Put $\Sigma = t^{-1}(T) \cap \Sigma_0$.

The first image $t(\sigma)$ generates a set of words in the target language. Each one of these words should have at least one sense in common with σ. However, in case one of the languages makes finer distinctions than the other in this area of the lexicon, we may get a sub-sense, or a super-sense, instead of Ξ_i. A typical case is given by family relationships, where Swedish has some lexical items with a more specific meaning than the common English counterparts, e.g., **grandmother** $\leftrightarrow$ **mormor, farmor** (mother's mother, father's mother).

It may also happen that some words in the first image have other senses than those covered by σ, i.e., a case of contrastive ambiguity. They may be homonyms or they may be polysemic in a manner that σ is not. An example is **pond** $\rightarrow$ **damm**, where the Swedish word may mean 'dust' as well as 'pond'.

As a result, the sense space of the first image need not be isomorphic with the sense space of σ. In the case of Swedish words having more specific meanings, one sense has been divided into two or more specific senses; in the case of contrastive ambiguities, we will introduce unrelated senses that have no counterpart in the original sense space.

The first inverse image Σ_0 will introduce a number of source words apart from σ. Considering all senses of all these words, we might have an enlargement of senses due to homonymy, polysemy and sense narrowings/widenings. Thus, the sense space spanned by all of the words surely includes the original sense space, but may be both enlarged, sub-specified, or generalized. It is unlikely, however, that the space is both specified and generalized, since a specification in one direction, should result in a generalization in the inverse direction, and vice versa. But we may have introduced senses semantically unrelated to the ones we started with.

At this stage, we have little knowledge of the structure of the sense space. However, we may see clusters of words σ_j that are reached from the seed word via different paths $\sigma \rightarrow \tau_i \rightarrow \sigma_j$.

When computing the second image we leave out the original word, σ, but map all the new source words obtained by the first inverse image. As an effect we may see new target words that we didn't see from the first image. Also, we may see new paths to the target words of the first image.

As compared with the target sense space of the first image, the sense space of the second image may contain sub-senses and super-senses to the senses of the first image. And, just as for the source language, we may have introduced senses that are unrelated to those of the first image.

For each word of the first image, we can see clusters via paths to the source side that share images.

In the second inverse image we exclude the target words from the first image and we are only interested in new paths to words of the first inverse image. As this is our last mirroring, any source word that has not be seen previously has uncertain relations to the target side and is seen as unhelpful.

We can now look at clusters formed by the words of the first inverse image, including or excluding the seed word. A graph can be obtained in the following way:

- each word $\sigma_j \in \Sigma$ is represented as a node j of the graph,
- if there is a path $\sigma_i \rightarrow \tau_j \rightarrow \sigma_k$ we register an edge between the nodes i and k,
- the weight of edge (i, k) is set equal to the number of paths linking nodes i and k,
- loops (i, i) are included, for reasons that will be discussed later.

It may happen that some node i has no edges with the other nodes (except self-loops). This may be taken as evidence that it represents a separate sense, which may or may not be part of the original sense space of σ, as it may be the result of target word homonymy or polysemy. Example: **pond** $\rightarrow$ **damm** $\rightarrow$ {**dust, pond, ...** } where **dust** has no semantic relation to **pond** and possibly not to any other word as well generated from **pond**.

Generally speaking, if the edge weight is high for (i, k), this is strong evidence that they belong to the same semantic cluster and share at least one sense, or super-sense. But other words may belong here too. Rather than defining clusters positively, we can look at weak relations between sets of nodes, and, if we find a weak relation, draw the tentative conclusion that there is no interesting cluster that include two nodes from the different sub-sets. This requires, however, that we can find a suitable definition of *weak relation.* To do this we turn to graph theory, and spectral graph partitioning, in particular.

2.3 Translation Matrix and Mirror Graph

We will now give a matrix/graph oriented description of the mirroring procedure. The lexicon in the preceding section defines a sparse matrix B. In our experiments we have worked with an English-Swedish lexicon consisting of 14850 English adjectives, and their corresponding 20714 Swedish translations. For concreteness we will refer to this lexicon in the sequel; obviously the method can be applied to an arbitrary lexicon between two languages.

The words in the two languages are $\sigma_1, \sigma_2, \ldots, \sigma_m$, and $\tau_1, \tau_2, \ldots, \tau_n$. We first illustrate the mirroring procedure in an example.

Example 2 The mirroring procedure is illustrated in Figure 2. For simplicity of notation we here assume that the words occurring in the translation are those with indices $1, 2, \ldots$. We do not want σ_1 to occur in the first inverse t-image,

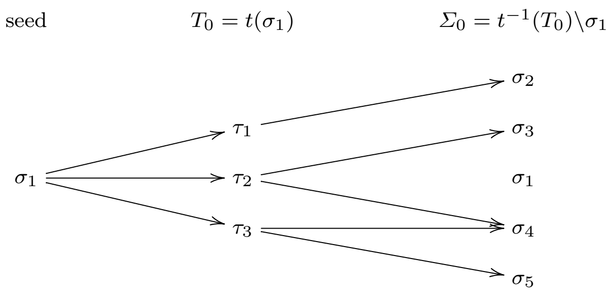


Fig. 2 First mirroring operation.

because in the second mirroring σ_1 would only reproduce the first mirroring.

Next we are going to perform a second mirroring, which will make it possible to group the words in the inverse t-image, $\Sigma_0 = \{\sigma_2, \sigma_3, \sigma_4, \sigma_5\}$, which, in turn, will reveal the different meanings of the seed word σ_1. In the second mirroring, see Figure 3, we omit the translations via τ_1, τ_2, τ_3, as those will only reproduce the first mirroring.

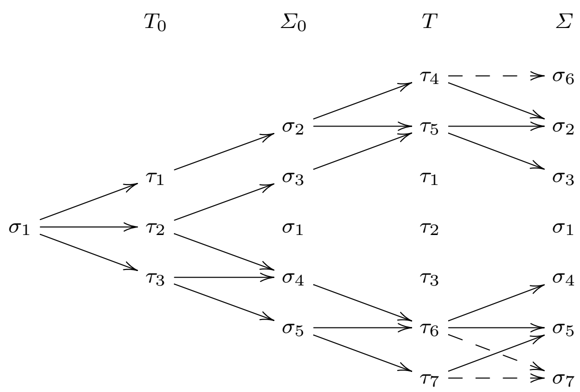


Fig. 3 First and second mirroring.

The second mirroring defines a mirror graph involving the words $\sigma_2, \sigma_3, \sigma_4, \sigma_5$. It is seen that the graph in the example is disconnected, which suggests that there are two different senses. As noted in the previous subsection the source words σ_6 and σ_7 have uncertain relations to the words in T and are therefore excluded from the final image Σ.

The *translation matrix* $B \in \mathbb{R}^{m \times n}$ is defined

$$b_{ij} = \begin{cases} 1, & \text{if } \sigma_i \text{ translates to } \tau_j, \\ 0, & \text{otherwise.} \end{cases}$$

The weighted adjacency matrix of the mirror graph can be computed from the translation matrix as follows. The translations of a *seed word*, σ, are found by the operation $B^{\mathsf{T}} e_\sigma$, where e_σ is a unit vector with zeros everywhere except in the position of σ. The nonzero components of $B^{\mathsf{T}} e_\sigma$ will indicate the Swedish words that are translations of σ, the t-image of σ_σ.

