MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction

David Hope and Bill Keller

University of Sussex
Cognitive and Language Processing Systems Group
Brighton, Sussex, UK
davehope@gmail.com, billk@sussex.ac.uk

Abstract. This paper introduces a linear time graph-based soft clustering algorithm. The algorithm applies a simple idea: given a graph, vertex pairs are assigned to the same cluster if either vertex has maximal affinity to the other. Clusters of varying size, shape, and density are found automatically making the algorithm suited to tasks such Word Sense Induction (WSI), where the number of classes is unknown and where class distributions may be skewed. The algorithm is applied to two WSI tasks, obtaining results comparable with those of systems adopting existing, state-of-the-art methods.

1 Introduction

A Natural Language Processing (NLP) task may require a set of words to be grouped or clustered into subsets, where each subset represents a distinct lexicological class. For example, Word Sense Induction (WSI), the task of automatically determining word senses from text, is approached in this paper by clustering words associated with a polysemous target word into subsets of semantically related words. The words in each subset are then taken to define a different sense of the target word. For example, if *orange* is a target word associated with the set of words {*red*, *apple*, *yellow*, *banana*, *green*, *pear*}, assignment of the words in this set to two subsets {{*red*, *yellow*, *green*},{*apple*, *banana*, *pear*}} defines two senses of *orange*: the first representing the colour sense of *orange*, the second, its fruit sense. In practice of course, the use of very large corpora in NLP means that there is a need to cluster much larger sets of words than illustrated here. A computationally efficient clustering process is therefore needed. In addition, as words associated with a target word may themselves be polysemous, clustering should also be able to assign words to two or more senses of the target word.

A WSI system such as that outlined above has the potential to alleviate the lexicographer's task of manually identifying, collating, and exemplifying word senses: an enormous undertaking, given both the number of existing senses and the rate at which new senses are introduced into language. In principle, WSI

A. Gelbukh (Ed.): CICLing 2013, Part I, LNCS 7816, pp. 368–381, 2013.

avoids reliance on a pre-defined sense inventory[1], as required in Word Sense Disambiguation (WSD). WSD systems assign pre-defined senses to words on the basis of context. In contrast, WSI systems follow the dictum that "*The meaning of a word is its use in the language.*" [3] to discover senses through examination of context of use in large text corpora. As a consequence, rare, fine-grained and domain specific senses not defined in existing inventories can be induced [4].

2 A Graph-Based Approach to Word Sense Induction

WSI is approached in this paper using a graph-based model of word co occurrence. A graph $G = (V, E)$ consists of a set of vertices V and a set of edges $E \subseteq V \times V$. In the present approach, each vertex $v \in V$ represents a word. An edge $(u, v) \in E$ is a pair of vertices. An edge represents a symmetrical relationship between words u and v; here, that u and v co-occur in the contexts of a target word. An edge-weighted graph assigns to each edge a weight $w(u, v)$. In the present work edge-weights can be understood as quantifying the strength or significance of word co-occurrence relationships.

Figure 1 shows an edge-weighted graph G_{orange} in which the target word vertex is *orange* and the set of words associated with *orange* are represented as adjacent vertices (i.e. words found to co-occur in *orange*'s contexts of use).

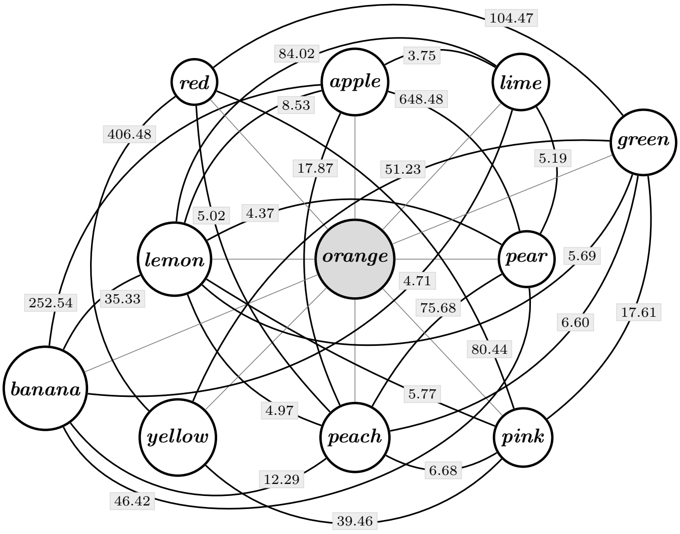

Fig. 1. G_{orange}

[1] In practice, evaluation of a WSI approach requires the use of a gold standard sense inventory, for example WordNet [1] or OntoNotes [2].

