Hierarchical Verb Clustering Using Graph Factorization

Lin Sun and Anna Korhonen
University of Cambridge, Computer Laboratory
15 JJ Thomson Avenue, Cambridge CB3 0GD, UK
ls418,alk23@cl.cam.ac.uk

Abstract

Most previous research on verb clustering has focussed on acquiring flat classifications from corpus data, although many manually built classifications are taxonomic in nature. Also Natural Language Processing (NLP) applications benefit from taxonomic classifications because they vary in terms of the granularity they require from a classification. We introduce a new clustering method called Hierarchical Graph Factorization Clustering (HGFC) and extend it so that it is optimal for the task. Our results show that HGFC outperforms the frequently used agglomerative clustering on a hierarchical test set extracted from VerbNet, and that it yields state-of-the-art performance also on a flat test set. We demonstrate how the method can be used to acquire novel classifications as well as to extend existing ones on the basis of some prior knowledge about the classification.

1 Introduction

A variety of verb classifications have been built to support NLP tasks. These include syntactic and semantic classifications, as well as ones which integrate aspects of both (Grishman et al., 1994; Miller, 1995; Baker et al., 1998; Palmer et al., 2005; Kipper, 2005; Hovy et al., 2006). Classifications which integrate a wide range of linguistic properties can be particularly useful for tasks suffering from data sparseness. One such classification is the taxonomy of English verbs proposed by Levin (1993) which is based on shared (morpho-)syntactic and semantic properties of verbs. Levin's taxonomy or its extended version in VerbNet (Kipper, 2005) has proved helpful for various NLP application tasks, including e.g. parsing, word sense disambiguation, semantic role labeling, information extraction, question-answering, and machine translation (Swier and Stevenson, 2004; Dang, 2004; Shi and Mihalcea, 2005; Zapirain et al., 2008).

Because verbs change their meaning and behaviour across domains, it is important to be able to tune existing classifications as well to build novel ones in a cost-effective manner, when required. In recent years, a variety of approaches have been proposed for automatic induction of Levin style classes from corpus data which could be used for this purpose (Schulte im Walde, 2006; Joanis et al., 2008; Sun et al., 2008; Li and Brew, 2008; Korhonen et al., 2008; Ó Séaghdha and Copestake, 2008; Vlachos et al., 2009). The best of such approaches have yielded promising results. However, they have mostly focussed on acquiring and evaluating flat classifications. Levin's classification is not flat, but taxonomic in nature, which is practical for NLP purposes since applications differ in terms of the granularity they require from a classification.

In this paper, we experiment with hierarchical Levin-style clustering. We adopt as our baseline method a well-known hierarchical method – agglomerative clustering (AGG) – which has been previously used to acquire flat Levin-style classifications (Stevenson and Joanis, 2003) as well as hierarchical verb classifications not based on Levin (Ferrer, 2004; Schulte im Walde, 2008). The method has also been popular in the related task of noun clus-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1023–1033,
Edinburgh, Scotland, UK, July 27–31, 2011.

tering (Ushioda, 1996; Matsuo et al., 2006; Bassiou and Kotropoulos, 2011).

We introduce then a new method called Hierarchical Graph Factorization Clustering (HGFC) (Yu et al., 2006). This graph-based, probabilistic clustering algorithm has some clear advantages over AGG (e.g. it delays the decision on a verb's cluster membership at any level until a full graph is available, minimising the problem of error propagation) and it has been shown to perform better than several other hierarchical clustering methods in recent comparisons (Yu et al., 2006). The method has been applied to the identification of social network communities (Lin et al., 2008), but has not been used (to the best of our knowledge) in NLP before.

We modify HGFC with a new tree extraction algorithm which ensures a more consistent result, and we propose two novel extensions to it. The first is a method for automatically determining the tree structure (i.e. number of clusters to be produced for each level of the hierarchy). This avoids the need to predetermine the number of clusters manually. The second is addition of soft constraints to guide the clustering performance (Vlachos et al., 2009). This is useful for situations where a partial (e.g. a flat) verb classification is available and the goal is to extend it.

Adopting a set of lexical and syntactic features which have performed well in previous works, we compare the performance of the two methods on test sets extracted from Levin and VerbNet. When evaluated on a flat clustering task, HGFC outperforms AGG and performs very similarly with the best flat clustering method reported on the same test set (Sun and Korhonen, 2009). When evaluated on a hierarchical task, HGFC performs considerably better than AGG at all levels of gold standard classification. The constrained version of HGFC performs the best, as expected, demonstrating the usefulness of soft constraints for extending partial classifications.

Our qualitative analysis shows that HGFC is capable of detecting novel information not included in our gold standards. The unconstrained version can be used to acquire novel classifications from scratch while the constrained version can be used to extend existing ones with additional class members, classes and levels of hierarchy.

