Abstract
We describe and evaluate the application of a spectral clustering technique (Ng et al., 2002) to the unsupervised clustering of German verbs. Our previous work has shown that standard clustering techniques succeed in inducing Levin-style semantic classes from verb subcategorisation information. But clustering in the very high dimensional spaces that we use is fraught with technical and conceptual difficulties. Spectral clustering performs a dimensionality reduction on the verb frame patterns, and provides a robustness and efficiency that standard clustering methods do not display in direct use. The clustering results are evaluated according to the alignment (Cristianini et al., 2002) between the Gram matrix defined by the cluster output and the corresponding matrix defined by a gold standard.

1 Introduction
Standard multivariate clustering technology (such as k-Means) can be applied to the problem of inferring verb classes from information about the estimated prevalence of verb frame patterns (Schulte im Walde and Brew, 2002). But one of the problems with multivariate clustering is that it is something of a black art when applied to high-dimensional natural language data. The search space is very large, and the available techniques for searching this large space do not offer guarantees of global optimality.

In response to this insight, the present work applies a spectral clustering technique (Ng et al., 2002) to the verb frame patterns. At the heart of this approach is a transformation of the original input into a set of orthogonal eigenvectors. We work in the space defined by the first few eigenvectors, using standard clustering techniques in the reduced space. The spectral clustering technique has been shown to handle difficult clustering problems in image processing, offers principled methods for initializing cluster centers, and (in the version that we use) has no random component.

The clustering results are evaluated according to their alignment with a gold standard. Alignment is Pearson correlation between corresponding elements of the Gram matrices, which has been suggested as a measure of agreement between a clustering and a distance measure (Cristianini et al., 2002). We are also able to use this measure to quantify the fit between a clustering result and the distance matrix that serves as input to clustering. The evidence is that the spectral technique is more effective than the methods that have previously been tried.

2 Verb valency description
The data in question come from a subcategorization lexicon induced from a large German newspaper corpus (Schulte im Walde, 2002). The verb valency information is provided in form of probability distributions over verb frames for each verb. There are two conditions: the first with 38 relatively coarse syntactic verb subcategorisation frames, the second a more delicate classification subdividing the verb frames of the first condition using prepositional phrase information (case plus preposition), resulting in 171 possible frame types.

The verb frame types contain at most three arguments. Possible arguments in the frames are nominative (n), dative (d) and accusative (a) noun phrases, reflexive pronouns (r), prepositional phrases (p), expletive es (x), non-finite clauses (i), finite clauses (s-2 for verb second clauses, s-dass for dass-clauses, s-ob for ob-clauses, s-w for indirect wh-questions), and copula constructions (k). For example, subcate-
recognising a direct (accusative case) object and a non-finite clause would be represented by nai. Table 1 shows an example distribution for the verb glauben ‘to think/believe’. The more delicate version of subcategorisation frame was done by distributing the frequency mass of prepositional phrase frame types (np, nap, ndp, npr, xp) over the prepositional phrases, according to their frequencies in the corpus. Prepositional phrases are referred to by case and preposition, such as ‘Dat.mit’, ‘Akk.für’. The present work uses the latter, more delicate, verb valency descriptions.

| Frame    | Prob  | Frame  | Prob  |
|----------|-------|--------|-------|
| ns-dass  | 0.27945 | npr    | 0.00261 |
| ns-2     | 0.27358 | nds-dass | 0.00253 |
| np       | 0.09551 | ndi    | 0.00161 |
| n        | 0.08811 | nrs-dass | 0.00029 |
| na       | 0.08046 | ndr    | 0.00029 |
| ni       | 0.05015 | nrs-w  | 0.00029 |
| nd       | 0.03932 | nir    | 0.00027 |
| nad      | 0.03235 | nds-w  | 0.00024 |
| nds-2    | 0.01011 | xd     | 0.00017 |
| nai      | 0.00894 | ns-ob  | 0.00014 |
| ns-w     | 0.00859 | nas-ob | 0.00014 |
| nas-w    | 0.00681 | nds-ob | 0.00000 |
| nap      | 0.00594 | nrs-ob | 0.00000 |
| nr       | 0.00455 | x      | 0.00000 |
| nar      | 0.00436 | xa     | 0.00000 |
| nrs-2    | 0.00391 | xp     | 0.00000 |
| ndp      | 0.00356 | xr     | 0.00000 |
| nas-dass | 0.00342 | xs-dass | 0.00000 |
| nas-2    | 0.00281 | k      | 0.00000 |