In principle, 'contexts of use' might be interpreted as sentences, paragraphs, or context windows containing the target word. In the work described, context words are nouns occurring in co-ordination patterns [5][2]. Edge weights are scores provided by the Log Likelihood Ratio (LLR) [6], a measure of how significant it is that two words u and v co-occur [7].

Senses may be induced by applying a clustering algorithm to identify subgraphs of G_{orange}, as illustrated in Fig. 2. Each subgraph (cluster) can then be assigned a sense of the target word, either by mapping the cluster to a sense given in an inventory (e.g. WordNet [8, 9]) or to a gold standard class [10–12].

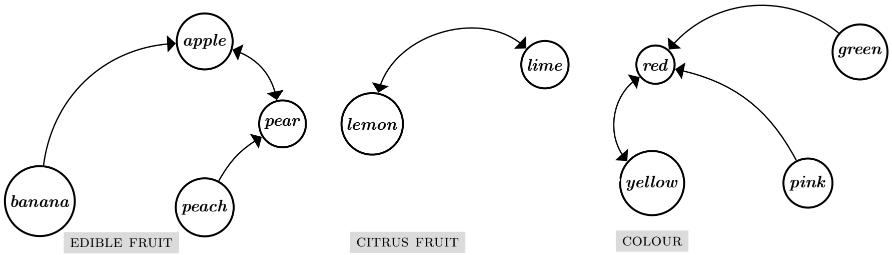

Fig. 2. $C_{G_{orange}}$, a clustering solution for the graph G_{orange} shown in Fig. 1. Sense labels are obtained using the cluster to sense mapping algorithm proposed in [8]

The word co-occurrence model outlined here is similar to models previously applied in WSI, notably to those presented in [5, 10]. A key difference however is the use of a novel clustering algorithm, MaxMax.

3 MaxMax

MaxMax is a non-parameterised, soft-clustering algorithm applicable to edge-weighted graphs. A notion of *maximal affinity* is used, where affinity between vertex pairs u and v is quantified by edge weights $w(u, v)$. A vertex u is said to have maximal affinity to a vertex v if the edge weight $w(u, v)$ is maximal amongst the weights of all edges incident on u. In this case, v is said to be a *maximal vertex* of u (v need not be unique). Two principles are applied: 1) vertex pairs u,v are assigned to the same cluster if either vertex is a maximal vertex of the other; and 2) maximal affinity implies a *directed* relationship: if v is a maximal vertex of u then there is a directed relationship from v to u.

[2] Nouns are extracted from the British National Corpus (BNC) using the regular expression `NP(, NP)*,?( CJC NP)+` where `CJC` = `(and|or|nor)` and `NP`, a noun phrase, = `AT?( CRD)*( ADJ)*( NOUN)+`.

Algorithm 1. MaxMax

```
1: procedure MaxMax($G = (V, E)$)
2:     construct a directed graph $G' = (V, E')$ where:
3:         $(v, u) \in E'$ iff $(u, v) \in E$ and $v$ is a maximal vertex for $u$
4:     mark all vertices of $G'$ initially as root
5:     for each vertex $v$ of $G'$ do
6:         if $v$ is marked root then
7:             mark any descendant $u$ of $v$ $(u \neq v)$ as $\neg$root
8:         end if
9:     end for
10: end procedure
```

MaxMax consists of two discrete stages:

Stage 1. Graph Transformation. In stage 1 (lines 2 and 3 of Algorithm 1) MaxMax takes a weighted graph G and transforms it to an unweighted, directed graph (*digraph*) G'. The maximal affinity relationships between vertices of G are used to determine the direction of the edges in G'. An example of the way in which a weighted undirected graph is transformed to an unweighted, directed graph is shown in Fig. 3.