2 Target classification and test sets

The taxonomy of Levin (1993) groups English verbs (e.g. *break, fracture, rip*) into classes (e.g. 45.1 *Break* verbs) on the basis of their shared meaning components and (morpho-)syntactic behaviour, defined in terms of diathesis alternations (e.g. the causative/inchoative alternation, where an NP frame alternates with an intransitive frame: *Tony broke the window* $\leftrightarrow$ *The window broke*). It classifies over 3000 verbs in 57 top level classes, some of which divide further into subclasses. The extended version of the taxonomy in VerbNet (Kipper, 2005) classifies 5757 verbs. Its 5 level taxonomy includes 101 top level and 369 subclasses. We used three gold standards (and corresponding test sets) extracted from these resources in our experiments:

T1: The first gold standard is a flat gold standard which includes 13 classes appearing in Levin's original taxonomy (Stevenson and Joanis, 2003). We included this small gold standard in our experiments so that we could compare the flat version of our method against previously published methods. Following Stevenson and Joanis (2003), we selected 20 verbs from each class which occur at least 100 times in our corpus. This gave us 260 verbs in total.

T2: The second gold standard is a large, hierarchical gold standard which we extracted from VerbNet as follows: 1) We removed all the verbs that have less than 1000 occurrences in our corpus. 2) In order to minimise the problem of polysemy, we assigned each verb to the class which, according to VerbNet, corresponds to its predominant sense in WordNet (Miller, 1995). 3) In order to minimise the sparse data problem with very fine-grained classes, we converted the resulting classification into a 3-level representation so that the classes at the 4th and 5th level were combined. For example, the sub-classes of *Declare* verbs (numbered as 29.4.1.1.$\{$1,2,3$\}$) were combined into 29.4.1. 4) The classes that have fewer than 5 members were discarded. The total number of verb senses in the resulting gold standard is 1750, which is 33.2% of the verbs in VerbNet. T2 has 51 top level, 117 second level, and 133 third level classes.

T3: The third gold standard is a subset of T2 where singular classes (top level classes which do not divide into subclasses) are removed. This gold

standard was constructed to enable proper evaluation of the constrained version of HGFC (introduced in the following section) where we want to compare the impact of constraints across several levels of classification. T3 provides classification of 357 verbs into 11 top level, 14 second level, and 32 third level classes.

For each verb appearing in T1-T3, we extracted all the occurrences (up to 10,000) from the British National Corpus (Leech, 1992) and North American News Text Corpus (Graff, 1995).

3 Method

3.1 Features and feature extraction

Previous works on Levin style verb classification have investigated optimal features for this task (Stevenson and Joanis, 2003; Li and Brew, 2008; Sun and Korhonen, 2009)). We adopt for our experiments a set of features which have performed well in recent verb clustering works:

A: Subcategorization frames (SCFs) and their relative frequencies with individual verbs.

B: A with SCFs parameterized for prepositions.

C: B with SCFs parameterized for subjects appearing in grammatical relations associated with the verb in parsed data.

D: B with SCFs parameterized for objects appearing in grammatical relations associated with the verb in parsed data.

These features are purely syntactic. Although semantic features – verb selectional preferences – proved the best (when used in combination with syntactic features) in the recent work of Sun and Korhonen (2009), we left such features for future work because we noticed that different levels of classification are likely to require semantic features at different granularities.

We extracted the syntactic features using the system of Preiss et al. (2007). The system tags, lemmatizes and parses corpus data using the RASP (Robust Accurate Statistical Parsing toolkit (Briscoe et al., 2006)), and on the basis of the resulting grammatical relations, assigns each occurrence of a verb as a member of one of the 168 verbal SCFs. We parameterized the SCFs as described above using the information provided by the system.

3.2 Clustering

We introduce the agglomerative clustering (AGG) and Hierarchical Graph Factorization Clustering (HGFC) methods in the following two subsections, respectively. The subsequent two subsections present our extensions to HGFC: (i) automatically determining the cluster structure and (ii) adding soft constraints to guide clustering performance.

3.2.1 Agglomerative clustering

AGG is a method which treats each verb as a singleton cluster and then successively merges two closest clusters until all the clusters have been merged into one. We used the SciPy's implementation (Oliphant, 2007) of the algorithm. The cluster distance is measured using linkage criteria. We experimented with four commonly used linkage criteria: Single, Average, Complete and Ward's (Ward Jr., 1963). Ward's criterion performed the best and was used in all the experiments in this paper. It measures the increase in variance after two clusters are merged. The output of AGG tends to have excessive number of levels. Cut-based methods (Wu and Leahy, 1993; Shi and Malik, 2000) are frequently applied to extract a simplified view. We followed previous verb clustering works and cut the AGG hierarchy manually.

AGG suffers from two problems. The first is error propagation. When a verb is misclassified at a lower level, the error propagates to all the upper levels. The second is local pairwise merging, i.e. the fact that only two clusters can be combined at any level. For example, in order to group clusters representing Levin classes 9.1, 9.2 and 9.3 into a single cluster representing class 9, the method has to produce intermediate clusters, e.g. 9.{1,2} and 9.3. Such clusters do not always have a semantic interpretation. Although they can be removed using a cut-based method, this requires a pre-defined cut-off value which is difficult to set (Stevenson and Joanis, 2003). In addition, a significant amount of information is lost in pair-wise clustering. In the above example, only the clusters 9.{1,2} and 9.3 are considered, while alternative clusters 9.{1,3} and 9.2 are ignored. Ideally, information about all the possible intermediate clusters should be aggregated, but this is intractable in practice.

