ABSINTH: A Small World of Semantic Similarity

Maja Hoffmann **Victor Zimmermann**
Heidelberg University, Department of Computational Linguistics
{hoff|zimmer}mann@cl.uni-heidelberg.de

Abstract

ABSINTH[1] provides a novel graph based approach to word sense induction for Task 11 of SemEval-2013, combining work from multiple fields of natural language processing, most notably Hyperlex (Véronis, 2004) and sentiment propagation (Hamilton et al., 2016).

1 Introduction

As late as twelve years after publication, the graph based approach to word sense induction proposed in Véronis (2004) is still cited as 'state-of-the-art' (Tripodi and Pelillo 2016, Ustalov et al. 2017). We build on the principles laid out in Hyperlex (Véronis, 2004) with a more dynamic feature set, as well as recent methods previously used mostly for sentiment analysis and tasks unrelated to natural language processing.

Our system provides a simple yet efficient two-step solution to SemEval-2013 Task 11 (Navigli and Vannella, 2013). To achieve this, we utilise the properties of small world graphs for language (Cancho and Solé, 2001) in general and semantic relations (Newman, 2003) in particular. We extract senses using the root hub algorithm proposed in Véronis (2004) with adjusted, flexible features for corpora of varying sizes.

For word sense disambiguation we use the sense inventory created in previous steps and a graph propagation algorithm to assign each node a sense distribution vector. Lastly, the vectors of each word in a given context are summed up and the context is assigned the sense of the best cumulative weight.

In addition to the SemEval scoring methods to evaluate our results we use Characteristic Path Length and Global Clustering Coefficient to evaluate the properties of our cooccurrence graphs.

Our system results lie within the expected performance set by the original task participants.

[1] Association Based Semantic Induction Tools from Heidelberg

PARAMETER	OUR SYSTEM	HYPERLEX	BASELINE
MIN. CONTEXT	4	4	4
MIN. #NODES	AVG. #NODES	10	9
MIN. #EDGES	AVG. #EDGES	5	3
MAX. WEIGHT	0.9	0.9	0.9

Table 1: Minimum context size, minimum number of nodes, minimum number of edges and maximum edge weight for our system, Hyperlex and our Baseline.

2 Related Work

Graph based approaches to word sense induction have been successfully used since the early 2000s (Véronis 2004, Di Marco and Navigli 2013). Véronis proposes the use of root hub detection and minimum spanning trees (Kruskal, 1956) to induce senses and disambiguate search results.

The usefulness of small world graph properties for sense disambiguation has previously been shown in Newman (2003). The term 'small world' was introduced by Travers and Milgram, using it to describe the connectedness of acquaintance networks (Travers and Milgram, 1969). According to their findings, the average path length between two people living in the United States lies around five or six, even though they are selected from a relatively large number of people. The properties of these small world graphs have been formally described in Watts and Strogatz (1998), we show that the our graphs are indeed small world graphs with the words connected in a similar way to real world relations between people.

Because of this property, nodes with a high degree (number of outgoing edges) can be selected as so called 'root hubs'. It is assumed, that words belonging to a sense are clustered around these root hubs and meaning can be induced by mapping a vocabulary to them.

Véronis uses paragraphs including the target string from a web corpus as contexts for building cooccurrence graphs, with two words occurring within a context being an edge. Paragraphs with fewer than 4 words are discarded, further limits on nodes, edges and their weights are introduced (See table 1). The target string is not included in the

graph.
Higher associated edges are assigned lower weights using a weighting system described in (Véronis, 2004). Why this weighting algorithm is chosen over a more traditional measure like Dice weights is not further explained, but we expect an algorithm using Dice weights would artificially limit the number of possible neighbours for each node and therefore reduce the number of possible root hubs significantly.
Root hubs are chosen from the set of graph nodes, limited by the following criteria:

1. the number of neighbours, excluding root hubs and neighbours of root hubs,
2. the mean weight of the candidate's most frequent neighbours, excluding root hubs and neighbours of root hubs.