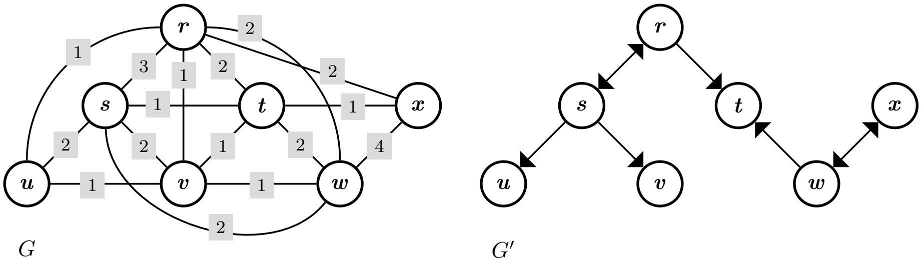

Fig. 3. G and its transformation to an unweighted directed graph G'

Stage 2. Identifying Clusters. In a digraph G', a vertex v is said to be a *descendant* of a vertex u if there is a directed path from u to v. For example, in Fig. 3 vertex v is a descendant of vertices s and r. In stage 2, clusters are found by tracing directed paths in G' to identify rooted subgraphs of a particular type (lines 4 to 9 of Algorithm 1). The vertices of each subgraph define a distinct cluster. This is made precise as follows.

A directed graph is said to be *quasi-strongly connected* (QSC) if for any vertices v_i and v_j, there is a vertex v_k (not necessarily distinct from v_i and v_j) such that there is a directed path from v_k to v_i and a directed path from v_k to v_j.

It is not hard to show that a QSC digraph must contain at least one vertex v_r which is a *root* in the sense that every other vertex can be reached by following a directed path from v_r. Given a directed graph G', a subgraph of G' is a *maximal* QSC subgraph if it is a QSC digraph and it is not possible to add any further vertices or edges from G' without rendering the subgraph non-QSC.

Clusters are identified by finding the *root* vertices of maximal QSC subgraphs of G'. This is achieved simply by marking all descendants of a given vertex as $\neg root$. For example, consider vertex s in the directed graph G' of Fig 4, which is initially marked as a *root*. The descendant vertices of s are u and v thus marked as $\neg root$. In turn, s, as a descendant of r, is marked $\neg root$[3]. At the end of stage two, vertices that are still marked as *root* vertices uniquely identify clusters, since they correspond to the roots of maximal QSC subgraphs of G'.

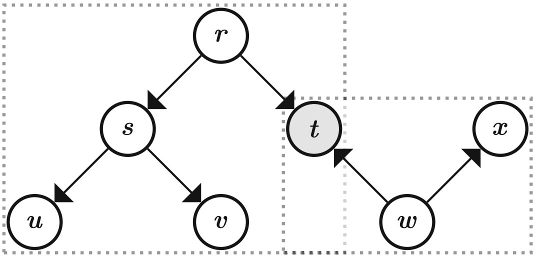

Fig. 4. Two clusters in G'

As Fig. 4 shows, this process allows vertices to be soft clustered to more than one cluster. In this particular example, vertex t is soft clustered to cluster $\{r, s, t, u, v\}$ and cluster $\{w, t, x\}$.

3.1 Time Complexity

It can be shown that for a connected graph $G = (V, E)$, MaxMax runs in time $O(|E|)$, that is, linear in the number of edges of G. The transformation of an edge weighted graph G to an unweighted directed graph G' in the first stage can be computed in $O(|E|)$. In constructing G' it is necessary to find maximal vertices of each vertex in G. For a given vertex u, the set of maximal vertices can be identified by scanning each of the edges from u to a vertex adjacent to u in order to determine those of maximal weight. This is done for each vertex of G, with each edge in G inspected just once[4]. Consequently, G' can be constructed in time linear in the number of edges of G.

[3] In Fig. 3 vertex r or vertex s is a permissible *root* of the cluster $\{r, s, t, u, v\}$; similarly, either x or w may be the *root* of the cluster $\{t, w, x\}$.

[4] Connections u to v and v to u are considered to be two separate edges in undirected graphs [13].

In the second stage, vertices are initially marked as *root* (line 4 of Algorithm 1), taking $O(|V|)$ time. The for loop (lines 5 to 9) iterates over vertices to identify descendant vertices (line 7) that should be marked $\neg root$. Naively tracing all of the descendants of each vertex in turn could in the worst case entail visiting $O(|V|)$ vertices on each pass through the loop and so, result in an overall complexity of $O(|V|^2)$. However, it is easy to show that once a vertex has been marked $\neg root$ then none of its descendants needs to be visited again. Equivalently, no directed edge needs to be traversed more than once thus, overall complexity of the for loop is linear in the number of edges of G' (and hence linear in the number of edges of G). This yields an overall time complexity of $O(|E|)$.