3.2.2 Hierarchical Graph Factorization Clustering

Our new method HGFC derives a probabilistic bipartite graph from the similarity matrix (Yu et al., 2006). The local and global clustering structures are learned via the random walk properties of the graph.

The method does not suffer from the above problems with AGG. Firstly, there is no error propagation because the decision on a verb's membership at any level is delayed until the full bipartite graph is available and until a tree structure can be extracted from it by aggregating probabilistic information from all the levels. Secondly, the bipartite graph enables the construction of a hierarchical structure without any intermediate classes. For example, we can group classes 9.{1,2,3} directly into class 9.

We use HGFC with the distributional similarity measure Jensen-Shannon Divergence ($d_{js}(v, v')$). Given a set of verbs, $V = \{v_n\}_{n=1}^{N}$, we compute a similarity matrix W where $W_{ij} = \exp(-d_{js}(v_1, v_2))$. W can be encoded by a undirected graph G (Figure 1(a)), where the verbs are mapped to vertices and the W_{ij} is the edge weight between vertices i and j.

The graph G and the cluster structure can be represented by a bipartite graph $K(V, U)$. V are the vertices on G. $U = \{u_p\}_{p=1}^{m}$ represent the hidden m clusters. For example, looking at Figure 1(b), V on G can be grouped into three clusters u_1, u_2 and u_3. The matrix B denotes the $n \times m$ adjacency matrix, with b_{ip} being the connection weight between the vertex v_i and the cluster u_p. Thus, B represents the connections between clusters at an upper and lower level of clustering. A flat clustering algorithm can be induced by computing B.

The bipartite graph K also induces a similarity (W') between v_i and v_j: $w'_{ij} = \sum_{p=1}^{m} \frac{b_{ip} b_{jp}}{\lambda_p} = (B\Lambda^{-1}B^T)_{ij}$ where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_m)$. Therefore, B can be found by approximating the similarity matrix W of G using W' derived from K. Given a distance function ζ between two similarity matrices, B approximates W by minimizing the cost function $\zeta(W, B\Lambda^{-1}B^T)$. The coupling between B and Λ is removed by setting $H = B\Lambda^{-1}$:

$$\min_{H,\Lambda} \zeta(W, H\Lambda H^T), s.t. \sum_{i=1}^{n} h_{ip} = 1 \quad (1)$$

We use the divergence distance: $\zeta(X, Y) = \sum_{ij}(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij})$. Yu et al. (2006) showed that this cost function is non-increasing under the update rule:

$$\tilde{h}_{ip} \propto h_{ip} \sum_j \frac{w_{ij}}{(H\Lambda H^T)_{ij}} \lambda_p h_{jp} \text{ s.t.} \sum_i \tilde{h}_{ip} = 1 \quad (2)$$

$$\tilde{\lambda}_p \propto \lambda_p \sum_j \frac{w_{ij}}{(H\Lambda H^T)_{ij}} h_{ip} h_{jp} \text{ s.t.} \sum_p \tilde{\lambda}_p = \sum_{ij} w_{ij} \quad (3)$$

w_{ij} can be interpreted as the probability of the direct transition between v_i and v_j: $w_{ij} = p(v_i, v_j)$, when $\sum_{ij} w_{ij} = 1$. b_{ip} can be interpreted as:

$$\begin{aligned} p(u_p, u_q) &= p(u_p)p(u_p|u_q) = \sum_{i=1}^{n} \frac{b_{ip} b_{iq}}{d_i} \\ &= (B^T D^{-1} B)_{pq} \quad (4) \\ D &= \text{diag}(d_1, ..., d_n) \text{ where } d_i = \sum_{p=0}^{m} b_{ip} \end{aligned}$$

$p(u_p, u_q)$ is the similarity between the clusters. It takes into account of a weighted average of contributions from all the data. This is different from the linkage method where only the data from two clusters are considered.

Given the cluster similarity $p(u_p, u_q)$, we can construct a new graph G_1 (Figure 1(d)) with the clusters U as vertices. The cluster algorithm can be applied again (Figure 1(e)). This process can go on iteratively, leading to a hierarchical graph.

Algorithm 1 HGFC algorithm (Yu et al., 2006)

Require: N verbs V, number of clusters m_l for L levels
  Compute the similarity matrix W_0 from V
  Build the graph G_0 from W_0 , and $m_0 \leftarrow n$
  for $l = 1, 2$ to L **do**
    Factorize G_{l-1} to obtain bipartite graph K_l with the adjacency matrix B_l (eq. 1, 2 and 3)
    Build a graph G_l with similarity matrix $W_l = B_l^T D_l^{-1} B_l$ according to equation 4
  end for
  return $B_L, B_{L-1}...B_1$

Additional steps need to be performed in order to extract a tree from the hierarchical graph. Yu et al. (2006) performs the extraction via a propagation of probabilities from the bottom level clusters. For a verb v_i, the probability of assigning it to cluster $v_p^{(l)}$ at level l is given by:

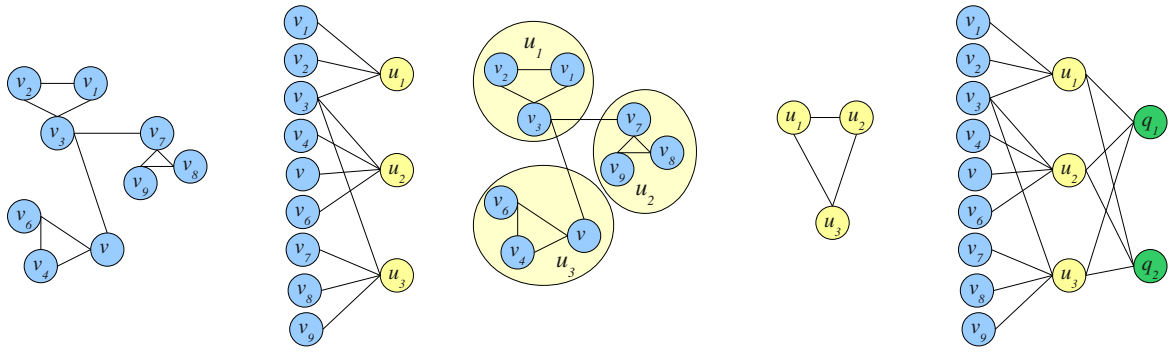

Figure 1: (a) An undirected graph G representing the similarity matrix; (b) The bipartite graph showing three clusters on G; (c) The induced clusters U; (d) The new graph G_1 over clusters U; (e) The new bipartite graph over G_1

$$p(v_p^{(l)}|v_i) = \sum_{V_{l-1}} ... \sum_{V_1} p(v_p^{(l)}|v^{(l-1)})...p(v^{(1)}|v_i)$$
$$= (D_1^{(-1)} B_1 D_2^{-1} B_2 D_3^{-1} B_3 ... D_l^{-1} B_l)_{ip} \quad (5)$$

This method might not extract a consistent tree structure, because the cluster membership at the lower level does not constrain the upper level membership. This prevented us from extracting a Levin style hierarchical classification in our initial experiments. For example, where two verbs were grouped together at a lower level, they could belong to separate clusters at an upper level. We therefore propose a new tree extraction algorithm (Algorithm 2).

The new algorithm starts from the top level bipartite graph, and generates consistent labels for each level by taking into account of the tree constraints set at upper levels.

Algorithm 2 Tree extraction algorithm for HGFC

Require: Given N, (B^l, m_l) on each level for L levels
On the top level L, collect the labels T^L (eq. 5)
Define C to be a $(m_{L-1} \times m_L)$ zero matrix, $C_{ij} \leftarrow 1$, where $i, j = \arg\max_{i,j}\{B_{ij}^L\}$
for $l = L-1$ to 1 **do**
  for $i = 1$ to N **do**
    Compute $p(v_p^l|v_i)$ for each cluster p (eq. 5)
    $t_i^l = \text{argmax}_p\{p(v_p^l|v_i)|p = 1...m_l, C_{pt_i^{l+1}} \neq 0\}$
  end for
  Redefine C to be a $(m_{l-1} \times m_l)$ zero matrix, $C_{ij} \leftarrow 1$, where $i, j = \arg\max_{i,j}\{B_{ij}^l\}$
end for
return Tree consistent labels $T^L, T^{L-1}...T^1$

3.2.3 Automatically determining the number of clusters for HGFC

HGFC needs the number of levels and clusters at each level as input. However, this information is not always available (e.g. when the goal is to actually learn this information automatically). We therefore propose a method for inferring the cluster structure from data. As shown in figure 1, a similarity matrix W models one-hop transitions that follow the links from vertices to neighbors. A walker can also go to other vertices via multi-hop transitions. According to the chain rule of the Markov process, the multi-hop transitions indicate a decaying similarity function on the graph (Yu et al., 2006). After t transitions, the similarity matrix (W_t) becomes:

$$W_t = W_{t-1} D_0^{-1} W_0$$

Yu et al. (2006) proved the correspondence between the HGFC levels (l) and the random walk time: $t = 2^{l-1}$. So the vertices at level l induce a similarity matrix of verbs after t-hop transitions. The decaying similarity function captures the different scales of clustering structure in the data (Azran and Ghahramani, 2006b). The upper levels would have a smaller number of clusters which represent a more global structure. After several levels, all the verbs are expected to be grouped into one cluster. The number of levels and clusters at each level can thus be learned automatically.

We therefore propose a method that uses the decaying similarity function to learn the hierarchical clustering structure. One simple modification to algorithm 1 is to set the number of clusters at level l

(m_l) to be $m_{l-1} - 1$. m is denoted as the number of clusters that have at least one member according to eq. 5. We start by treating each verb as a cluster at the bottom level. The algorithm stops when all the data points are merged into one cluster. The increasingly decaying similarity causes many clusters to have 0 members especially at lower levels, which are pruned in the tree extraction.