Additionally, the candidate may not be neighbour to a previously chosen root hub.
Before building the minimum spanning tree, the target string is inserted back into the graph with a distance of 0 to each root hub. This results in the root hubs being selected as the direct children of the target string, allowing the easy mapping of components to a hub.
For disambiguation, Véronis iterates over each node v in the minimum spanning tree and assigns each a weight vector ω:

$$\omega_i = \begin{cases} \frac{1}{1+d(h_i,v)}, & \text{if v belongs to component i,} \\ 0 & \text{else.} \end{cases}$$

with $d(h_i, v)$ being the distance between a root hub h_i and a node v.
For a given context, the weight vectors of each token are added up and the sense with the highest cumulative weight is chosen.
We use Véronis' root hub algorithm broadly with more flexible parameters for our corpus. Our disambiguation system still uses Hyperlex' minimum spanning tree as a backup, but fundamentally builds on labelled graph propagation (Hamilton et al., 2016).

3 Task Set-up

We will be working on Task 11 of the SemEval-2013 Workshop (Navigli and Vannella, 2013). The aim of the task is to develop a Word Sense Induction (WSI) tool, that can be used in Web Search Result Clustering. The data is structured as follows:
Each topic is given by a target string. For every topic there is a list of the first hundred internet search results, containing information for the result, namely the URL, title and a text snippet.

3.1 Corpus

Our system uses an unordered plain-text Wikipedia dump from 2014. As the sense set used in the task hails from Wikipedia, using Wikipedia itself seemed like a natural fit. Because of soft limits on how many nodes and edges our system considers, an ordered corpus may favour one sense over another based on if its article randomly fell into our sample.
Additionally we add the titles and snippets of each query to our corpus, since it offers us a guaranteed baseline of around 500 nodes per sense.

4 Motivation

Our graphs are so called 'small world graphs'. The connection topography of a small world graph, as described in Watts and Strogatz (1998), lies between a completely random and a completely ordered graph. Therefore small world graphs can be highly clustered, but still have relatively short path lengths between the nodes.
The structural properties of these graphs are defined by Characteristic Path Length L(p), which measures the average separation between nodes of a graph and Global Clustering Coefficient C(p), which measures the cliquishness of a typical neighbourhood. The Global Clustering Coefficient ranges between 0 (for a completely disconnected graph) and 1 (for a highly connected graph). Characteristic Path Length and Global Clustering Coefficient are calculated as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} d_{min}(i,j)$$

$$C = \frac{1}{N}\sum_{i=1}^{N} \frac{|E(\Gamma(i))|}{\binom{|\Gamma(i)|}{2}},$$

with node count (N), the shortest distance between two nodes i, j ($d_{min}(i, j)$), degree of a node i ($\Gamma(i)$) and proportion of connection between neighbours $\Gamma(i)$ of a node i ($E(\Gamma(i))$). To determine whether a graph is indeed a small world graph, L(p) and C(p) have to be evaluated against a random connection topography of a graph of the

Target	L_{sys}	C_{sys}	L_{rand}	C_{rand}
COOL_WATER	3.675	.528	6.025	0.030
SOUL_FOOD	4.664	0.604	4.992	0.022
STEPHEN_KING	3.649	0.552	3.791	0.014
THE_BLOCK	3.905	0.329	3.721	0.006
AVERAGE	3.973	0.503	4.632	0.018

Table 2: Characteristic Path Length (L) and Global Clustering Coefficient (C) for our system and a random graph.