4 WSI Tasks

In this section MaxMax is applied to two different WSI tasks. Results show that approaches based on MaxMax are comparable with those of state-of-the-art WSI systems.

4.1 Task 1: The SemEval 2010 WSI Task

The SemEval 2010 Word Sense Induction and Disambiguation task [12] provides a formal evaluation framework, enabling participants to compare systems. Systems may be evaluated on a supervised WSD task or alternatively on an unsupervised WSI task. The WSI task is considered here.

Participants are required to induce the senses of 100 target words: 50 verbs and 50 nouns. The test set consists of 8,915 instances (sentences or paragraphs) containing a target word: 5,285 for nouns; 3,630 for verbs. Instances are tagged with OntoNotes senses [2]. Participants are required to tag each instance in the test set with a sense of the target word, the sense being derived by the participant's WSI system.

The SNN$_{\text{swf}}$ System. The WSI system applied in this evaluation, SNN$_{\text{swf}}$, adapts the Shared Nearest Neighbours (SNN) algorithm [14] to fit the task. SNN$_{\text{swf}}$ first extracts unordered and ordered pairs of words from test instances. For example, given an instance $[w_1\ tw\ w_2\ w_3]$, where tw represents the target word and w_1, w_2, w_3 represent context words, the following information is extracted -

1. unordered pairs: $\{w_1,\ w_2\}$, $\{w_1,\ w_3\}$, $\{w_2,\ w_3\}$.
2. ordered pairs: $(w_1,\ w_2)$, $(w_1,\ w_3)$, $(w_2,\ w_3)$.

Each context word in a target word instance is associated with a set of *word features*. Thus, for the instance above, w_2 is associated with the word features

w_1 and w_3 and the relative word order features (extracted from the ordered pairs) $w_{1_{Left2}}, w_{3_{Right1}}$. The rationale for the inclusion of the word order features is that these may function as proxies for dependency relations between context words [15].

Word features are filtered using the Log Likelihood Ratio (LLR) measure [6]. In this evaluation, a LLR threshold is set at 10.83[5]. Thus, if the LLR score between w_2 and w_3 is greater than or equal to 10.83, then w_3 is taken to be a *significant word feature* of w_2. The threshold filters out features shared by many words. Features passing the LLR threshold should provide strong indicators for the senses of a target word.

Similarity between instance pairs is then calculated as the number of shared significant word features (for all context words in both instances). This approach allows both first order and second order similarity to be computed. Thus, even if two instances have no words in common, the words themselves may share many features, indicating a degree of semantic relatedness between two instance pairs.

A target word graph is constructed using similarity between instance (vertex) pairs as edge weights. MaxMax is then applied to identify a set of sense clusters. A perfect clustering solution would thus assign test instances of each sense of the target word to a separate cluster.

Evaluation Measures. Two evaluation measures are used to assess system performance: the V-Measure [16] and the Paired F-Score [17]. Both measures purport to reflect alignment between a hypothesis K, the clusters returned by a system, and a reference C, the set of gold standard classes in the test set. The V-Measure is defined as the harmonic mean of homogeneity and completeness, where homogeneity is the degree to which each cluster in K consists of instances belonging to a single gold standard class in C and completeness is the degree to which each cluster in K consists of all instances of a single gold standard class in C. The Paired F-Score pairs instances in gold standard classes C and instances in clusters K, then measures the extent to which pairs in C and K overlap.

Results. Tables 1 and 2 report results for $\mathrm{SNN}_{\mathrm{swf}}$ along with the best, worst, and average score returned by participating systems. The baselines, provided by the organisers of the task, are: *1CPI*, one cluster per instance; *MFS*, most frequent senses (all instances in one cluster), and *Random*, which randomly assigns instances to one of four clusters. Table 1 shows that $\mathrm{SNN}_{\mathrm{swf}}$ is the best performing system, by some margin, according to the V-measure. However, Table 2 shows that $\mathrm{SNN}_{\mathrm{swf}}$ is the worst performing system if the Paired F-Score is applied.

[5] Lower LLR thresholds were applied (3.84, 6.63, 7.9), returning worse results. Given the number of possible word, feature pairs in the test set, a threshold set higher than 10.83 would be statistically invalid.