3.2.4 Adding constraints to HGFC

The basic version of HGFC makes no prior assumptions about the classification. It is useful for learning novel verb classifications from scratch. However, when wishing to extend an existing classification (e.g. VerbNet) it may be desirable to guide the clustering performance on the basis of information that is already known. We propose a constrained version of HGFC which makes uses of labels at the bottom level to learn upper level classifications. We do this by adding soft constraints to clustering, following Vlachos et al. (2009).

We modify the similarity matrix W as follows: If two verbs have different labels ($l_i \neq l_j$), the similarity between them is decreased by a factor a, and $a < 1$. We set a to 0.5 in the experiments. The resulting tree is generally consistent with the original classification. The influence of the underlying data (domain or features) is reduced according to a.

4 Experimental evaluation

We applied the clustering methods introduced in section 3 to the test sets described in section 2 and evaluated them both quantitatively and qualitatively, as described in the subsequent sections.

4.1 Evaluation methods

We used class based accuracy (ACC) and adjusted rand index (R_{adj}) to evaluate the results on the flat test set T1 (see section 2 for details of T1-T3).

ACC is the proportion of members of dominant clusters DOM-CLUST$_i$ within all classes c_i.

$$\mathrm{ACC} = \frac{\sum_{i=1}^{C} \text{verbs in DOM-CLUST}_i}{\text{number of verbs}}$$

The formula of R_{adj} is (Hubert and Arabie, 1985):

$$\mathrm{R}_{adj} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}] - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}$$

where n_{ij} is the size of the intersection between class i and cluster j.

We used normalized mutual information (NMI) and F-Score (F) to evaluate hierarchical clustering results on T2 and T3. NMI measures the amount of statistical information shared by two random variables representing the clustering result and the gold-standard labels. Given random variables A and B:

$$\mathrm{NMI}(A,B) = \frac{I(A;B)}{[H(A)+H(B)]/2}$$

$$\mathrm{I}(A,B) = \sum_k \sum_j \frac{|(v_k \cap c_j|}{N} \log \frac{N|v_k \cap c_j|}{|v_k||c_j|}$$

where $|v_k \cap c_j|$ is the number of shared membership between cluster v_k and gold-standard class c_j. The normalized variant of mutual information (MI) enables the comparison of clustering with different cluster numbers (Manning et al., 2008).

F is the harmonic mean of precision (P) and recall (R). P is calculated using modified purity – a global measure which evaluates the mean precision of clusters. Each cluster is associated with its prevalent class. The number of verbs in a cluster K that take this class is denoted by $n_{prevalent}$(K).

$$\mathrm{mPUR} = \frac{\sum_{n_{prevalent(k_i)>2}} n_{prevalent(k_i)}}{\text{number of verbs}}$$

R is calculated using ACC.

$$\mathrm{F} = \frac{2 \cdot \mathrm{mPUR} \cdot \mathrm{ACC}}{\mathrm{mPUR} + \mathrm{ACC}}$$

F is not suitable for comparing results with different cluster numbers (Rosenberg and Hirschberg, 2007). Therefore, we only report NMI when the number of classes in clustering and gold-standard is substantially different.

Finally, we supplemented quantitative evaluation with qualitative evaluation of clusters produced by different methods.

4.2 Quantitative evaluation

We first evaluated AGG and the basic (unconstrained) HGFC on the small flat test set T1. The main purpose of this evaluation was to compare the results of our methods against previously published results on the same test set. The number of clusters (K) and levels (L) were inferred automatically for HGFC as described in section 3.2.3. However, to

make the results comparable with previously published ones, we cut the resulting hierarchy at the level of closest match (12 clusters) to the K (13) in the gold-standard. For AGG, we cut the hierarchy at 13 clusters.

Method	ACC	R_{adj}
HGFC	41.2	17.4
AGG (reproduced)	32.7	9.9
AGG (Stevenson and Joanis (2003)	31.0	9.0

Table 1: Comparison against Stevenson and Joanis (2003)'s result on T1 (using similar features).

Table 1 shows our results and the results of Stevenson and Joanis (2003) on T1 when employing AGG using Ward as the linkage criterion. In this experiment, we used the same feature set as Stevenson and Joanis (2003) (set B, see section 3.1) and were therefore able to reproduce their AGG result with a difference smaller than 2%. When using this simple feature set, HGFC outperforms the best performing AGG clearly: 8.5% in ACC and 7.3% in R_{adj}.

We also compared HGFC against the best reported clustering method on T1 to date – that of spectral clustering by Sun and Korhonen (2009). We used the feature sets C and D which are similar to the features (SCF parameterized by lexical prefences) in their experiments. HGFC obtains F of 49.93% on T1 which is 5% lower than the result of Sun and Korhonen (2009). The difference comes from the tree consistency requirement. When the HGFC is forced to produce a flat clustering (a one level tree only), it achieves the F of 52.55% which is very close to the performance of spectral clustering.

We then evaluated our methods on the hierarchical test sets T2 and T3. In the first set of experiments, we pre-defined the tree structure for HGFC by setting L to 3 and K at each level to be the K in the hierarchical gold standard. The hierarchy produced by AGG was cut into 3 levels according to Ks in the gold standard. This enabled direct evaluation of the results against the 3 level gold standards using both NMI and F.