Table 1: Probability distribution for glauben

3 Problems with standard clustering

Our previous work on the valency data applied k-Means (a standard technique) to the task of inducing semantic classes for German verbs (Schulte im Walde and Brew, 2002). We compared the results of k-Means clustering with a gold standard set prepared according to the principles of verb classification advocated by (Levin, 1993), and reported on the sensitivity of the classes to linguistically motivated “lesioning” of the input verb frame. The verb classes we used are listed in Table 2.

The verb classes are closely related to Levin’s English classes. They also agree with the German verb classification in (Schumacher, 1986) as far as the relevant verbs appear in his less extensive semantic ‘fields’. The rough glosses and the references to Levin’s classes in the table are primarily to aid the intuition of non-native speakers of German.

Clustering can be thought of as a process that finds a discrete approximation to a distance measure. For any data set of \( n \) items over which a distance measure is defined, the Gram matrix is the symmetric \( n \times n \) matrix whose elements \( M_{ij} \) are the distances between items \( i \) and \( j \). The diagonal elements \( M_{ii} \) of this matrix will all be 0. Every clustering corresponds to a block-diagonal Gram matrix. Clustering \( n \) items into \( k \) classes corresponds to the choice of an ordering for the labels on the axes of the Gram matrix and the choice of \( k - 1 \) change points marking the boundaries between the blocks. Thus, the search space of clusters is very large. The available techniques for searching this large space do not (and probably cannot) offer guarantees of global optimality. Standard nostrums include transformations of the underlying data, and the deployment of different strategies for initializing the cluster centers. These may produce intuitively attractive clusters, but when we apply these ideas to our verb frame data many questions remain, including the following:

- When the solutions found by clustering differ from our intuitions, is this because of failures in the features used, the clustering techniques, or the intuitions?
- How close are the local optima found by the clustering techniques to the best solutions in the space defined by the data?
- Is it even appropriate to use frequency information for this problem? Or would it suffice to characterize verb classes by the pattern of frames that their members can inhabit, without regard to frequency?
- Does the data support some clusters more strongly than others? Are all the distinctions made in classifications such as Levin’s of equal validity?

In response to these questions, the present paper describes an application to the verb data of a particular spectral clustering technique (Ng et al., 2002). At the heart of this approach is a transformation of the original verb frame data into a set of orthogonal eigenvectors. We work
in the space defined by the first few eigenvectors, using standard clustering techniques in the transformed space.

4 The spectral clustering algorithm

The spectral clustering algorithm takes as input a matrix formed from a pairwise similarity function over a set of data points. In image segmentation two pixels might be declared similar if they have similar intensity, similar hue or similar location, or if a local edge-detection algorithm does not place an edge between them. The technique is generic, and as (Longuet-Higgins and Scott, 1990) point out, originated not in computer science or AI but in molecular physics. Most of the authors nevertheless “adopt the terminology of image segmentation (i.e. the data points are pixels and the set of pixels is the image), keeping in mind that all the results are also valid for similarity-based clustering” (Meilă and Shi, 2001). Our natural language application of the technique uses straightforward similarity measures based on verb frame statistics, but nothing in the algorithm hinges on this, and we plan in future work to elaborate our similarity measures. Although there are several roughly analogous spectral clustering techniques in the recent literature (Meilă and Shi, 2001; Longuet-Higgins and Scott, 1990; Weiss, 1999), we use the algorithm from (Ng et al., 2002) because it is simple to implement and understand.

Here are the key steps of that algorithm: Given a set of points $S = \{s_1, \ldots, s_n\}$ in a high dimensional space.

1. Form a distance matrix $D \in \mathbb{R}^2$. For (Ng et al., 2002) this distance measure is Euclidean, but other measures also make sense.

2. Transform the distance matrix to an affinity matrix by $A_{ij} = \exp(-\frac{D_{ij}^2}{\sigma^2})$ if $i \neq j$, 0 if $i = j$. The free parameter $\sigma^2$ controls the rate at which affinity drops off with distance.