same size.
The random measures are calculated as follows:

$$L_{rand} \sim log(N)/log(k)$$
$$C_{rand} \sim 2k/N.$$

A small world graph is defined as follows (Véronis, 2004):

$$L \sim L_{rand}$$
$$C >> C_{rand}.$$

As can be seen in table 2, our graphs resemble small world graphs, as they feature short Average Path Lengths, but significantly higher Clustering Coefficients, compared to what would be expected of random graphs.
Véronis uses these properties mostly for root hub detection. We included a graph propagation system for disambiguation, that utilises these graph properties as well.
Because our corpus is much less balanced than Véronis (2004) and our task is more varied[2], we use a more flexible set of parameters and methods. The task set-up does not support the use of heuristic variables, as some terms are simply too infrequently represented in our corpus to build meaningful graph representations. While setting the euclidean mean of node/edge frequency as a minimum offers a solution to the problem of sparse graphs for less represented terms, more frequent terms seem to over-generate root hubs.
Graph propagation offers a simple method in reducing the total number of senses by essentially merging related root hubs, while retaining the characteristic distribution of senses shown in (Véronis, 2004).

5 System

Every step of our induction system works with the properties of small world graphs in mind. The density of certain nodes makes them ideal root hubs, from which a sense distribution can be propagated somewhat organically. The work flow of our system can be roughly translated into induction and disambiguation. The goal of the first task is to produce sensible root hubs. These can be more varied and numerous than in Véronis (2004), as our system merges and shifts the overlying concepts after initial induction. It is important to view the root hubs in our system less as definitions and more as a list of most influential context words to induce meaning. The system can tell meaning from a root hub, but the root hub itself is not the meaning.

[2] Véronis mostly disambiguates highly polysemous terms and no proper names.

5.1 Word Sense Induction

Induction consists of two steps:

1. Construction and weighting of a cooccurrence graph.

2. Inducing root hubs from this graph.

Our graph is constructed in a straightforward approach, only considering paragraphs including our target string. All nouns and verbs of this subcorpus are counted, with each cooccurrence within a paragraph being an edge. Stop words are filtered, as is the target string itself, after which every paragraph containing less than four relevant tokens is discarded.
Every node or edge whose frequency falls under a certain threshold (See table 1.) is also discarded. Our system uses the average number of occurrences instead of a heuristic measure, as our system is robust enough to deal with over-generation of root hubs and our sub-corpora vary in size too considerably to allow heuristic senses without under-generating root hubs for less frequent targets.
The graph is weight using the following method:

$$\omega_{a,b} = 1 - max[p(A|B), p(B|A)], \quad \texttt{with}$$
$$p(A|B) = f_{A,B}/f_B \quad \texttt{and}$$
$$p(B|A) = f_{A,B}/f_A.$$

This weighting method is preferred to a measure like Sørensen-Dice-Weight, as it allows root hubs to have many outgoing edges, while their neighbours can each have a meaningful relation to the root hub without the edge being discarded. We use the algorithm shown in Véronis (2004) to detect

Parameter	Our System	HyperLex
Min. degree	5	6
Max. mean weight	0.8	0.8

Table 3: Meta parameters for building a cooccurrence graph for the analysed systems.

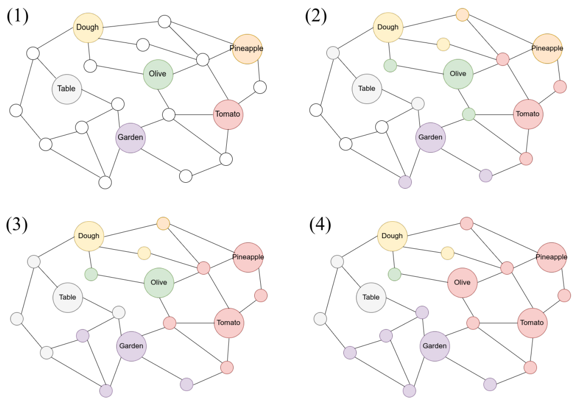

Figure 1: Example of Propagation for the target 'Pizza'.

root hubs, iteratively choosing hubs by their degree and average weight with their most frequent neighbours (See table 3). We then delete the root hub and its neighbours from the graph before selecting the next hub. After no viable candidates are left, the list of root hubs is returned.