The results are reported in tables 2 and 3. In these tables, N_c is the number of clusters in HGFC clustering while N_l is the number of classes in the gold standard (the two do not always correspond perfectly because a few clusters have zero members).

N_c	N_l	HGFC unconstrained		AGG	
		NMI	F	NMI	F
130	133	57.31	36.65	54.22	32.62
114	117	54.67	37.96	51.35	32.44
50	51	37.75	40.00	32.61	32.78

Table 2: Performance on T2 using a pre-defined tree structure.

N_c	N_l	HGFC unconstrained		HGFC constrained		AGG	
		NMI	F	NMI	F	NMI	F
31	32	51.65	42.01	91.47	92.07	49.70	40.30
15	14	42.75	47.70	82.16	82.80	39.19	43.69
11	11	38.91	51.17	71.69	75.00	34.88	44.80

Table 3: Performance on T3 using a pre-defined tree structure.

Table 2 compares the results of the unconstrained version of HGFC against those of AGG on our largest test set T2. As with T1, HGFC outperforms AGG clearly. The benefit can now be seen at 3 different levels of hierarchy. On average, the HGFC outperforms AGG 3.5% in NMI and 4.8% in F. The difference between the methods becomes clearer when moving towards the upper levels of the hierarchy.

Table 3 shows the results of both unconstrained and constrained versions of HGFC and those of AGG on the test set T3 (where singular classes are removed to enable proper evaluation of the constrained method). The results are generally generally better on this test set than on T2 – which is to be expected since T3 is a refined subset of T2[1].

Recall that the constrained version of HGFC learns the upper levels of classification on the basis of soft constraints set at the bottom level, as described earlier in section 3.2.4. As a consequence, NMI and F are both greater than 90% at the bottom level and the results at the top level are notably lower because the impact of the constraints degrades the further away one moves from the bottom level. Yet, the relatively high result across all levels shows that the constrained version of HGFC can be employed a useful method to extend the hierarchical structure of known classifications.

[1] NMI is higher on T2, however, because NMI has a higher baseline for larger number of clusters (Vinh et al., 2009). NMI is not ideal for comparing the results of T2 and T3.

T2			T3		
N_c	N_l	HGFC	N_c	N_l	HGFC
148	133	53.26	64	32	54.91
97	117	49.85	35	32	50.83
46	51	33.55	20	14	44.02
19	51	25.80	10	14	34.41
9	51	19.17	6	11	32.27
3	51	13.06			

Table 4: NMI of unconstrained HGFC when trees for T2 and T3 are inferred automatically.

Finally, Table 4 shows the results for the unconstrained HGFC on T2 and and T3 when the tree structure is not pre-defined but inferred automatically as described in section 3.2.3. 6 levels are learned for T2 and 5 for T3. The number of clusters produced ranges from 3 to 148 for T2 and from 6 to 64 for T3. We can see that the automatically detected cluster numbers distribute evenly across different levels. The scale of the clustering structure is more complete here than in the gold standards.

In the table, N_c indicates the number of clusters in the inferred tree, while N_l indicates the closest match to the number of classes in the gold standard. This evaluation is not fully reliable because the match between the gold standard and the clustering is poor at some levels of hierarchy. However, it is encouraging to see that the results do not drop dramatically until the match between the two is really poor.

4.3 Qualitative evaluation

To gain a better insight into the performance of HGFC, we conducted further qualitative analysis of the clusters the two versions of this method produced for T3. We focussed on the top level of 11 clusters (in the evaluation against the hierarchical gold standard, see table 3) as the impact of soft constraints is the weakest for the constrained method at this level.

As expected, the constrained HGFC kept many individual verbs belonging to same Verbnet subclass together (e.g. verbs *enjoy, hate, disdain, regret, love, despise, detest, dislike, fear* for the class 31.2.1) so that most clusters simply group lower level classes and their members together. Three nearly clean clusters were produced which only include sub-classes of the same class (e.g. 31.2.0 and 31.2.1 which both belong to 31.2 *Admire* verbs). However, the remaining 8 clusters group together sub-classes (and their members) belonging to unrelated parent classes. Interestingly, 6 of these make both syntactic and semantic sense. For example, several such 37.7 *Say* verbs and 29.5 *Conjencture* verbs are found together which share the meaning of communication and which take similar sentential complements.

In contrast, none of the clusters produced by the unconstrained HGFC represent a single VerbNet class. The majority represent a high number of classes and fewer members per class. Yet many of the clusters make syntactic and semantic sense. A good example is a cluster which includes member verbs from 9.7 *Spray/Load* verbs, 21.2 *Carve* verbs, 51.3.1 *Roll* verbs, and 10.4 *Wipe* verbs. The verbs included in this cluster share the meaning of specific type of motion and show similar syntactic behaviour.

Thorough Levin style investigation of especially the unconstrained method would require looking at shared diathesis alternations between cluster members. We left this for future work. However, the analysis we conducted confirmed that the constrained method could indeed be used for extending known classifications, while the unconstrained method is more suitable for acquiring novel classifications from scratch. The errors in clusters produced by both methods were mostly due to syntactic idiosyncracy and the lack of semantic information in clustering. We plan to address the latter problem in our future work.