3. Form the diagonal matrix $D$ whose element is the sum of $A$’s $i$th row, and create the matrix $L = D^{-1/2}AD^{-1/2}$. 

Table 2: Levin-style verb classes for German

| Aspect                      | 55.1 | anfangen, aufhören, beenden, beginnen, enden |
|-----------------------------|------|---------------------------------------------|
| Propositional Attitude      | 29.5 | ahnen, denken, glauben, vermuten, wissen   |
|                             |      | sense, think, think, guess, know            |
| Transfer of Possession      | 29.5 | bekommen, erhalten, erlangen, kriegen       |
| (Obtaining)                 |      | receive, obtain, acquire, get               |
| Transfer of Possession      | 11.1 | bringen, liefern, schicken, vermitteln, zustellen |
| (Supply)                    |      | bring, deliver, send, procure, deliver       |
| Manner of Motion            | 51.4 | fahren, fliegen, rudern, segeln              |
|                             |      | drive, fly, row, sail                        |
| Emotion                     | 31.1 | ärgeren, freuen                              |
|                             |      | irritate, delight                           |
| Announcement                | 37.7 | ankündigen, bekanntgeben, eröffnen, verkünden|
|                             |      | announce, make known, disclose, proclaim     |
| Description                 | 29.2 | beschreiben, charakterisieren, darstellen, interpretieren |
|                             |      | describe, characterise, present, interpret   |
| Insistence                  | -    | beharren, bestehen, insistieren, pochen      |
|                             |      | all mean insist                             |
| Position                    | 50   | liegen, sitzen, stehen                      |
|                             |      | lie, sit, stand                             |
| Support                     | -    | dienen, folgen, helfen, unterstützen        |
|                             |      | serve, follow, help, support                |
| Opening                     | 45.4 | öffnen, schließen                           |
|                             |      | open, close                                 |
| Consumption                 | 39.4 | essen, konsumieren, lesen, saufen, trinken   |
|                             |      | eat, consume, read, drink (esp. animals or drunkards), drink |
| Weather                     | 57   | blitzen, donnern, dämmern, nieseln, regnen, schneien |
|                             |      | flash, thunder, dawn/grow dusky, drizzle, rain, snow |

The spectral clustering algorithm takes as input a matrix formed from a pairwise similarity function over a set of data points. In image segmentation two pixels might be declared similar if they have similar intensity, similar hue or similar location, or if a local edge-detection algorithm does not place an edge between them. The technique is generic, and as (Longuet-Higgins and Scott, 1990) point out, originated not in computer science or AI but in molecular physics. Most of the authors nevertheless “adopt the terminology of image segmentation (i.e. the data points are pixels and the set of pixels is the image), keeping in mind that all the results are also valid for similarity-based clustering” (Meilă and Shi, 2001). Our natural language application of the technique uses straightforward similarity measures based on verb frame statistics, but nothing in the algorithm hinges on this, and we plan in future work to elaborate our similarity measures. Although there are several roughly analogous spectral clustering techniques in the recent literature (Meilă and Shi, 2001; Longuet-Higgins and Scott, 1990; Weiss, 1999), we use the algorithm from (Ng et al., 2002) because it is simple to implement and understand.

Here are the key steps of that algorithm: Given a set of points $S = \{s_1, \ldots, s_n\}$ in a high dimensional space.

1. Form a distance matrix $D \in \mathbb{R}^2$. For (Ng et al., 2002) this distance measure is Euclidean, but other measures also make sense.

2. Transform the distance matrix to an affinity matrix by $A_{ij} = \exp(-\frac{D_{ij}^2}{\sigma^2})$ if $i \neq j$, 0 if $i = j$. The free parameter $\sigma^2$ controls the rate at which affinity drops off with distance.

3. Form the diagonal matrix $D$ whose element is the sum of $A$’s $i$th row, and create the matrix $L = D^{-1/2}AD^{-1/2}$. 