Translation back to English of all the Swedish words is then obtained by the multiplication $BB^{\mathsf{T}} e_\sigma$. As we do not want σ to occur in the first inverse t-image (see above), we remove it by multiplication by $I - e_\sigma e_\sigma^{\mathsf{T}}$,

$$f = (I - e_\sigma e_\sigma^{\mathsf{T}}) BB^{\mathsf{T}} e_\sigma.$$

Assume that the nonzero elements of f are in positions $i_1, i_2, \ldots, i_s$, which means that the first inverse t-image is $\Sigma_0 = \{\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_s}\}$. Define the first inverse t- image matrix

$$F_s = \begin{pmatrix} e_{i_1} & e_{i_2} & \cdots & e_{i_s} \end{pmatrix}.$$

To avoid using translations via the first t-image (cf. the example above), we use a modified translation matrix $B_{(2)}$, where the columns corresponding to (the Swedish) words in the first t-image $\mathbb{T}_0$ have been replaced by zeros, as well as the column corresponding to σ. The second inverse t-image is then obtained from the multiplication $B_{(2)} B_{(2)}^{\mathsf{T}} F_s$. Finally, we are not interested in the English words that occur for the first time in the second inverse t-image, so we remove those by multiplication by F_s^{T}. Thus the adjacency matrix for the graph, where the nodes represent words in the first inverse t-image is given by the nonzero rows and columns of

$$A = F_s^{\mathsf{T}} B_{(2)} B_{(2)}^{\mathsf{T}} F_s. \tag{1}$$

The adjacency matrix is *weighted*: a mirroring of σ_k via two different τ_p and τ_q back into σ_l indicates a stronger relation between σ_k and σ_l than a mirroring via only one Swedish word. The multiplication (1) gives the weight 2 to the corresponding edge.

Example 3 The translation matrix of Example 2 can be written (if we exclude all words that do not occur in the present translation)

$$B = \begin{pmatrix} 1&1&1&0&0&0&0 \\ 1&0&0&1&1&0&0 \\ 0&1&0&0&1&0&0 \\ 0&1&1&0&0&1&0 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&1 \end{pmatrix}.$$

The weighted adjacency matrix becomes

$$A = \begin{pmatrix} 2&1&0&0 \\ 1&1&0&0 \\ 0&0&1&1 \\ 0&0&1&2 \end{pmatrix},$$

where the rows and columns correspond to the words $\sigma_2, \sigma_3, \sigma_4, \sigma_5$. Note that the matrix is reducible, which is equivalent to the graph being disconnected and here having two components.

The elements on the diagonal are the number of self-loops. We will let the weighted adjacency matrix have self-loops, based on the following heuristics. An English word with many self-loops but few edges corresponds to a wider meaning in Swedish than in English (cf. assumption 2 on page 3), and that should be given a weight when we try to cluster the English words according to different senses. However, we will demonstrate later that the use of node weight does not influence the ordering of the nodes induced by the spectral partitioning method, it only influences our measure of sparseness of the graph.

Example 4 Consider the graph in Figure 4. A single English word σ_1 that has

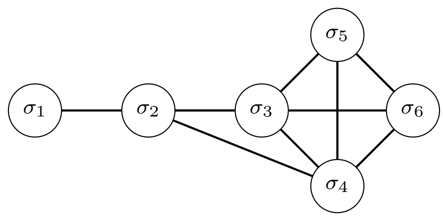


Fig. 4 Example 4. Partitioning of a graph, where σ_1 is assumed to have several self-loops.

translations back to itself via many Swedish words (many self-loops) is more likely to represent a different meaning from $\sigma_2, \ldots, \sigma_6$ than if it has only one self-loop, say. In this situation, when the weight of the node σ_1 is high, the graph should be partitioned between σ_1 and σ_2.

We will come back to this example when we have defined a measure of the "well-connectedness" of a partitioning of a graph.

3 Spectral Graph Partitioning

We first give some basic definitions and state properties of graphs related to spectral partitioning. For extensive presentations of the theory, we refer e. g. to [4, 23].

Let $A \in \mathbb{R}^{n \times n}$ be the weighted adjacency matrix of an undirected, connected graph $\mathcal{A}$ on n nodes. The *degree* d_i of node i is the number of edges that emanate from the node, including self-loops (counted once). The degree vector $d = \begin{pmatrix} d_1 & d_2 & \cdots & d_n \end{pmatrix}^{\mathsf{T}}$ satisfies

$$d = Ae,$$

where $e \in \mathbb{R}^n$ is the vector of all ones. Let D be the diagonal matrix made up from d, i.e. $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$. The *Laplacian matrix* L of the graph is defined as

$$L = D - A.$$

Obviously e is an eigenvector of L with eigenvalue 0. Since $\mathcal{A}$ is connected (equivalently, A is irreducible) this is the only zero eigenvalue, all the rest are positive. Denote the eigenvalues of L by

$$0 = \lambda_0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-1},$$

and the corresponding eigenvectors are u_i, $i = 0, 1, \dots, n-1$.

In this paper we will mainly use the *normalized Laplacian matrix* L_n,

$$L_n = I - D^{-1/2} A D^{-1/2}.$$

Here $D^{1/2}e$ is an eigenvector of L_n with eigenvalue 0. The eigenvalues of L_n are

$$0 = \nu_0 < \nu_1 \le \nu_2 \le \cdots \le \nu_{n-1},$$

and the eigenvectors are v_i, $i = 0, 1, \dots, n-1$.

The relations between the properties of the graph and those of the eigenvalue problems for the two Laplacian matrices are similar.

The first nonzero eigenvalue[3] ν_1 (or λ_1) is called the *Fiedler eigenvalue*; we will refer to the corresponding eigenvector v_1 as the *Fiedler vector*[4]. The connectivity of the graph is strongly related to the value of ν_1: loosely speaking, the closer the value of ν_1 to zero, the closer the graph is to being disconnected. For more precise statements of this property, see [4] and below.

We want to partition a connected graph in two, by removing some edges. Thus we want to find the part of the graph, where it can be split up in two subgraphs by breaking as few edges as possible. Let $\mathcal{S}$ be a subgraph[5] of $\mathcal{A}$, and $\bar{\mathcal{S}}$ its complement. The *volume* of $\mathcal{S}$ is defined $\mathrm{vol}(\mathcal{S}) = \sum_{i \in \mathcal{S}} d_i$; this is the sum of the degrees of the nodes in $\mathcal{S}$. Then define the *edge boundary* $\delta\mathcal{S}$ of $\mathcal{S}$ to consist of all edges with exactly one endpoint in $\mathcal{S}$. We first formulate, tentatively, the following partitioning problem:

> Find a subgraph $\mathcal{S}$ such that its edge boundary $\delta\mathcal{S}$ contains as few edges (counted with weights) as possible.

However, that problem may be too simplistic: $\mathcal{S}$ consisting of any node with only one edge of weight one would be a solution to that problem. To avoid that situation we require that the two subgraphs are not too small [4]:

> **Partitioning problem:** For a fixed number m, find a subgraph $\mathcal{S}$ with $m \le \mathrm{vol}(\mathcal{S}) \le \mathrm{vol}(\bar{\mathcal{S}})$ such that the edge boundary $\delta\mathcal{S}$ contains as few edges as possible.

[3] It need not be unique: there are graphs for which $\nu_1 = \nu_2 = \cdots = \nu_k$, for some k. In our application usually this poses no problems.

[4] The terminology is not always consistent here: sometimes in the literature $D^{-1/2}v_1$ is called the Fiedler vector.

[5] With some abuse of notation, we will refer to a subset of the nodes as a subgraph, whose edges are those between the nodes in the subset.