5.2 Word Sense Disambiguation

For allocating contexts to senses, our system uses the graph and list of root hubs built in previous steps. Again, disambiguation is a two step process, mirroring the induction process.
First, nodes are labelled according to their 'sense preference' using a propagation algorithm similar to ones used to model voting behaviour (H. Fowler, 2007) or for sentiment analysis (Newman, 2003). The result is a labelled graph with a sense distribution vector for each node. The best sense of the cumulative vector for a given context is chosen for clustering.
Véronis' algorithm using minimum spanning trees[3] is used as a backup for contexts that could not be matched using the propagation algorithm.

5.2.1 Sense Propagation

The goal of our propagation algorithm is to provide an approximation of how indicative a node is for a sense from the root hub inventory. Given that our system adheres to the principle that the sense of a word is defined by its neighbours, it would follow that whether or not a node is indicative of a sense is also defined by its neighbours. Véronis (2004) offers an algorithm that maps senses to nodes in a binary fashion, but in our understanding a probabilistic distribution would be a more fitting annotation of each node, as this leaves the possibility of a node supporting multiple senses while excluding others, without dividing sense groups.
Our system does not necessarily retain all original root hubs, as they too can be assigned a different sense during iteration (See figure 1). This allows us to over-generate root hubs in earlier steps without much repercussion.

Algorithm 1 Graph Labelling

procedure LABEL_GRAPH
  $G \leftarrow$ cooccurrence *graph*
  $H \leftarrow$ *list* of root hubs
  $stable \leftarrow False$
  for `node` $\in$ `G` **do**
    $node.\omega \leftarrow (\omega_1 ... \omega_n)$
    $\omega_1^0 ... \omega_n^0 \leftarrow 0$
    if `node` $= h \in$ `H` **then**
      $\omega_h^0 \leftarrow 1$
  $i \leftarrow 1$
  while $stable = False$ **do**
    $stable = True$
    for `node` $\in$ `G`, `h` $\in$ `H` **do**
      for `nbr` $\in$ `neighbours` **do**
        if $h = argmax(nbr.\omega)$ **then**
          $\omega_h^i \leftarrow \omega_h^i + (1 - d(node, nbr))$
    $node.\omega \leftarrow \frac{1}{i+1} \sum_{j=0}^{i} \omega^j$
    if $argmax(\omega) \neq argmax(\frac{1}{i} \sum_{j=0}^{i-1} \omega^j)$ **then**
      $stable = False$
    $i \leftarrow i + 1$
  return G

Algorithm 1 shows the process in which each node is assigned a sense distribution vector. Notably only the best sense of each neighbour and the weight of their edge[4] (d) is considered, not the entire distribution. As our graph is undirected, two conflicting nodes would, should a node's distribution be based on a neighbours own vector, tend to balance each other out, with the graph only reaching a stable state when every connected node features the same distribution, including the same 'best sense'. This is of course not a desirable outcome.

[3] A minimum spanning tree is defined as a sub-graph containing all nodes of the original graph and whose cumulative edge weights are a minimum (Kruskal, 1956).

[4] We defined the weight of an edge earlier as the inverted cooccurrence probability. As we aim to match the node to the highest score, we chose to invert the measure back for this step. An *argmin* function would work in much the same way as our method.

Algorithm 2 Disambiguation w/ Labelled Graph

procedure DISAMBIGUATE
  $S \leftarrow$ context *string*
  $G \leftarrow$ labelled *graph*
  $H \leftarrow$ *list* of root hubs
  $v \leftarrow$ score *vector* with length H
  for $token \in S$ **do**
    if $token \in G$ **then**
      for $h \in H$ **do**
        $v_h \leftarrow v_h + token.\omega_h \cdot \frac{1}{1+d(token,h)}$
  return $argmax(v)$

Our disambiguation algorithm (See algorithm 2) uses a score vector with weights for each root hub. For each token in a given context, the sense distribution vector is added to the score vector, with each sense weight adjusted by the distance of the token to the root hub.
Our system retains some binding of a sense to a root hub, using the adjustment to counteract a sense straying to far from its root during the propagation step.