Table 2: Levin-style verb classes for German

| Aspect                      | 55.1 | anfangen, aufhören, beenden, beginnen, enden |
|-----------------------------|------|---------------------------------------------|
| Propositional Attitude      | 29.5 | ahnen, denken, glauben, vermuten, wissen   |
|                             |      | sense, think, think, guess, know            |
| Transfer of Possession      | 29.5 | bekommen, erhalten, erlangen, kriegen       |
| (Obtaining)                 |      | receive, obtain, acquire, get               |
| Transfer of Possession      | 11.1 | bringen, liefern, schicken, vermitteln, zustellen |
| (Supply)                    |      | bring, deliver, send, procure, deliver       |
| Manner of Motion            | 51.4 | fahren, fliegen, rudern, segeln              |
|                             |      | drive, fly, row, sail                        |
| Emotion                     | 31.1 | ärgeren, freuen                              |
|                             |      | irritate, delight                           |
| Announcement                | 37.7 | ankündigen, bekanntgeben, eröffnen, verkünden|
|                             |      | announce, make known, disclose, proclaim     |
| Description                 | 29.2 | beschreiben, charakterisieren, darstellen, interpretieren |
|                             |      | describe, characterise, present, interpret   |
| Insistence                  | -    | beharren, bestehen, insistieren, pochen      |
|                             |      | all mean insist                             |
| Position                    | 50   | liegen, sitzen, stehen                      |
|                             |      | lie, sit, stand                             |
| Support                     | -    | dienen, folgen, helfen, unterstützen        |
|                             |      | serve, follow, help, support                |
| Opening                     | 45.4 | öffnen, schließen                           |
|                             |      | open, close                                 |
| Consumption                 | 39.4 | essen, konsumieren, lesen, saufen, trinken   |
|                             |      | eat, consume, read, drink (esp. animals or drunkards), drink |
| Weather                     | 57   | blitzen, donnern, dämmern, nieseln, regnen, schneien |
|                             |      | flash, thunder, dawn/grow dusky, drizzle, rain, snow |
4. Obtain the eigenvectors and eigenvalues of \( L \).

5. Form a new matrix from the vectors associated with the \( k \) largest eigenvalues. Choose \( k \) either by stipulation or by picking sufficient eigenvectors to cover 95\% of the variance.

6. Each item now has a vector of \( k \) coordinates in the transformed space. Normalize these vectors to unit length.

7. Cluster in \( k \)-dimensional space. Following (Ng et al., 2002) we use k-Means for this purpose, but any other algorithm that produces tight clusters could fill the same role. In (Ng et al., 2002) an analysis demonstrates that there are likely to be \( k \) well-separated clusters.

We carry out the whole procedure for a range of values of \( \sigma \). In our experiments \( \sigma \) is searched in steps of 0.001 from 0.01 to 0.059, since that always sufficed to find the best aligned set of clusters. If \( \sigma \) is set too low no useful eigenvectors are returned, but this situation is easy to detect. We take the solution with the best alignment (see definition below) to the (original) distance measure. This is how (Cristianini et al., 2002) choose the best solution, while (Ng et al., 2002) explain that they choose the solution with the tightest clusters, without being specific on how this is done.

In general it matters how initialization of cluster centers is done for algorithms like k-Means. (Ng et al., 2002) provide a neat initialization strategy, based on the expectation that the clusters in their space will be orthogonal. They select the first cluster center to be a randomly chosen data point, then search the remaining data points for the one most orthogonal to that. For the third data point they look for one that is most orthogonal to the previous two, and so on until sufficient have been obtained. We modify this strategy slightly, removing the random component by initializing \( n \) times, starting out at each data point in turn. This is fairly costly, but improves results, and is less expensive than the random initializations and multiple runs often used with k-Means.

---

1Srini Parthasarathy suggested this dodge for allowing the eigenvalues to select the appropriate number of clusters.

5 Experiments and evaluation

We clustered the verb frames data using our version of the algorithm in (Ng et al., 2002). To calculate the distance \( d \) between two verbs \( v_1 \) and \( v_2 \) we used a range of measures: the cosine of the angle between the two vectors of frame probabilities, a flattened version of the cosine measure in which all non-zero counts are replaced by 1.0 (labelled \( bcos \), for binarized cosine, in Table 3), and skew divergence, recently shown as an effective measure for distributional similarity (Lee, 2001). This last is defined in terms of KL-divergence, and includes a free weight parameter \( w \), which we set to 0.9, following (Lee, 2001), Skew-divergence is asymmetric in its arguments, but our technique needs a symmetric measure, so we calculate it in both directions and use the larger value.