To quantify the relation between the solution of the best partitioning problem we define the *conductance* of a partitioning. Assume that we have partitioned $\mathcal{A}$ into two subgraphs $\mathcal{S}$ and $\bar{\mathcal{S}}$ by removing a set of edges. The conductance $\psi(\mathcal{S})$ is defined

$$\psi(\mathcal{S}) = \frac{|E(\mathcal{S},\bar{\mathcal{S}})|}{\min(\mathrm{vol}(\mathcal{S}),\mathrm{vol}(\bar{\mathcal{S}}))}, \tag{2}$$

where $|E(\mathcal{S},\bar{\mathcal{S}})|$ is the total weight of the edges that are removed. The conductance of the graph[6] $\mathcal{A}$ is defined

$$\psi(\mathcal{A}) = \min_{\mathcal{S}} \psi(\mathcal{S}),$$

where the minimum is taken over all possible partitionings. The *Cheeger inequalities* [4, Section 2] give a relation between the Fiedler value and the conductance of the graph:

$$2\psi(\mathcal{A}) \geq \nu_1 \geq \frac{(\psi(\mathcal{A}))^2}{2}.$$

Finding the optimal partition, i.e. the one that minimizes the conductance over all possible partitionings, is NP-hard [1, 23]. However, *spectral partitioning* provides a heuristic.

> **Spectral partitioning:** Given the Fiedler vector v_1 (or u_1), reorder its elements in ascending order. This defines a permutation of the integers $\{1,2,\ldots,n\}$. This induces a reordering of the nodes of the graph: Apply the permutation to the nodes of the graph, and modify the set of edges accordingly. For each partitioning $\mathcal{S}_\eta = \{1,\ldots,\eta\}$, $\overline{\mathcal{S}_\eta} = \{\eta+1,\ldots,n\}$, $\eta = 1,2,\ldots,n$, of the reordered graph, compute the corresponding conductance. Choose the partitioning with the smallest conductance.

The heuristic is based on the following well-known Rayleigh quotient property of the Fiedler vector.

Proposition 1 *The Fiedler vector u_1 is the solution of the minimization problem,*

$$\min_{x \perp e} \frac{x^{\mathsf{T}} L x}{x^{\mathsf{T}} x} = \frac{1}{2} \min_{x \perp e} \frac{\sum_{i,j} a_{ij}(x_i - x_j)^2}{x^{\mathsf{T}} x} =: \mu_1, \tag{3}$$

and the Fiedler value μ_1 is the minimum.

In the normalized case the Fiedler vector is $v_1 = D^{1/2}\hat{y}$, where $\hat{y}$ is the solution of

$$\min_{y \perp D^{1/2} e} \frac{y^{\mathsf{T}} L_n y}{y^{\mathsf{T}} y} = \frac{1}{2} \min_{y \perp D^{1/2} e} \frac{\sum_{i,j} a_{ij}\left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}}\right)^2}{y^{\mathsf{T}} y} =: \nu_1, \tag{4}$$

and the Fiedler value is the minimum.

The proof can be found, e.g., in [3, 4, 23]. We only give a sketch here.

[6] Also called the *Cheeger constant* [4].

Proof From the identity $L = D - A$ we see that $L = \sum L_{ij}$, where the edge between nodes i and j is represented by the matrix

$$(L_{ij})_{pq} = a_{ij} \begin{cases} 1 & \text{if} \quad p = q = i, \quad \text{or} \quad p = q = j, \\ -1 & \text{if} \quad p = i,\ q = j, \quad \text{or} \quad p = j,\ q = i, \\ 0 & \text{otherwise.} \end{cases}$$

Then, since $x^{\mathsf{T}} L_{ij} x = a_{ij}(x_i - x_j)^2$, the relation (3) follows from the Rayleigh quotient characterization of the eigenvalues and eigenvectors of a symmetric matrix.

The proof of the second part is analogous, using the identity $L_n = D^{-1/2} L D^{-1/2}$.

Denote the components of the Fiedler vector $u_1 = (\xi_1, \xi_2, \ldots, \xi_n)^T$, and let $\mathbb{U}$ be the mapping of the nodes onto the real line, $\mathbb{U} : i \to \xi_i,\ i = 1, 2, \ldots, n$. From the minimization property (3) we see that nearby nodes ($a_{ij} \neq 0$), are mapped closely on the real line. In addition, edges (i, j) with high weight, force the image points ξ_i and ξ_j closer to each other. Now, if we reorder the elements of the Fiedler vector in ascending order, and induce the same reordering on the nodes, then nearby nodes i_1 and i_2 ($a_{i_1 i_2} \neq 0$) in the new ordering, will have $|i_1 - i_2|$ small. Therefore, if the graph has two well connected subgraphs and a weak coupling between them, it is likely that the weak coupling will be found by testing (computing the conductance of) partitionings ($\mathcal{S}_\eta$, $\bar{\mathcal{S}_\eta}$) as defined in the heuristic.

A similar interpretation can be done for the heuristic based on the normalized Laplacian. An examination of a path graph (i.e. one with a tridiagonal adjacency matrix), where one edge has much larger weight than the others, shows that in the normalized case the edge with large weight is deemphasized as compared to the unnormalized case.

In [23, Section 6] an alternative derivation of the heuristic is given, based on the notion of random walks on graphs, where the matrix of transition probabilities is $D^{-1}A$:

> "... spectral clustering can be interpreted as trying to find a partition of the graph such that the random walk stays long within the same cluster and seldom jumps between clusters."

In Section 2.3 we argued that self-loops should be allowed. That leads to elements on the diagonal of the adjacency matrix. Let $A = D_0 + A_0$, where D_0 is diagonal with the weights of the self-loops, and A_0 has a zero diagonal. Then, since $D = \mathrm{diag}(Ae) = D_0 + \mathrm{diag}(A_0 e)$, it is easily seen that $D - A = \mathrm{diag}(A_0 e) - A_0$. Therefore, the eigenvalues or eigenvectors of the Laplacian are independent of the self-loops. However, from the definition (2) we see that the self-loops influence the conductance of the partitioning.

Example 5 Define two partitionings of the graph in Example 4,

$$\mathcal{S}_1 = \{\sigma_1\}, \qquad \mathcal{S}_2 = \{\sigma_1, \sigma_2\},$$

and let the number of self-loops in nodes 1 and 2 be s_1 and s_2, respectively, with $s_i \geq 0$, $i = 1, 2$. Assume that all edges have weight 1. If we assume that for the partitionings $\mathrm{vol}(\mathcal{S}_i) \leq \mathrm{vol}(\bar{\mathcal{S}_i})$, $i = 1, 2$, then the conductances are

$$\psi(\mathcal{S}_1) = \frac{1}{s_1 + 1}, \qquad \psi(\mathcal{S}_2) = \frac{2}{s_1 + s_2 + 4}.$$

It follows that

$$\psi(\mathcal{S}_1) < \psi(\mathcal{S}_2) \quad \Longleftrightarrow \quad 2 + s_2 < s_1.$$

Thus, if we do not include the self-loops, i.e. if we put $s_1 = s_2 = 0$, then, based on conductance, we will not be able to partition the graph between nodes 1 and 2.

3.1 Partitioning Mirror Graphs

In Algorithm 3.1 we give a pseudocode for the partitioning of a connected graph. It is assumed that the function MinConductance computes the conductance of partitionings along the ordering of the graph.