5.2.2 Minimum Spanning Tree

Contexts that could not be disambiguated using the propagation algorithm are then processed by the algorithm proposed in Véronis (2004). Target string and root hubs are added to the graph with edge weights of 0. A minimum spanning tree is constructed (Kruskal, 1956) and each node assigned a score in a similar way as above:

$$score_{node} = \frac{1}{1 + d(node, roothub)}$$

Again, the scores for each token in a context are cumulated and the best sense is chosen for clustering.

Our systems returns this cumulative mapping of our propagation algorithm, supported by Véronis' components algorithm.

5.3 Baseline

We will be comparing our results to different Baselines. Firstly we will use singleton and all-in-one clustering. These are not linguistically or even mathematically motivated clustering methods, our Baseline, which is a more naïve approach to graph based word sense induction, features a basic version of Véronis' algorithm, but using conceptually simple methods and measures. Instead of the root hub selection algorithm detailed above, the baseline simply selects the ten most frequent nodes as root hubs.

The propagation and minimum spanning tree algorithms are replaced by a distance based scoring measure. Nodes v are assigned one-hot-vectors based on distance d to each root hub $h \in H$.

$$\omega_i = \begin{cases} 1, & \text{if } h_i = argmax_{h \in H}(d(h_i, v)), \\ 0 & \text{else.} \end{cases}$$

The final cumulative score vector for a given context of length n is essentially comprised of the counts of tokens w corresponding to each sense. The sense with the highest score is selected:

$$sense = argmax_{h \in H}(\sum_{h \in H} \omega_{w_1}, ..., \omega_{w_n}).$$

6 Evaluation

We use the MORESQUE dataset, consisting of 114 topics and their according search results.
To evaluate the properties of our cooccurrence graph, we use the characteristic path length and the clustering coefficient (See table 2).

6.1 Clustering Quality

SemEval-2013 Task 11 evaluates Clustering Quality on the basis of the following four metrics:

- F1-Measure
- Rand Index
- Adjusted Rand Index
- Jaccard Index

Additionally, S-recall at *K* and S-precision at *r* are measured, as well as the average number of clusters and average cluster size.

7 Results

System	F1	JI
OUR SYSTEM	**55.21**	31.73
W/O MST	53.57	33.00
W/O LABELLING	50.13	**46.20**
BASELINE	49.87	42.52
SINGLETONS	**68.66**	0.00
ALL-IN-ONE	47.42	**51.00**

Table 4: Results for Jaccard Index (JI) and F1 measure.

System	RI	ARI
OUR SYSTEM	54.73	6.98
W/O MST	**56.21**	**9.08**
W/O LABELLING	53.63	5.51
BASELINE	51.76	3.26
SINGLETONS	49.00	-0.07
ALL-IN-ONE	51.00	0.00

Table 5: Results for Rand Index (RI) and Adjusted Rand Index (ARI).

We will compare the results of our system to the results of two different versions of it. The first one doesn't use minimum spanning tree for disambiguation. The second is based on the algorithm proposed in Véronis (2004) and uses the same parameters (w/o Labelling). It however is not a faithful recreation of the original system, as the corpus used is not extracted from the target URLs. We use these two versions for ablation studies.

System	50	60	70	80
OUR SYSTEM	33.99	22.51	**17.78**	**14.51**
W/O MST	**36.82**	**22.98**	17.18	13.94
W/O LABELLING	31.73	20.68	15.83	12.57
BASELINE	32.75	22.47	15.21	13.96