Table 3 contains four results for each of three distance measures (\( cos, bcos \) and \( skew \)). The first line of each set gives the results when the spectral algorithm is provided with the prior knowledge that \( k = 14 \). The second line gives the results when the standard k-Means algorithm is used, again with \( k = 14 \). In the third line of each set, the value of \( k \) is determined from the eigenvalues, as described above. For \( cos \) 12 clusters are chosen, for \( bcos \) the chosen value is 17, and for \( skew \) it is 16. The final line of each set gives the results when the standard algorithm is used, but \( k \) is set to the value selected for that distance measure by the spectral method.

For standard k-Means, the initialization strategy from (Ng et al., 2002) does not apply (and does not work well in any case), so we used 100 random replications of the initialization, each time initializing the cluster centers with \( k \) randomly chosen data points. We report the result that had the highest alignment with the distance measure (cf. Section 5.1).

(Meiša and Shi, 2001) provide analysis indicating that their \( MNcut \) algorithm (another spectral clustering technique) will be exact when the eigenvectors used for clustering are piecewise constant. Figure 1 shows the top 16 eigenvectors of a distance matrix based on skew divergence, with the items sorted by the first eigenvector. Most of the eigenvectors appear to be piecewise constant, suggesting that the conditions for good performance in clustering are indeed present in the language data. Many of
the eigenvectors appear to correspond to a partition of the data into a small number of tight clusters. Taken as a whole they induce the clusterings reported in Table 3.

5.1 Alignment as an evaluation tool
Pearson correlation between corresponding elements of the Gram matrices has been suggested as a measure of agreement between a clustering and a distance measure (Cristianini et al., 2002). Since we can convert a clustering into a distance measure, alignment can be used in a number of ways, including comparison of clusterings against each other.

For evaluation, three alignment-based measures are particularly relevant:

- The alignment between the gold standard and the distance measure reflects the presence or absence in the distance measure of evidential support for the relationships that the clustering algorithm is supposed to infer. This is the column labelled “Support” in Table 3.
- The alignment between the clusters inferred by the algorithm and the distance measure reflects the confidence that the algorithm has in the relationships that it has chosen. This is the column labelled “Confidence” in Table 3.
- The alignment between the gold standard and the inferred clusters reflects the quality of the result. This is the column labelled “Quality” in Table 3.

We hope that when the algorithms are confident they will also be right, and that when the data strongly supports a distinction the algorithms will find it.

5.2 Results
Table 3 contains our data. The columns based on various forms of alignment have been discussed above. Clusterings are also sets of pairs, so, when the Gram matrices are discrete, we can also provide the standard measures of precision, recall and F-measure. Usually it is irrelevant whether we choose alignment or the standard measures, but the latter can yield unexpected results for extreme clusterings (many small clusters or few very big clusters). The remaining columns provide these conventional performance measures.

For all the evaluation methods and all the distance measures that we have tried, the algorithm from (Ng et al., 2002) does better than direct clustering, usually finding a clustering that aligns better with the distance measure than does the gold standard. Deficiencies in the result are due to weaknesses in the distance measures or the original count data, rather than search errors committed by the clustering algorithm. Skew divergence is the best distance measure, cosine is less good and cosine on binarized data the worst.

5.3 Which verbs and clusters are hard?
All three alignment measures can be applied to a clustering as whole, as above, or restricted to a subset of the Gram matrix. These can tell us how well each verb and each cluster matches the distance measure (or indeed the gold standard). To compute alignment for a verb we cal-