Algorithm 3.1 Spectral partitioning of a connected graph $\mathcal{S}^{(j)}$

function $[\mathcal{S}^{(a)}, \mathcal{S}^{(b)}] = \text{Partition}(\mathcal{S}^{(j)})$
  $v = \text{FiedlerVector}(\mathcal{S}^{(j)})$
  $[v_{\text{sorted}}, iv] = \text{Sort(v)}$
  $\mathcal{S}^{(j)} = \mathcal{S}^{(j)}(iv)$ {Reorder the nodes}
  $p = \text{MinConductance}(\mathcal{S}^{(j)})$ {Compute partitioning $(\mathcal{S}_p, \bar{\mathcal{S}_p})$ with minimum conductance}
  $\mathcal{S}^{(a)} = \mathcal{S}_p$; $\mathcal{S}^{(b)} = \bar{\mathcal{S}_p}$

In the overall algorithm the set of graphs $\mathcal{S}$ is initialized as a connected graph. At the stage when it has been partitioned into k disconnected graphs, $\mathcal{S} = \{\mathcal{S}^{(j)}\}_{j=1}^{k}$, the one with the smallest Fiedler value is further partitioned. The pseudocode is given in Algorithm 3.1. This algorithm is related to one of the normalized Laplacian algorithms in [16] ($k = 1$), but it differs in the way the cut is made: we are computing explicitly the conductance along the new node ordering to find the cut with minimum conductance.

Algorithm 3.2 Partitioning of a graph $\mathcal{S}$

$k = 1$
while $k < k_{\max}$ **do**
  $\mathcal{S}^{(\min)}$=MinFiedlerValue($\mathcal{S}$)
  $\mathcal{S} = \mathcal{S} \backslash \mathcal{S}^{(\min)}$
  $[\mathcal{S}^{(a)}, \mathcal{S}^{(b)}] = \text{Partition}(\mathcal{S}^{(\min)})$
  $\mathcal{S} = \mathcal{S} \cup \mathcal{S}^{(a)} \cup \mathcal{S}^{(b)}$
  $k = k + 1$

When the original graph has been partitioned into $k_{\max}$ disconnected graphs, where $k_{\max}$ is a user-supplied parameter (perhaps given interactively), this final result can be considered as a weighted supergraph, where each node represents a cluster of words. The edge weights are the sums of the weights of the corresponding edges in the original graph.

The stopping criterion in Algorithm 3.1 based on $k_{\max}$ is suitable in the case when the partioning is performed in a supervised way. In an automatic setting one can take the Fiedler values as stopping criterion: continue partitioning as long as any $\mathcal{S}^{(j)}$ has Fiedler value below a specified threshold, cf. [16]. Note that 1 is an upper bound for the largest Fiedler value for a normalized Laplacian; therefore an absolute threshold can be used.

4 Experiments

We have conducted a number of experiments with the lexicon of Swedish and English adjectives. We followed the mirroring procedure of Section 2.3, and performed spectral partitioning of the graph according to Section 3.1 using the normalized Laplacian. In most examples very similar results were obtained with the unnormalized Laplacian (see, however, the word **destitute** below).

Small graphs

Some words give rise only to one-word disconnected subgraphs or very small graphs that cannot be partitioned in any reasonable way. For instance, mirroring of **American** gives two disconnected graphs with the words *Yankee* and *stateside*. Similarly, **British** gives the word *Britannic* and the small graph in Figure 5, indicating that the three words are synonyms.

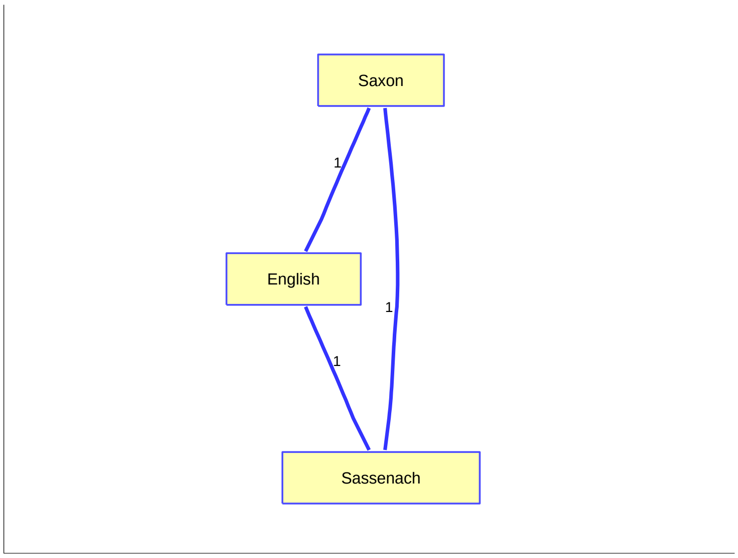


Fig. 5 Graph for **British**.

Destitute

The original graph (with 14 nodes) for the second mirroring of **destitute** is illustrated in Figure 6. Inspection of the graph[7] suggests that there are three groups

[7] The layout of the nodes in the figure is based on the Fiedler vector.

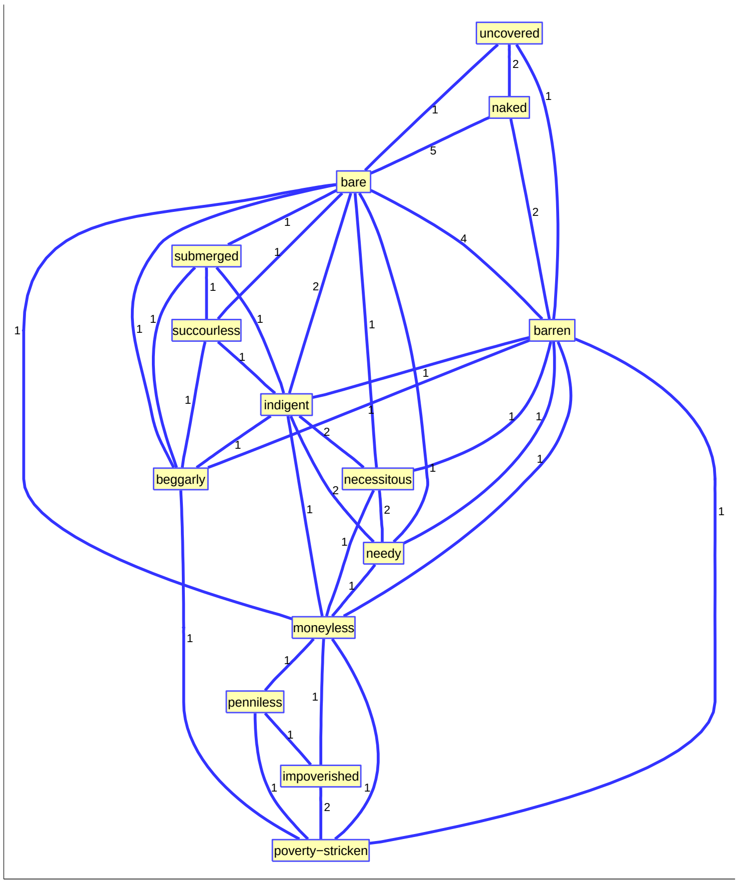


Fig. 6 Original graph of **destitute**.

of words. The graph was partitioned twice; the Fiedler values for the cuts were 0.16, 0.15. The words of the three clusters for **destitute** are given in Table 1.

bare	poverty-stricken	indigent	beggarly
barren	moneyless	needy	submerged
naked	impoverished	necessitous	
uncovered	penniless	succourless	

Table 1 Three clusters for **destitute**.

In order to investigate synonymy between the original seed word and the clusters, we performed the mirroring procedure with the words **bare**, **poverty-stricken**, and **indigent**.