Table 1. V-Measure results

System	Verbs	Nouns	|Clusters|
1CPI	25.6	35.8	89.15
$\mathbf{SNN_{swf}}$	**24.6**	**32.8**	32.31
Hermit	**15.6**	16.7	10.78
UoY	8.5	**20.6**	11.54
Average	6.37	7.73	4.07
Random	4.6	4.2	4.00
Duluth-WSI-SVD-Gap	0.1	0.0	1.02
MFS	0.0	0.0	1.00

Table 2. Paired F-Score results

System	Verbs	Nouns	|Clusters|
MFS	72.7	57.0	1.00
Duluth-WSI-SVD-Gap	**72.4**	**57.0**	1.02
Average	52.3	42.8	4.07
Random	34.1	30.4	4.00
Hermit	30.1	24.4	10.78
$\mathbf{SNN_{swf}}$	14.4	13.2	32.31
1CPI	0.08	0.11	89.15

It can be observed that Paired F-Score is biased towards clustering solutions returning large clusters: each instance in a cluster of size n pairs with $n-1$ other instances, and so punishes misclassification in small clusters disproportionately [18, 19]. This penalises the MaxMax system, which tends to generate relatively high numbers of fine-grained senses. Such senses may very well have high standard Precision and Recall [19]. The V-Measure on the other hand favours clustering solutions returning numerous small clusters [18, 20, 19, 21]. In this case the bias is due to the normalisation applied in the completeness term of the measure which monotonically increases with the number of induced clusters [20].

4.2 Task 2: Inducing WordNet Senses

The aim of this task is to induce the senses, as defined in WordNet 3.0, of the 27,071 nouns found in co-ordination patterns extracted from the British National Corpus (BNC). The evaluation methodology follows that in [5], which reports results comparable with those reported in [9] – the best results reported to date for this task. In [5], an unweighted graph G_{tw} is constructed, where each vertex represents a noun found in co-ordination patterns and each edge represents noun co-occurrence. The vertex cohesion measure of *curvature* [22, 23] is applied to partition the graph into a set of clusters. The graph theoretical concept of *percolation* [24, 25] is used to find a suitable curvature threshold to apply. Nouns in each cluster are taken to represent a candidate WordNet sense.

A problem observed with this approach is that many semantically unrelated words may be assigned to the same cluster [19]. Consequently, an alternative approach is adopted here. The weighted graph G_{tw} is first transformed to a weighted graph G^{T}_{tw}. G_{tw} consists of a target word (a noun in co-ordination patterns) and its adjacent neighbours. Edge weights $w(u, v)$ between vertices u and v are values returned by an association measure for two nouns co-occurring in patterns. G^{T}_{tw} is derived by deleting edges in G_{tw} with edge weight $\leq$ a predefined threshold. MaxMax is then applied to G^{T}_{tw}, returning a set of candidate sense clusters for the target word.

Candidate sense clusters are mapped to WordNet senses using the method proposed in [9]. This method returns a similarity score between a cluster and the sense of the target word the cluster maximises. If the similarity score exceeds a predefined threshold, the cluster is taken to be a sense of the target word. Validity of cluster to sense mappings is measured using Precision, Recall and F-Score [26]. Precision for a target word *tw* is defined as:

$$Precision(tw) = \frac{|\{c_i \in C_{tw} \mid \exists\, s_j \in S_{tw} : similarity(c_i, s_j) \geq \sigma\}|}{|C_{tw}|}, \qquad (1)$$

and Recall as:

$$Recall(tw) = \frac{|\{s_i \in S_{tw} \mid \exists\, c_j \in C_{tw} : similarity(c_j, s_i) \geq \sigma\}|}{|S_{tw}|}. \qquad (2)$$

In (1) and (2) C_{tw} denotes the set of clusters returned by MaxMax given G^{T}_{tw}, and S_{tw} is the set of WordNet senses of *tw*. σ is the cluster-sense similarity threshold applied[6].

Precision is defined in (1) as a *many to one* mapping; that is, many clusters may map to a single sense of the target word *tw*[7]. Arguably, this is a fairer measure for evaluating WSI approaches than that of standard Precision [26] as a sense of a target word may be distributed across a number of clusters. Note that each cluster mapped to a sense of *tw* must pass the similarity threshold σ thus, each cluster counted in the numerator of (1) is, according to the definition given in [9], a valid sense of the target word.