5 Discussion and conclusion

We have introduced a new graph-based method – HGFC – to hierarchical verb clustering which avoids some of the problems (e.g. error propagation, pairwise cluster merging) reported with the frequently used AGG method. We modified HGFC so that it can be used to automatically determine the tree structure for clustering, and proposed two extensions to it which make it even more suitable for our task. The first involves automatically determining the number of clusters to be produced, which is useful when this is not known in advance. The second involves adding soft constraints to guide the clustering performance, which is useful when aiming to extend existing classification.

The results reported in the previous section are promising. On a flat test set (T1), the unconstrained version of HGFC outperforms AGG and performs very similarly with the best current flat clustering method (spectral clustering) evaluated on the same dataset. On the hierarchical test sets (T2 and T3), the unconstrained and constrained versions of HGFC outperform AGG clearly at all levels of classification. The constrained version of HGFC detects the missing hierarchy from the existing gold standards with high accuracy. When the number of clusters and levels is learned automatically, the unconstrained method produces a multi-level hierarchy. Our evaluation against a 3-level gold standard shows that such a hierarchy is fairly accurate. Finally, the results from our qualitative evaluation show that both constrained and unconstrained versions of HGFC are capable of learning valuable novel information not included in the gold standards.

The previous work on Levin style verb classification has mostly focussed on flat classifications using methods suitable for flat clustering (Schulte im Walde, 2006; Joanis et al., 2008; Sun et al., 2008; Li and Brew, 2008; Korhonen et al., 2008; Ó Séaghdha and Copestake, 2008; Vlachos et al., 2009). However, some works have employed hierarchical clustering as a method to infer flat clustering.

For example, Schulte im Walde and Brew (2002) employed AGG to initialize the KMeans clustering for German verbs. This gave better results than random initialization. Stevenson and Joanis (2003) used AGG for flat clustering on T1. They cut the hierarchy at the number of classes in the gold standard and found that it is difficult to automatically determine a good cut-off. Our evaluation in the previous section shows that HGFC outperforms their implementation of AGG.

AGG was also used by Ferrer (2004) who performed hierarchical clustering of 514 Spanish verbs. The results were evaluated against a hierarchical gold standard resembling that of Levin's classification in English (Vázquez et al., 2000). R_{adj} of 0.07 was reported for a 15-way classification which is comparable to the result of Stevenson and Joanis (2003).

Hierarchical clustering has also been performed for the related task of semantic verb classification. For example, Basili et al. (1993) identified the problems of AGG, and applied a conceptual clustering algorithm (Fisher, 1987) to Italian verbs. They used semi-automatically acquired semantic roles and the concept types as features. No quantitative results were reported. The qualitative evaluation shows that the resulting clusters are very fine-grained.

Schulte im Walde (2008) performed hierarchical clustering of German verbs using human verb association as features and AGG as a method. They focussed on two small collections of 56 and 104 verbs and evaluated the result against flat gold standard extracted from GermaNet (Kunze and Lemnitzer, 2002) and German FrameNet (Erk et al., 2003), respectively. They reported F of 62.69% for the 56 verbs, and F of 34.68% for the 104 verbs.

In the future, we plan to extend this research line in several directions. First, we will try to determine optimal features for different levels of clustering. For example, the general syntactic features (e.g. SCF) may perform the best at top levels of a hierarchy while more specific or refined features (e.g. SCF+pp) may be optimal at lower levels. We also plan to investigate incorporating semantic features, like verb selectional preferences, in our feature set. It is likely that different levels of clustering require more or less specific selectional preferences. One way to obtain the latter is hierarchical clustering of relevant noun data.

In addition, we plan to apply the unconstrained HGFC to specific domains to investigate its capability to learn novel, previously unknown classifications. As for the constrained version of HGFC, we will conduct a larger scale experiment on the VerbNet data to investigate what kind of upper level hierarchy it can propose for this resource (which currently has over 100 top level classes).

Finally, we plan to compare HGFC to other hierarchical clustering methods that are relatively new to NLP but have proved promising in other fields, including Bayesian Hierarchical Clustering (Heller and Ghahramani, 2005; Teh et al., 2008) and the method of Azran and Ghahramani (2006a) based on spectral clustering.

6 Acknowledgement

Our work was funded by the Royal Society University Research Fellowship (AK), the Dorothy Hodgkin Postgraduate Award (LS), the EPSRC

grants EP/F030061/1 and EP/G051070/1 (UK) and the EU FP7 project 'PANACEA'.

References

Arik Azran and Zoubin Ghahramani. A new approach to data driven clustering. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 57–64, New York, NY, USA, 2006a. ISBN 1-59593-383-2.

Arik Azran and Zoubin Ghahramani. Spectral methods for automatic multiscale data clustering. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pages 190–197. IEEE Computer Society Washington, DC, USA, 2006b.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *In COLING-ACL*, pages 86–90, 1998.