The graph for **bare** had 101 nodes. After partitioning into 8 clusters, there was one cluster with the words *bald, naked, nude, undraped, unclad, unclothed, featherless, sky-clad, bareassed, uncovered*, and one with the words *Mickey Mouse,*

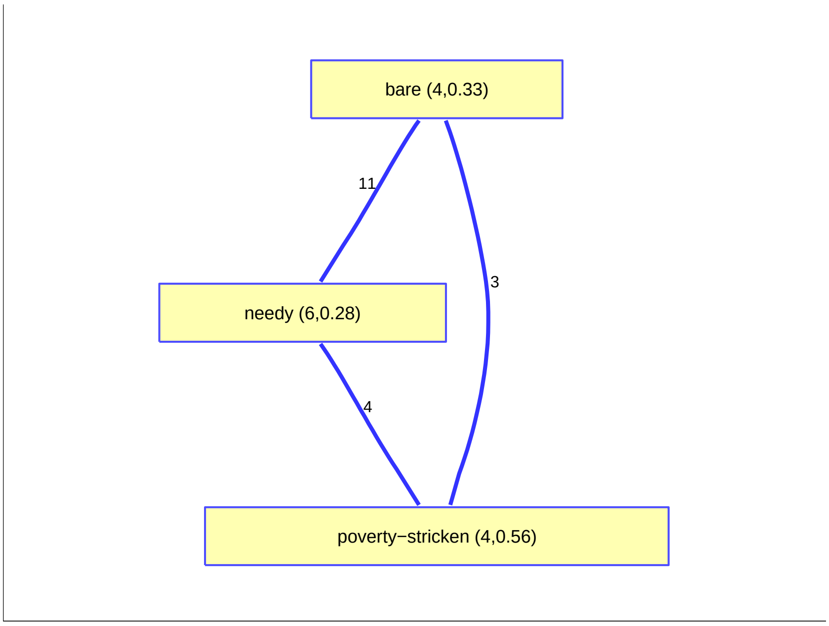


Fig. 7 Clustering for **destitute**. The first number in the node label is the number of words in the group, the second is the Fiedler value of the cluster (for further partitioning). The edge labels show the total weight of the edges between the subgraphs.

needy, penurious, necessitous, indigent, moneyless, downscale, destitute, submerged, beggarly, succourless (cf. Tables 1 and 3).

The adjective **poverty-stricken** generated one graph with 50 nodes. Partitioning it three times (Fiedler values 0.13, 0.19, and 0.56) we obtained the clusters in Table 2.

destitute	wretched	miserable	scanty	mean
moneyless	vile	mangy	bald	penurious
impoverished	paltry	woeful	unsubstantial	barren
penniless	low-down	abject	insubstantial	arid
	ratty	sorry	meager	skimpy
	sordid	distressful	tenuous	beggarly
	forlorn	rotten	jejune	comfortless
	unhappy	deplorable	parsimonious	frugal
	back	verminous	lean	homely
	dusty	misbegotten	hungry	coarse
	squalid	godforsaken	poky	poor
	miscreated			threadbare

Table 2 Four clusters for **poverty-stricken**. The clusters are given in the order of appearance in the partitioning procedure.

We partitioned the graph for **indigent** into four clusters (Fiedler values 0.26, 0.19, and 0.26). The clustering is given in Table 3.

needy necessitous deprived moneyless	mean poor barren bare beggarly	submerged destitute succourless	Mickey Mouse penurious downscale

Table 3 Three clusters for **indigent**.

Notice the overlap between the clusters two for **destitute** (Table 1) and one for **poverty-stricken** (Table 2), which indicates synonymy.

Yellow

The original graph had two components, one consisting of the words *xanthous, jaundiced, amber*, and the other one with 21 nodes. After one partitioning (with Fiedler value 0.11) of the large component we have the clustering given in Table 4.

jaundiced amber xanthous	base mean-spirited faint-hearted cowardly caitiff pigeon-hearted	weak-hearted white-livered poor-spirited chicken recreant craven	skulking yellow-bellied gutless chicken-livered chicken-hearted lily-livered	windy funky trembly

Table 4 Clustering of **yellow**.

Blue

Two mirrorings of **blue** gave two disconnected graph components, one with 146 nodes and the other with only one (*porny*). As a graph of 146 nodes becomes illegible when displayed, we give in Figure 8 instead a spy plot of the adjacency matrix (i.e. a plot where each non-zero matrix element is shown as a dot).

We then partitioned the graph with Fiedler values between 0.048 and 0.150 . The results are given in Table 5 and Figure 9.

Green

Two mirrorings of **green** gave two disconnected graph components, one with 144 nodes and the other with only one (*unfired*). In Figure 10 a spy plot of the adjacency matrix is given.

We then partitioned the graph with Fiedler values between 0.0079 and 0.22 . The results are given in Table 6 and Figure 11.

4.0.1 Raw

The word is contained in the last cluster for **green** in Table 6. Two mirrorings of that word gave two disconnected graph components, one with 160 nodes and the other with three. In Figure 12 a spy plot of the adjacency matrix is given.

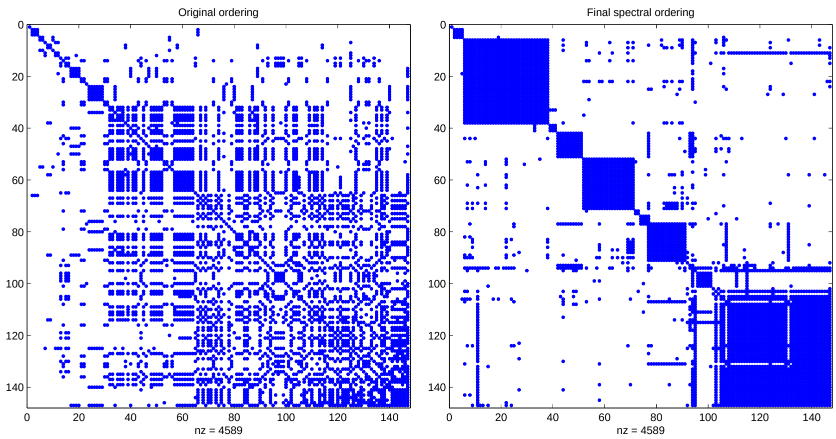


Fig. 8 Left: Spy plot of the 146 node unordered and unpartitioned largest graph component for **blue**. Right: The final ordering of the graph after partitioning into nine clusters.

porny	conservative	livid	downbeat	depressed	dejected
	old-line	azure	doomy	drooping	despondent
	Tory		pissed-off	crestfallen	discouraged
	standpat			dispirited	chapfallen
				downcast	flat

mean	low	ignoble	humble	sordid	gentle
menial	base	ornery	short	bass	stumpy
low-pitched	neap	low-slung	splenetic		

foul	free	nasty	greasy	broad	filthy
obscene	coarse	smutty	foul-mouthed	lewd	scabrous
raunchy	immodest	naughty	improper	indecent	ribald
prurient	bawdy				

rude	hard	close	tight	severe	harsh
grim	rigorous	strict	stern	inclement	gruelling
intense	stringent	rigid	unrelaxed	ironclad	hard-handed
austere	dour	scrupulous	astringent	unrelenting	chaste
strait	exacting	iron	iron-bound	unsparing	censorious
hardshell	tight-laced	blue-nosed	puritan	Puritan	puritanical

heavy	dull	sad	dark	gloomy	murky
grave	dismal	cloudy	sullen	woeful	dreary
dusky	overcast	melancholy	tenebrous	wan	saturnine
doleful	sunless	mournful	low-browed	Stygian	sombre
blackbrowed	deep	obscure	heavy-hearted	bleak	leaden
funereal	mirthless	grey	woebegone	cheerless	morose
depressing	disconsolate	gaunt	lonely	glum	lugubrious
beetle-browed	dyspeptic	Lenten	lenten	rueful	chill
depressive	offputting	discouraging	dismaying	disheartening	atrabilious
melancholic					

Table 5 Clustering of **blue**.