Results. Table 3 reports results, where |Words| is the number of words that can be evaluated by a particular measure, *LLR* is the Log Likelihood Ratio [6] threshold used to transform G_{tw} to G^{T}_{tw}, and *Counts* is a word co-occurrence model using raw co-occurrence counts as edge weights between noun pairs in coordination patterns (counts > 1). Results show that the graph transformation approach outperforms the curvature approach. Coverage of words is also far higher. It is interesting to note that the best results are returned by a simple graph model of word co-occurrence (*Counts*). Coverage here, at 9101 words, is

[6] σ is set to 0.25, the threshold applied in [9] and [5].

[7] Defined as accuracy in [14].

Table 3. Results for the curvature and graph transformation approaches

System	Precision	Recall	F-Score	|Words|
$MaxMax_{LLR=15.13}$	72.1	53.2	61.2	11138
$MaxMax_{LLR=10.83}$	65.5	53.1	58.7	16850
$MaxMax_{LLR=6.63}$	59.6	55.8	57.6	21716
$MaxMax_{LLR=3.84}$	56.6	**59.1**	57.8	22899
$MaxMax_{Counts}$	**74.2**	58.6	**65.5**	9101
Curvature	60.9	40.7	48.8	3906

relatively low yet, is still over twice the number of words that can be evaluated using the curvature approach. This result suggests that comparatively complex measures of word association may not be required to induce word senses.

5 Discussion

Graph-based models (GBM) have been previously applied to WSI, using words as vertices in [27, 28, 10, 29–31] and word collocations as vertices in [32–35]. Alternative approaches include the use of a vector space model (VSM) in [36, 37, 9], Latent Semantic Analysis (LSA) in [38, 39], and a Bayesian approach in [40]. Surveys of WSI approaches are provided in [41–43].

As noted in [28, 44], a VSM approach using context words can conflate senses, as each vector merges the senses context words take. A GBM clearly delineates the uses of context words. LSA, a dimensionality reduction technique, aims to remove information that is irrelevant to the problem space however, this can lead to information pertinent to finding rarer senses being discarded. In contrast, a GBM retains all information. A GBM using collocations as vertices is based on Yarowsky's tenet: 'one sense per collocation' [45]. The argument given [32–35] is that collocation vertices are less sense conflating than single word vertices. Arguably though, collocation vertices are not required if the set of word vertices that define target word senses is filtered using a significance threshold set on edge weights.

Existing soft clustering algorithms such as Fuzzy c-Means [46] and Expectation Maximization [47] require the number of clusters to be pre-defined. Parameter tuning is therefore necessary in order to find a good clustering solution [14]. Consequently, these algorithms are not well-suited to WSI, as the number of senses target words take is often undefined.

MaxMax bears some resemblance to single-link Hierarchical Agglomerative Clustering (HAC) [14] in that leaf vertices (clusters consisting of one vertex) in the first iteration of HAC are clustered using maximal affinity between vertex pairs. However, whereas HAC hard clusters vertices in $O(|V|^3)$, MaxMax hard/soft clusters vertices in linear time. MaxMax also shares some properties with Chinese Whispers [48], a non-parameterised, graph-based clustering algorithm shown to have utility in NLP [4, 49–52, 30]. Both algorithms use affinity within the local neighbourhood of vertices to generate clusters and both have linear run times. However, there are key differences. MaxMax is deterministic, whilst Chinese Whispers may return different solutions for the same graph. In addition, MaxMax is able to soft cluster vertices, whilst Chinese Whispers cannot. Thus, given the input graph G in Fig. 5, Chinese Whispers randomly assigns vertex c to cluster 1 or 2 whereas MaxMax returns the clustering solution C_G.