| Algorithm   | k | Support | Confidence | Quality | Precision | Recall | F-Measure |
|-------------|---|---------|------------|---------|-----------|--------|-----------|
| Cos (Ng)    | 14| 0.80    | 0.83       | 0.81    | 0.30      | 0.43   | 0.35      |
| Cos (Direct)| 14| -       | 0.78       | 0.74    | 0.21      | 0.44   | 0.28      |
| Cos (Ng)    | 12| -       | 0.79       | 0.81    | 0.26      | 0.40   | 0.32      |
| Cos (Direct)| 12| -       | 0.72       | 0.79    | 0.20      | 0.45   | 0.28      |
| BCos (Ng)   | 14| 0.86    | 0.86       | 0.86    | 0.21      | 0.23   | 0.22      |
| BCos (Direct)| 14| -       | 0.81       | 0.78    | 0.16      | 0.21   | 0.18      |
| BCos (Ng)   | 17| -       | 0.86       | 0.83    | 0.28      | 0.23   | 0.25      |
| BCos (Direct)| 17| -       | 0.87       | 0.80    | 0.13      | 0.11   | 0.12      |
| Skew (Ng)   | 14| 0.84    | 0.85       | 0.85    | 0.37      | 0.47   | 0.41      |
| Skew (Direct)| 14| -       | 0.84       | 0.78    | 0.22      | 0.34   | 0.27      |
| Skew (Ng)   | 16| -       | 0.86       | 0.88    | 0.49      | 0.47   | 0.48      |
| Skew (Direct)| 16| -       | 0.84       | 0.84    | 0.35      | 0.41   | 0.37      |

Table 3: Performance of the clustering algorithms
Figure 1: The top 16 eigenvectors of the distance matrix

culate Spearman correlation over its row of the Gram matrix. For a cluster we do the same, but over all the rows corresponding to the cluster members. The second column of Table 4, labelled “Support”, gives the contribution of that verb to the alignment between the gold standard clustering and the skew-divergence distance measure (that is, the empirical support that the distance measure gives to the human-preferred placement of the verb). The third column, labelled “Confidence” contains the contribution of the verb to the alignment between the skew-divergence and the clustering inferred by our algorithm (this is the measure of the confidence that the clustering algorithm has in the correctness of its placement of the verb, and is what is maximized by Ng’s algorithm as we vary $\sigma$). The fourth column, labelled “Correctness”, measures the contribution of the verb to the alignment between the inferred cluster and the gold standard (this is the measure of how correctly the verb was placed). To get a feel for performance at the cluster level we measured the alignment with the gold standard. We merged and ranked the lists proposed by skew divergence and binary cosine. The figure of merit, labelled “Score” is the geometric mean of the alignments for the members of the cluster. The second column, labelled “Method”, indicates which distance measure or measures produced this cluster. Table 5 shows this ranking. Two highly ranked clusters (Emotion and a large subset of Weather) are selected by both distance measures. The highest ranked cluster proposed only by binary cosine is a sub-

| Verb   | Support | Confidence | Correctness |
|--------|---------|------------|-------------|
| freuen | 0.97    | 0.97       | 1.00        |
| ärger  | 0.95    | 0.95       | 1.00        |
| stehen | 0.93    | 0.93       | 1.00        |
| sitzen | 0.93    | 0.93       | 1.00        |
| liegen | 0.92    | 0.92       | 1.00        |
| glauben| 0.90    | 0.96       | 0.89        |
| dienen | 0.90    | 0.94       | 0.96        |
| pochen | 0.89    | 0.91       | 0.96        |
| beharren| 0.89   | 0.91       | 0.96        |
| segeln | 0.89    | 0.87       | 0.89        |
| ...    | 0.87    | 0.92       | 0.93        |
| ...    | 0.82    | 0.87       | 0.93        |
| ...    | 0.76    | 0.86       | 0.68        |
| unterstützen | 0.71 | 0.79 | 0.68 |
| beenden | 0.68    | 0.80       | 0.65        |

Table 4: Empirical support, confidence and alignment for skew-divergence

set of Position, but this is dominated by skew-divergence’s correct identification of the whole class (see Table 2 for a reminder of the definitions of these classes). The systematic superiority of the probabilistic measure suggests that there is after all useful information about verb classes in the non-categorical part of our verb frame data.

6 Related work

Levin’s (Levin, 1993) classification has provoked several studies that aim to acquire lexical semantic information from corpora using cues pertaining to mainly syntactic structure
Table 5: Cluster quality by origin

(Levin, 1993) use grammatical features (acquired from corpora) to classify verbs into three semantic classes: unergative, unaccusative, and object-drop. These classes are abstractions of Levin’s (Levin, 1993) classes and as a result yield a coarser classification. The classifier used is a decision tree learner. (Schulte im Walde, 2000) uses subcategorization information and selectional restrictions to cluster verbs into (Levin, 1993) compatible semantic classes. Subcategorization frames are induced from the BNC using a robust statistical parser (Carroll and Rooth, 1998). The selectional restrictions are acquired using Resnik’s (Resnik, 1993) information-theoretic measure of selectional association which combines distributional and taxonomic information in order to formalise how well a predicate associates with a given argument.