We then partitioned the graph with Fiedler values between 0.053 and 0.086 . The results are given in Table 7.

In Figure 13 we show the clustering into 6 clusters. We see that the Fiedler value for the cluster *rough* is quite low. Therefore we performed three more steps in the procedure, which led to the addition, in order, of the cluster *mean, foul,*

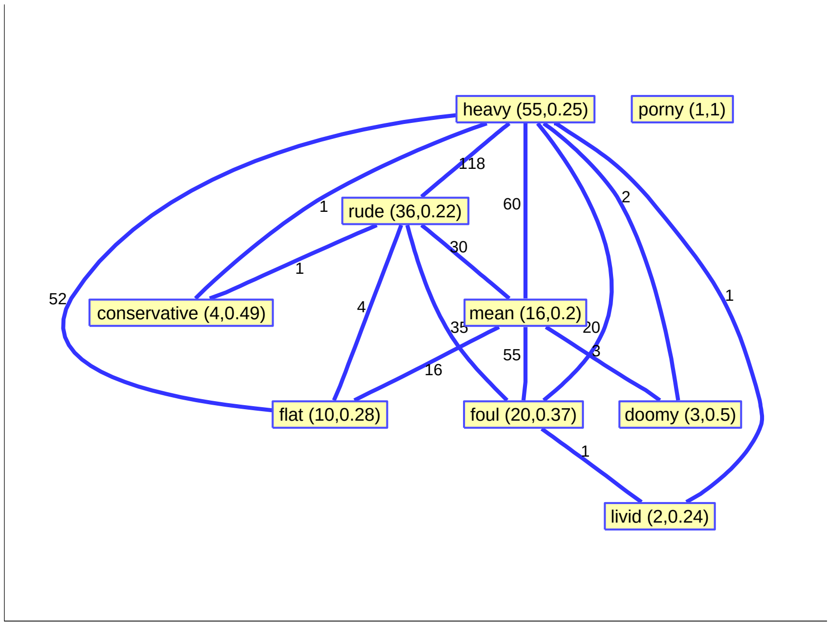


Fig. 9 Clustering of **blue**.

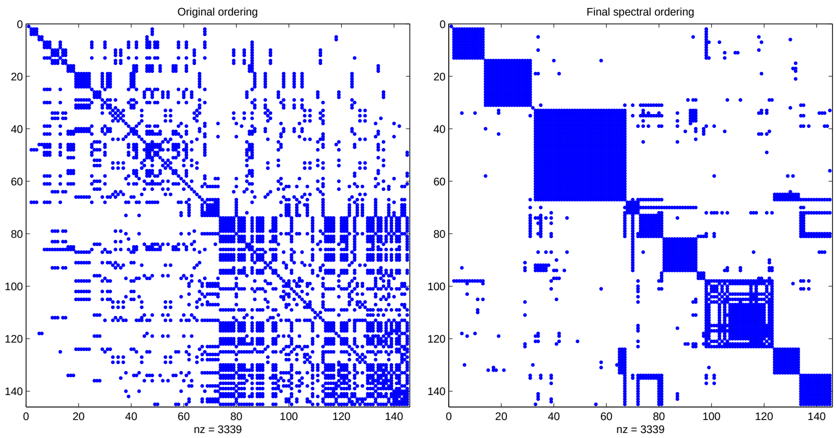


Fig. 10 Left: Spy plot of the 144 node unordered and unpartitioned largest disconnected graph component for **green**. Right: The final ordering of the graph after partitioning into eleven clusters.

low, vile, nasty, sordid, mangy, shabby, ignoble, base, shoddy, low-down, homely, paltry, dirty, scurvy, menial, humble, ornery, ratty, illiberal, infamous, tacky, sodden, caddish, tatty, plebeian, common, cheap, scummy, loud, ugly, snide, unhandsome, greasy, shivery, with partitioning Fiedler value 0.31, *green, fresh, half-baked, young, unfledged, verdant, callow, unseasoned, sucking, tinhorn, inexpert, unformed, unpractised, immature, untutored, untrained, uncultivated, undeveloped, undergrown, stunted, inchoate, unbred*, with partitioning Fiedler value 0.18, and finally *rusty*.

unfired	vegetable	lush	half-baked	inexpert	unexperienced
		virescent	untried	callow	unversed
		verdurous	unpractised	unskilled	freshwater
			sucking	unlearned	inexperienced

fit	chipper	simple	unworldly	wet	moist
buoyant	thoroughbred	unsophisticated	simple-minded	waterish	wettish
elastic	athletic	artless	soft-headed	vapoury	damp
racy	hale	innocent	dewy-eyed	soggy	springy
supple		untutored	simplistic	drizzly	
resilient		ingenuous	cabbage-looking	oozy	
vigorous		naive		humid	

juvenile	sweet	brisk	sickly	rough	feral
young	smart	whole	wishy-washy	rude	beastly
unfledged	keen	well-conditioned	washy	crude	unseasoned
verdant	fresh	untainted	bloodless	coarse	brute
unformed	clear	recent	wan	vulgar	swinish
untimely	snappy	tinhorn	pale	foul	greige
immature	sound	upstart	anodyne	raw	bestial
bread-and-	gradely	new-laid	watered-down	uncouth	brutish
butter	warm		whey-faced	gross	blackguardly
youthful	new		pasty-faced	broad	rheumy
adolescent	caller		vealy	tough	butcherly
juvenescent	original		white	brutal	dank
unripe	breezy		innocuous	indelicate	Gothic
underripe	well		mealy	barbarious	uncooked
sophomoric	rattling		untanned	scurrilous	ruffianly
	unvitiated		whitish	bearish	unchastened
	wholesome		pale-faced	foul-mouthed	clammy
	healthy		pallid	churlish	

Table 6 Clustering of **green**.

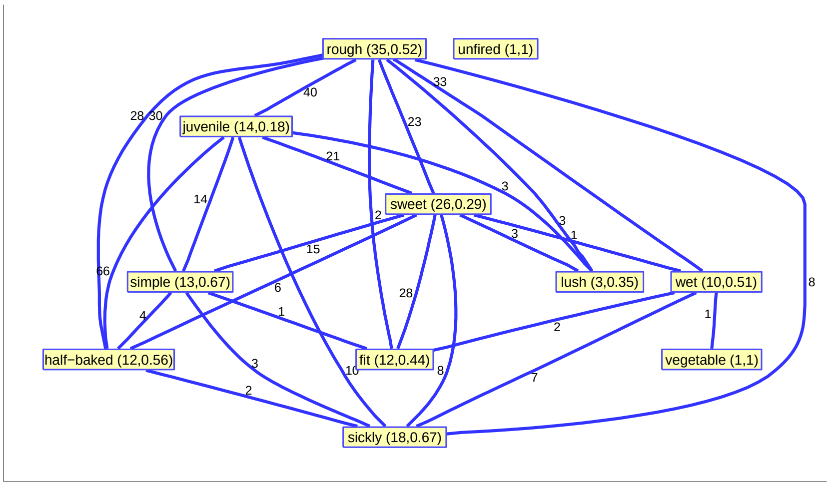


Fig. 11 Clustering of **green**.