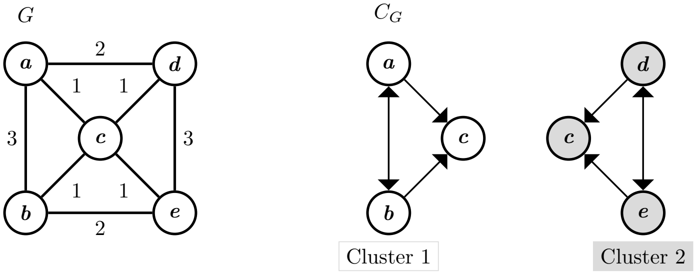

Fig. 5. Soft clustering example

6 Conclusions

This paper introduced MaxMax, a novel non-parameterised soft clustering algorithm that finds the number of clusters in a graph automatically by identifying root vertices of maximal quasi strongly connected subgraphs, a process shown to be computable in linear time. Examples showed that descendant vertices of more than one root vertex can be soft clustered, with a descendant vertex assigned to each cluster containing a vertex to which it has maximal affinity: a straightforward process that, in comparison to existing soft clustering algorithms, is both fast and transparent. As a non-parameterised clustering algorithm, MaxMax is well-suited to WSI or, indeed, to any task in which the number of clusters is not known in advance. To test its utility for WSI, MaxMax was incorporated into two induction systems. Results in two tasks showed the systems to return scores comparable with, if not better than, those of existing state-of-the-art systems. However, further tests are required thus, future research plans to apply the algorithm in the forthcoming SemEval 2013 WSI evaluations and to carry out a comparative analysis against the recently introduced SquaT++ and B-MST clustering algorithms [30]. Additionally, as the WSI tasks in this paper have no special requirement for soft clustering, future research also plans to apply the algorithm to networks in which soft clustering may be of use. For example, social networks [53] and contagion networks [54] typically have many vertices with ties to more than one maximal vertex therefore, MaxMax may be particularly suited to studying these types of networks.

Acknowledgements. The authors wish to thank the anonymous reviewers for their feedback on a previous version of this paper.

References

1. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography 3, 235 (1990)
2. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: the 90% Solution. In: Proceedings of the Human Language Technology Conference of the NAACL, pp. 57–60. Association for Computational Linguistics (2006)
3. Wittgenstein, L.: Philosophical Investigations. Blackwell (1953)
4. Klapaftis, I., Manandhar, S.: Word Sense Induction Using Graphs of Collocations. In: Proceeding of the 2008 Conference on ECAI, pp. 298–302 (2008)
5. Dorow, B.: A Graph Model for Words and their Meanings. PhD thesis, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart (2007)
6. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19, 61–74 (1993)
7. Evert, S.: The Statistics of Word Cooccurrences: Word Pairs and Collocations. PhD thesis, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart (2005)
8. Widdows, D.: Geometry and Meaning. CSLI Lecture Notes. CSLI Publications, Center for the Study of Language and Information (2004)
9. Pantel, P., Lin, D.: Discovering Word Senses from Text. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 613–619. ACM (2002)
10. Biemann, C.: Unsupervised and Knowledge-Free Natural Language Processing in the Structure Discovery Paradigm. PhD thesis, University of Leipzig (2007)
11. Agirre, E., Soroa, A.: SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In: Proceedings of the 4th International Workshop on Semantic Evaluations, pp. 7–12. Association for Computational Linguistics (2007)
12. Manandhar, S., Klapaftis, I., Dligach, D., Pradhan, S.: SemEval-2010 Task 14: Word Sense Induction and Disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 63–68. Association for Computational Linguistics (2010)
13. Dasgupta, S., Papadimitriou, C., Vazirani, U.: Algorithms. McGraw-Hill (2006)
14. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison Wesley (2006)
15. Yuret, D.: Discovery of Linguistic Relations Using Lexical Attraction. PhD thesis, Massachusetts Institute of Technology (1998)
16. Rosenberg, A., Hirschberg, J.: 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, pp. 410–420 (2007)
17. Artiles, J., Amigó, E., Gonzalo, J.: The Role of Named Entities in Web People Search. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 2, pp. 534–542 (2009)
18. Pedersen, T.: Duluth-WSI: SenseClusters Applied to the Sense Induction Task of SemEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 363–366. Association for Computational Linguistics (2010)