7 Conclusions

We have described the application to natural language data of a spectral clustering technique (Ng et al., 2002) closely related to kernel PCA (Cristianini et al., 2002). We have presented evidence that the dimensionality reduction involved in the clustering technique can give k-Means a robustness that it does not display in direct use. The solutions found by the spectral clustering are always at least as well-aligned with the distance measure as is the gold standard measure produced by human intuition, but this does not hold when k-Means is used directly on the untransformed data.

Since we work in a transformed space of low dimensionality, we gain efficiency, and we no longer have to sum and average data points in the original space associated with the verb frame data. In principle, this gives us the freedom to use, as is standardly done with SVMs (Cristianini and Shawe-Taylor, 2000), extremely high dimensional representations for which it would not be convenient to use k-Means directly. We could for instance use features which are derived not from the counts of a single frame but of two or more. This is linguistically desirable, since Levin’s verb classes are defined primarily in terms of alternations rather than in terms of single frames. We plan to explore this possibility in future work.

It is also clearly against the spirit of (Levin, 1993) to insist that verbs should belong to only one cluster, since, for example, both the German “dämmern” and the English “dawn” are clearly related both to verbs associated with weather and natural phenomena (because of “Day dawns.”) and to verbs of cognition (because of “It dawned on Kim that . . . ”). In order to accommodate this, we are exploring the consequences of replacing the k-Means step of our algorithm with an appropriate soft clustering technique.

References

Glenn Carroll and Mats Rooth. 1998. Valence induction with a head-lexicalized PCFG. In Nancy Ide and Atro Voutilainen, editors, Proceedings of the 3rd Conference on Empiri-
Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press.

Nello Cristianini, John Shawe-Taylor, and Jaz Kandola. 2002. Spectral kernel methods for clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Hoa Trang Dang, Joseph Rosenzweig, and Martha Palmer. 1997. Associating semantic components with intersective Levin classes. In *Proceedings of the 1st AMTA SIG-IL Workshop on Interlinguas*, pages 1–8, San Diego, CA.

Bonnie J. Dorr and Doug Jones. 1996. Role of word sense disambiguation in lexical acquisition: Predicting semantics from syntactic cues. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 322–327, Copenhagen, Denmark.

Bonnie J. Dorr. 1997. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):371–322.

Maria Lapata and Chris Brew. 1999. Using subcategorization to resolve verb class ambiguity. In Pascal Fung and Joe Zhou, editors, *Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, College Park, Maryland.

Maria Lapata. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 397–404, College Park, MD.

Lillian Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics 2001*, pages 65–72.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

H. Christopher Longuet-Higgins and Guy L. Scott. 1990. Feature grouping by 'relocalisation' of eigenvectors of the proximity matrix. In *Proceedings of the British Machine Vision Conference*, pages 103–8, Oxford, UK.

Diana McCarthy. 2000. Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of the 1st North American Annual Meeting of the Association for Computational Linguistics*, pages 256–263, Seattle, WA.

Marina Meilă and Jianbo Shi. 2001. A random walks view of spectral segmentation. In *Artificial Intelligence and Statistics 2001*.

Paola Merlo and Susanne Stevenson. 2001. Automatic verb classification based on statistical distribution of argument structure. *Computational Linguistics*, 27(3):373–408.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Philip Stuart Resnik. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

Sabine Schulte im Walde and Chris Brew. 2002. Inducing german semantic verb classes from purely syntactic subcategorization information. In *Association for Computational Linguistics, 40th Anniversary Meeting*, Philadelphia, Pa. To appear.

Sabine Schulte im Walde. 2000. Clustering verbs semantically according to their alternation behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 747–753, Saarbrücken, Germany.

Sabine Schulte im Walde. 2002. A subcategorisation lexicon for german verbs induced from a lexicalised PCFG. In *Proceedings of the 3rd Conference on Language Resources and Evaluation*, Las Palmas, Spain. To appear.

Helmut Schumacher. 1986. *Verben in Feldern*. de Gruyter, Berlin.

Yair Weiss. 1999. Segmentation using eigenvectors: A unifying view. In *ICCV (2)*, pages 975–982.