Table 6: S-precision@r

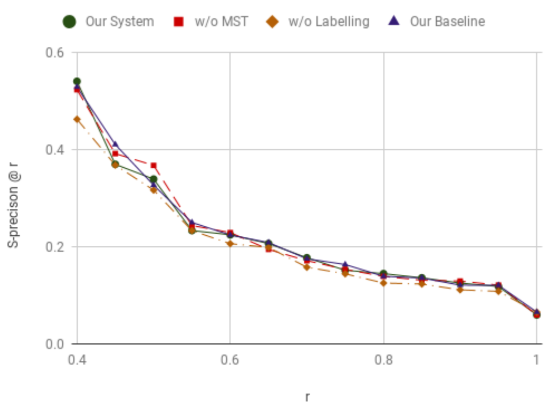

Figure 2: S-precision@r

Our system outperforms every baseline on the development data, as would be expected. The three versions of our system vary heavily depending on measure. Our system with our propagation algorithm and minimum spanning tree as backup performs well on F1-Measure, but lacks in Jaccard Index. Our recreation of Hyperlex has the best Jaccard Index, but is behind every other system in all other measures. Jaccard Index may be biased towards fewer larger clusters, as both our system without labelling and all-in-one clustering perform best in this category. Removing the minimum spanning tree as backup boosts Adjusted Rand Index significantly, with a smaller bump in Rand Index.

System	# cl	ACS
GOLD STANDARD	3.98	19.83
OUR SYSTEM	5.39	22.99
W/O MST	4.82	20.61
W/O LABELLING	1.46	74.81
BASELINE	4.54	33.69

Table 7: Average number of clusters (# cl.) and average cluster size (ACS).

The gold standard features a smaller number of clusters with a high average cluster size, which would indicate that the development data may not be an entirely accurate representation of most sense distributions, as other sets have shown to have different distributions (Navigli and Vannella, 2013). We expect better performance for Rand Index and Adjusted Rand Index on a different dataset.
We are hesitant to remove Véronis' components algorithm as backup, as the influence of the minimum spanning tree is only minimal, but it supports our system with a tried and tested approach, which may outweigh the performance gain indicated on the development set.
The low average cluster count may also have affected the remarkably high performance of all-in-one clustering, outperforming every other system in Jaccard Index and Rand Index by a large margin. We expect this performance to drop significantly when testing on datasets with higher cluster counts.
In terms of precision (See table 6) and recall (See table 8), our full system and our system without minimum spanning tree perform about the same, which is expected due to the small influence the minimum spanning tree has on the results. In both metrics, our system without label propagation and dynamic limits trails behind every other version of our system, as well as the baseline.
Across the board, Adjusted Rand Index has been the most reliable information about the performance of our system, with the other measures being more susceptible to changes in cluster size and count. While accurate prediction of number of senses is certainly an important part of the task, we

felt overall clustering quality had to be optimised before any reasonable approach in this direction could be taken.

System	5	10	20	40
OUR SYSTEM	51.58	**70.32**	**78.21**	**88.44**
W/O MST	53.46	69.52	77.83	88.21
W/O LABELLING	**55.99**	65.77	73.75	84.69
BASELINE	55.14	66.25	76.18	87.41

Table 8: S-recall@K

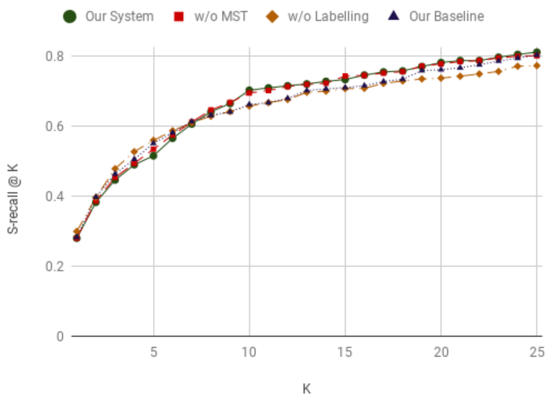

Figure 3: S-recall@K

8 Conclusion

Working with a graph based system has offered many opportunities to learn about a field of research we previously had little knowledge about. The similarity of cooccurrence networks and human relations in small world graphs lead to a broad spectrum of possible approaches to optimising a system that had been tried and tested for over a decade. Our system producing pretty good results on the development data has been an added bonus. We were surprised by the performance of our system so far. Hyperlex has proved to be a very robust baseline on which to build on. Working with a different and less balanced corpus than the original has lead to interesting problems, which we approached both with more flexible parameters and a different, more forgiving algorithm.