5 Related Work

The notion of synonym sets (synsets) was made primary in the construction of the English WordNet where it is used to represent a sense (Miller, 1995; [18,

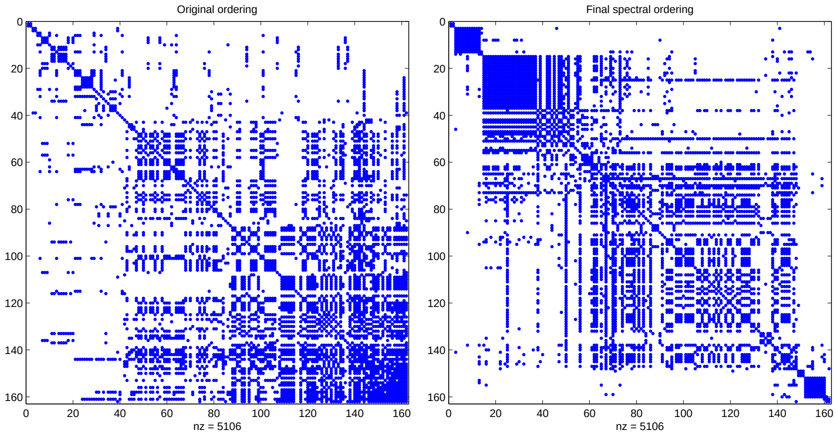


Fig. 12 Left: Spy plot of the 159 node unordered and unpartitioned largest disconnected graph component for **raw**. Right: The final ordering of the graph after partitioning into 6 clusters.

uncured	unprofessional	gentle	rough	bleak	rare
unhealed	unworkmanlike	tender	heavy	parky	sanguinary
	amateurish	sensitive	rugged		bloody
		sore	crude		cruel
		fond	vulgar		gory
		loving	coarse		bloodstained
		affectionate	hard		scathing
		yearning	foul		underdone
		amatory	gross		blood-and-guts
		fatherly	. . .		
		caressing	. . .		
		thin-skinned	unbred		

Table 7 Clustering of **raw**. The largest cluster has 134 words.

10]. Bilingual dictionaries have been used for the construction and population of WordNets in other languages. They are applied for the generation of candidate words and senses in the target language. These candidates are then filtered on the basis of some simple criteria. For instance, Chugur et al. [2] assigned a Spanish word to a WordNet sense if it appeared among the translations of at least two words from the English synset for that sense. Automated procedures of this kind work best for the cases where single-sense single word synsets map to single words in the target language and thus no synonym relations are produced.

Priss and Old [20] proposed to base the construction of senses and semantic relations from semantic mirrors on Formal Concept Analysis (FCA [12]). Direct comparisons of their approach with ours are difficult to make, however, as both aims and means are different. We are primarily interested in bringing information to light that is available, but not explicit, in a resource such as a bilingual dictionary, while Priss and Old are more concerned with the construction of "smaller size bilingual resources, such as ontologies and classification systems". We have in common, however, the belief that the generated data is often not the desired

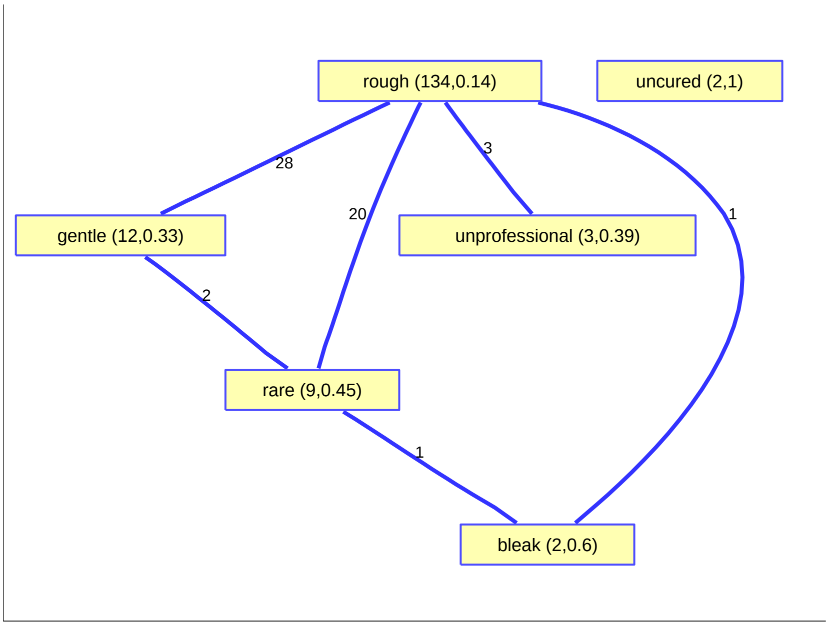


Fig. 13 Clustering of **raw** into 6 clusters.

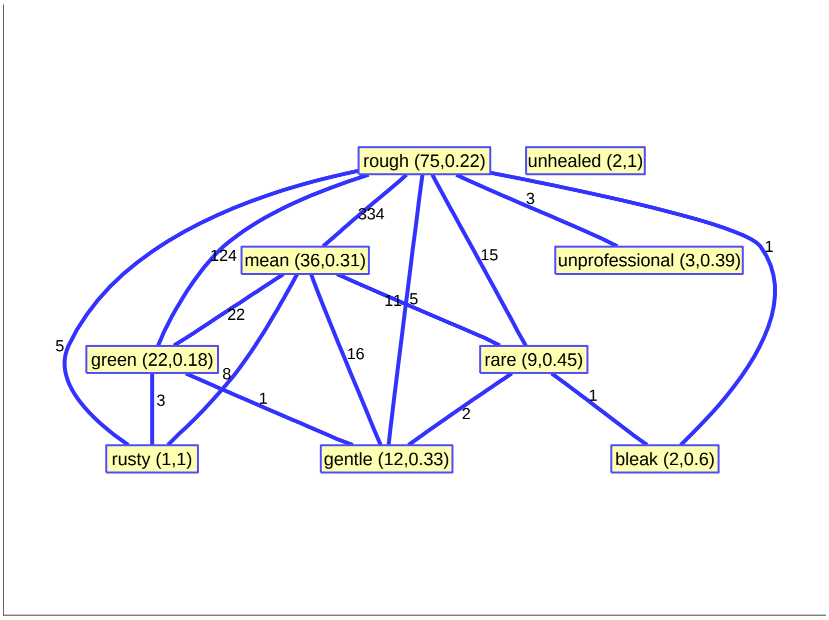


Fig. 14 Clustering of **raw** into 9 clusters.

end-product, but something that has to be reviewed and worked upon to fit a given purpose.

As for the means, both approaches use graph structures, but the graph structures model different aspects of a semantic mirror: (i) our graphs are monolingual, while graphs of [20] are bilingual, (ii) our graphs are not lattices; all nodes represent a word and there are no nodes representing meets or joins, (iii) edges in our graphs carry weights, representing the number of different translations common to two words, whereas the lattices in [20] do not.

Priss and Old hold that bilingual dictionaries are likely to be less useful than parallel corpora for the method of semantic mirrors as "different translations of a word in a bilingual dictionary will more often refer to different senses than to synonyms / ... /. For a single sense fewer translations can be expected than would be found in a parallel corpus." We believe this to be false. In a large dictionary, like the one used in our experiments, we will find many rare words, that will not occur at all in even a large parallel corpus[8]. Also, for a large enough parallel corpus, many of the translations will not be synonyms, and thus provide noise when the goal is to find synonym sets.

Liliehöök and Merkel [16] show one way of automatically combining overlapping synonym sets generated from different seed words into unified clusters by applying vector-space models on the output from the approach described in this article.