Roberto. Basili, Maria Teresa Pazienza, and Paola Velardi. Hierarchical clustering of verbs. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, 1993.

Nikoletta Bassiou and Constantine Kotropoulos. Long distance bigram models applied to word clustering. *Pattern Recogn.*, 44:145–158, January 2011. ISSN 0031-3203.

Ted Briscoe, John Carroll, and Rebecca Watson. The second release of the rasp system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, 2006.

Hoa Trang Dang. *Investigations into the Role of Lexical Semantics in Word Sense Disambiguation*. PhD thesis, CIS, University of Pennsylvania, 2004.

Katrin Erk, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. Towards a resource for lexical semantics: a large german corpus with extensive semantic annotation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 537–544, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

Eva Esteve Ferrer. Towards a semantic classification of spanish verbs based on subcategorisation information. In *Proceedings of the ACL 2004 workshop on Student research*, ACLstudent '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

Douglas H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987. ISSN 0885-6125.

David Graff. North american news text corpus. *Linguistic Data Consortium*, 1995.

Ralph Grishman, Catherine Macleod, and Adam Meyers. Comlex syntax: Building a computational lexicon. In *COLING*, pages 268–272, 1994.

Katherine A. Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005. ISBN 1595931805.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. ISSN 0176-4268.

Eric Joanis, Suzanne Stevenson, and David James. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(3):337–367, 2008.

Karin Kipper. *VerbNet: A broad-coverage, comprehensive verb lexicon*. 2005.

Anna Korhonen, Yuval Krymolowski, and Nigel Collier. The Choice of Features for Classification of Verbs in Biomedical Texts. In *Proceedings of COLING*, 2008.

Claudia Kunze and Lothar Lemnitzer. GermaNet-representation, visualization, application. In *Proceedings of LREC*, 2002.

Geoffrey Leech. 100 million words of english: the british national corpus. *Language Research*, 28(1):1–13, 1992.

Beth. Levin. English verb classes and alternations: A preliminary investigation. *Chicago, IL*, 1993.

Jianguo Li and Chris Brew. Which Are the Best Features for Automatic Verb Classification. In *Proceedings of ACL*, 2008.

Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceeding of the 17th international conference on World Wide Web*, pages 685–694, New York, NY, USA, 2008. ACM.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Yutaka Matsuo, Takeshi Sakaki, Kôki Uchiyama, and Mitsuru Ishizuka. Graph-based word clustering using a web search engine. In *Proceedings of the EMNLP*, pages 542–550, 2006.

George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Travis E. Oliphant. Python for scientific computing. *Computing in Science and Engineering*, 9:10–20, 2007. ISSN 1521-9615.

Diarmuid Ó Séaghdha and Ann Copestake. Semantic classification with distributional kernels. In *Proceedings of COLING*, 2008.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

Judita Preiss, Ted Briscoe, and Anna Korhonen. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proceedings of ACL*, pages 912–919, 2007.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

Sabine Schulte im Walde. Experiments on the automatic induction of german semantic verb classes. *Computational Linguistics*, 32(2), 2006.

Sabine Schulte im Walde. Human associations and the choice of features for semantic verb classification. *Research on Language and Computation*, 6:79–111, 2008. ISSN 1570-7075.

Sabine Schulte im Walde and Chris Brew. Inducing german semantic verb classes from purely syntactic subcategorisation information. In *Proceedings of ACL*, pages 223–230, 2002.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Lei Shi and Rada Mihalcea. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In *Proceedings of CICLING*, 2005.

Suzanne Stevenson and Eric Joanis. Semi-supervised verb class discovery using noisy features. In *Proceedings of HLT-NAACL 2003*, pages 71–78, 2003.

Lin Sun and Anna Korhonen. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the EMNLP 2009*, 2009.

Lin Sun, Anna Korhonen, and Yuval Krymolowski. Verb class discovery from rich syntactic data. *Lecture Notes in Computer Science*, 4919:16, 2008.

Robert Swier and Suzanne Stevenson. Unsupervised semantic role labelling. In *Proceedings of EMNLP*, pages 95–102, 2004.

Yee Whye Teh, Hal Daumé III, and Daniel Roy. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

Akira Ushioda. Hierarchical clustering of words. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1159–1162. Association for Computational Linguistics, 1996.

Gloria Vázquez, Ana Fernández-Montraveta, and M. Antònia Martí. *Clasificación verbal:(alternancias de diátesis)*. Universitat de Lleida, 2000. ISBN 8484090671.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1.

Andreas Vlachos, Anna Korhonen, and Zoubin Ghahramani. Unsupervised and constrained dirichlet process mixture models for verb clustering. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 74–82, 2009.

Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. ISSN 0162-1459.

Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, pages 1101–1113, 1993. ISSN 0162-8828.

Kai Yu, Shipeng Yu, and Volker Tresp. Soft clustering on graphs. *Advances in Neural Information Processing Systems*, 18:1553, 2006.

Beñat Zapirain, Eneko Agirre, and Lluís Màrquez. Robustness and generalization of role sets: PropBank vs. VerbNet. In *Proceedings of ACL-08: HLT*, pages 550–558, 2008.