19. Hope, D.: Graph-Based Approaches to Word Sense Induction. PhD thesis, University of Sussex (2012) (unpublished)
20. Reichart, R., Rappoport, A.: The NVI Clustering Evaluation Measure. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, pp. 165–173. Association for Computational Linguistics (2009)
21. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. Information Retrieval 12, 461–486 (2009)
22. Watts, D., Strogatz, S.: Collective Dynamics of 'Small-World' Networks. Nature 393, 440–442 (1998)
23. Eckmann, J., Moses, E.: Curvature of Co-Links Uncovers Hidden Thematic Layers in the World Wide Web. Proceedings of the National Academy of Sciences 99, 5825 (2002)
24. Erdős, P., Rényi, A.: On the Evolution of Random Graphs. Akad. Kiadó (1960)
25. Bollobás, B., Riordan, O.: Percolation. Cambridge University Press (2006)
26. Van Rijsbergen, C.: Information Retrieval. Butterworths (1979)
27. Dorow, B., Widdows, D.: Discovering Corpus-Specific Word Senses. In: Proceedings of the Tenth Conference, European Chapter of the Association for Computational Linguistics, vol. 2, pp. 79–82. Association for Computational Linguistics (2003)
28. Véronis, J.: Hyperlex: Lexical Cartography for Information Retrieval. Computer Speech & Language 18, 223–252 (2004)
29. Agirre, E., Martínez, D., de Lacalle, O., Soroa, A.: Two Graph-Based Algorithms for State-of-the-Art WSD. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 585–593. Association for Computational Linguistics (2006)
30. Di Marco, A., Navigli, R.: Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. Computational Linguistics 39(4) (2013)
31. Navigli, R., Crisafulli, G.: Inducing Word Senses to Improve Web Search Result Clustering. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 116–126. Association for Computational Linguistics (2010)
32. Dorow, B., Widdows, D., Ling, K., Eckmann, J., Sergi, D., Moses, E.: Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. In: 2nd Workshop Organized by the MEANING Project (2005)
33. Klapaftis, I., Manandhar, S.: Word Sense Induction Using Graphs of Collocations. In: Proceeding of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence, pp. 298–302. IOS Press (2008)
34. Klapaftis, I., Manandhar, S.: Word Sense Induction and Disambiguation Using Hierarchical Random Graphs. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 745–755. Association for Computational Linguistics (2010)
35. Bordag, S.: Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In: Proceedings of EACL 2006, Trento (2006)
36. Schütze, H.: Automatic Word Sense Discrimination. Computational Linguistics 24, 97–123 (1998)
37. Purandare, A., Pedersen, T.: Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. In: Proceedings of the Conference on Computational Natural Language Learning, pp. 41–48 (2004)

38. Van de Cruys, T.: Using Three Way Data for Word Sense Discrimination. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 929–936 (2008)
39. Van de Cruys, T., Apidianaki, M.: Latent Semantic Word Sense Induction and Disambiguation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT), pp. 1476–1485 (2011)
40. Brody, S., Lapata, M.: Bayesian Word Sense Induction. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 103–111. Association for Computational Linguistics (2009)
41. Navigli, R.: Word Sense Disambiguation: A Survey. ACM Computing Surveys (CSUR) 41, 10 (2009)
42. Apidianaki, M., Van de Cruys, T.: A Quantitative Evaluation of Global Word Sense Induction. In: Gelbukh, A.F. (ed.) CICLing 2011, Part I. LNCS, vol. 6608, pp. 253–264. Springer, Heidelberg (2011)
43. Navigli, R.: A Quick Tour of Word Sense Disambiguation, Induction and Related Approaches. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 115–129. Springer, Heidelberg (2012)
44. Klapaftis, I.: Unsupervised Concept Hierarchy Induction: Learning the Semantics of Words. PhD thesis, University of York (2008)
45. Yarowsky, D.: Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, pp. 189–196. Association for Computational Linguistics (1995)
46. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers (1981)
47. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B (Methodological), 1–38 (1977)
48. Biemann, C.: Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In: Proceedings of the HLT-NAACL 2006 Workshop on Textgraphs 2006 (2006)
49. Korkontzelos, I., Manandhar, S.: Detecting Compositionality in Multi-Word Expressions. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 65–68. Association for Computational Linguistics (2009)
50. Zhang, Z., Sun, L.: Improving Word Sense Induction by Exploiting Semantic Relevance. In: Proceedings of the 5th International Joint Conference on Natural Language Processing, pp. 1387–1391 (2011)
51. Jurgens, D.: An Evaluation of Graded Sense Disambiguation Using Word Sense Induction. In: Proceedings of *SEM First Joint Conference on Lexical and Computational Semantics. ACL (2012)
52. Fountain, T., Lapata, M.: Taxonomy Induction Using Hierarchical Random Graphs. In: 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 446–476 (2012)
53. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)
54. Newman, M., Barabási, A.L., Watts, D.J.: The Structure and Dynamics of Networks. Princeton University Press (2006)