6 Conclusions

In this paper we have discussed the spectral partitioning of graphs obtained using the mirroring method on a bilingual dictionary. The purpose was to find subgraphs that are only weakly coupled; we conjectured that the subgraphs would correspond to a clustering of words into groups with different meanings.

A lexical entry of a bilingual dictionary, or a similar resource created from a parallel corpus, gives information on a single word only. By making explicit what is implicit in this lexical resource, semantic mirroring enables a user to get a picture of large subsets of related words. In [7], the construction of senses is based on overlaps among the sets that are produced, first as images of individual words and then from images of the sets so constructed. The original seed word is supposed to be part of all subsets. Here, we do not make such a strong assumption, and, in addition, we include the number of common paths for different words when constructing the graph as this is likely to indicate a stronger semantic relationship. By representing the generated subset in this way, we can directly apply methods from spectral graph partitioning to structure our data. Graph partitioning can be applied iteratively yielding finer and finer clusters using the model parameters to guide the process.

Semantic mirroring has several potential applications within lexicography, terminology and ontology. Lexicographers could with the presented approach explore an existing bilingual dictionary and discover missing word senses in lexical entries, and add new synonyms to a thesaurus. For many languages there are no fully developed wordnets, but by using semantic mirroring along the lines presented here,

[8] For example, the following words in our dictionary, mealy, vealy, pallid, unvitiated all have frequency 0 in the English part of the Europarl corpus [15] totalling 45.8 million words.

wordnet-like resources could be created much more efficiently [7]. However, as the outcome is dependent on the basic resource, and languages do not correspond in a simple one-to-one fashion, a human expert is required, at some point, to decide whether a particular cluster corresponds to a sense, or forms a semantically interesting cluster of any kind. The basic approach presented here is centered around an interactive approach, meaning that different thresholds and parameters can be tested interactively by a linguist.

In terminology work, bilingual term lists extracted from parallel domain-specific texts [11] could be fed into the machinery and, given a slightly more developed framework, terms could be grouped into semantically related clusters. If the terminologist has access to an existing termbase, new term candidates have to be connected to existing terms in the termbase in order to make a decision on whether the new terms should be added to the termbase, and to what concept. If terms are added to the term base they can either be added to new non-existing concepts or to already existing concepts. The task of deciding of which status each term should have in a concept cluster is still tricky, e.g. what term is to be recommended, are there terms that should be regarded as forbidden or obsolete? However, the point is that the terminologist could get assistance in how to create or enrich terminologies by using the presented approach.

Methodologically, the technique described can be further developed. The current approach takes a seed word as input and returns a set of word clusters. In real-life applications, this can be looped over all source words in a bilingual dictionary. Such a loop has been implemented and tested in Lilliehöök and Merkel [16] where it was shown that the interactive tool proposed in this article can be extended to be run in an unsupervised fashion and where overlapping synonym sets can be unified into synonym sets of high quality using vector-space models.

We have chosen spectral partitioning because of its simplicity and because compared to other methods it has a solid mathematical foundation, see the discussion in Section 3. For instance, unlike some other clustering and partitioning methods, the algorithm is invariant under different original orderings of the nodes.

In other applications of spectral clustering, where it is essential that all the clusters are reasonably large, it is common to start out from the relaxation or approximation of a partitioning criterion such as Ncut or RatioCut (see e. g. [23, Section 5]). Here we use the the Fiedler vector to find a reordering of the nodes of the graph. Based on that ordering we have partitioned the graph where the value for the corresponding conductance is smallest, even if one of the subgraphs is considerably smaller than the other. We have used both the unnormalized and the normalized graph Laplacian. In our experiments we have seen small differences between these two approaches. In fact, little guidance is given in the spectral partitioning literature concerning qualitative differences between the two alternatives. It may well turn out that the inherent uncertainty in the determination of word senses, see Section 2.1, due to human differences and different regional practices, is as large as that in the choice of computational alternatives.

Acknowledgment

This work was was partially supported by a grant from the Swedish Research Council (VR). We are indebted to three anonymous referees for several suggestions that helped improve the presentation.

References

1. T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Inf. Process. Lett.*, 42:153–159, 1992.
2. I. Chugur, A. Penas, J. Gonzalo, and F. Verdejo. Monolingual and bilingual dictionary approaches to the enrichment of the Spanish WordNet with adjectives. In *Proceedings of WordNet and Other Lexical Resources Workshop, Pittsburgh: NAACL*, 2001.
3. F. Chung. Laplacians of graphs and Cheeger's inequalities. In D. Miklós, V. T. Sós, and T. Szonyi, editors, *Combinatorics, Paul Erdös is Eighty, Vol. 2*, pages 157–172. Vol. 2 of Bolyai Soc. Math. Stud., János Bolyai Math. Soc., Budapest, 1996.
4. F. Chung. *Spectral Graph Theory.* CBMS Regional Conference Series in Mathematics Number 92. American Mathematical Society, 1997.
5. M. Diab and P. Resnik. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 255–262, 2002.
6. H. Dyvik. A translational basis for semantics. In Stig Johansson and Signe Oksefjell, editors, *Corpora and Crosslinguistic Research: Theory, Method and Case Studies*, pages 51–86. Rodopi, 1998.
7. H. Dyvik. Translations as semantic mirrors: from parallel corpora to wordnet. In *23rd International Conference on English Language Research on Computerized Corpora of Modern and Medieval English (ICAME 23). Gothenburg.*, 2002.
8. H. Dyvik. Translations as a Semantic Knowledge Source. In *Proceedings of the Second Baltic Conference on Human Language Technologies*, pages 27–38, Tallinn, 2005. Institute of Cybernetics, Tallinn University of Technology & Institute of the Estonian Language.
9. Martin Fagerlund, Magnus Merkel, Lars Eldén, and Lars Ahrenberg. Computing word senses by semantic mirroring and spectral graph partitioning. In *Proceedings of TextGraphs-5 - 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 103–107, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
10. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge MA, 1998.
11. J. Foo and M. Merkel. Computer aided term bank creation and standardization: Building standardized term banks through automated term extraction and advanced editing tools. In *Terminology in Everyday Life.* John Benjamins, 2010.
12. B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin-Heidelberg, 1999.
13. N. Ide. Cross-lingual sense determination: Can it work? *Computers and the Humanities: Special issue on SENSEVAL*, 34:223–234, 2000.
14. A. Kilgarriff. Thesauruses for natural language processing. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 5 – 13, oct. 2003.
15. P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *The Tenth Machine Translation Summit, Phuket, Thailand, September 12-16*, 2005.
16. H. Lilliehök and M. Merkel. Clustering word senses from semantic mirroring data. In *Proceedings of the workshop on lexical semantic resources for NLP at NODALIDA, Oslo, Norway, NEALT Proceedings Series 19*, 2013.
17. K. Lindén. *Word sense discovery and disambiguation.* PhD thesis, University of Helsinki, Helsinki, 2005.
18. G. A. Miller. Wordnet: A lexical database for English. *Comm. ACM*, 38:39–41, 1995.
19. A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.
20. U. Priss and L. J. Old. Conceptual exploration of semantic mirrors. In *Formal Concept Analysis: Third International Conference, ICFCA 2005*, 2005.

21. P. Resnik and D. Yarowsky. Distinguishing systems and distinguishing senses: new evaluation methods for Word Sense Disambiguation. *Natural Language Engineering*, 5:113–133, 1999.
22. H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2-3):135 – 148, 1991. Parallel Methods on Large-scale Structural Analysis and Physics Applications.
23. U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007. 10.1007/s11222-007-9033-z.