Small world graphs, not really a native field of computational linguistic research, have proven themselves quite apt in modelling semantic relations, which we did not expect before researching this project. In the end, words are closer than people.

Even though the graphs our system built were useful and stable, better results could be obtained by using various sources instead of the Wikipedia

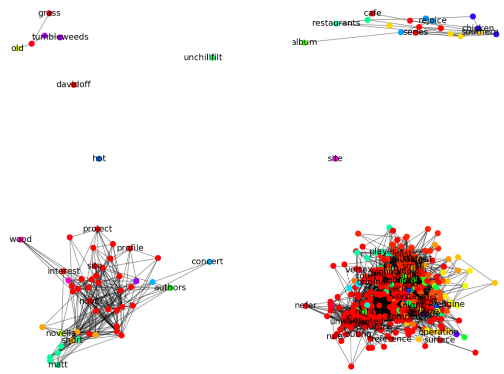

Figure 4: Graphs of different sizes.[6]

corpus. Especially proper names of obscure bands and other pop culture references have posed a challenge to our system, which could have been solved with a less informative and more entertainment based corpus.

We chose not to use the URLs for multiple reasons, mostly because with internet connectivity it would add another dependency to our system, without really offering the whole span of possibilities the web entails. Our corpus is formatted in plain text, which would not be available for text extracted with URLs. An HTML text extractor and special stop word list would have needed to be added as well and would have bloated our system with little gain.

We would have liked to have tested a few more features with more time and a larger development set. The number of minimum neighbours for a root hub is still heuristic[5] and the same reasoning for making minimum frequencies dynamic would have applied here. $log(\Gamma(i)) \cdot \Gamma(i)$ was tested to promising results, but still performed worse than the heuristic measure.

Additionally we would have liked to create a version of our system where the disambiguation task is fully independent of our root hub list after labelling. In the end, the possible gain did not outweigh the overall overhead the implementation would create. Still, making our system less dependent on Véronis (2004) was remains a daunting idea to us. In the end, this has been a learning experience beyond semantics and graphs. We did pick up quite a lot about tools and resources in researching and managing a project of this size. This

[5] We lowered it from 6 to 5 for our system based on limited tests on the development data.

[6] From top left to bottom right: cool_water, soul_food, stephen_king, the_block

will certainly come in handy when it comes to the software project and other course projects.

Acknowledgements

We wish to thank our previous project collaborator Bente Nittka for believing in us and our system. Special thanks to Atila Martens and Michael Staniek for proof reading. And finally, Marinco Holzinger and Simon Will for their magic [7] in getting our system up and running as smooth as humanly possible.
Also cats.

References

Ramon Ferrer i Cancho and Richard V. Solé. 2001. The small world of human language. *Proceedings of the Royal Society of London B: Biological Sciences* 268(1482):2261–2265. https://doi.org/10.1098/rspb.2001.1800.

Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics* 39(3):709–754.

James H. Fowler. 2007. Turnout in a small world .

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 595–605. http://aclweb.org/anthology/D/D16/D16-1057.pdf.

Joseph B. Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7(1):48–50. http://www.jstor.org/stable/2033241.

Roberto Navigli and Daniele Vannella. 2013. Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In Mona T. Diab, Timothy Baldwin, and Marco Baroni, editors, *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*. The Association for Computer Linguistics, pages 193–201. http://aclweb.org/anthology/S/S13/S13-2035.pdf.

M. E. J. Newman. 2003. The structure and function of complex networks. *SIAM REVIEW* 45:167–256.

Jeffrey Travers and Stanley Milgram. 1969. An experimental study of the small world problem. *SOCIOMETRY* 32(4):425–443.

[7] i.e. unwilling support

Rocco Tripodi and Marcello Pelillo. 2016. A game-theoretic approach to word sense disambiguation. *CoRR* abs/1606.07711. http://arxiv.org/abs/1606.07711.

Dmitry Ustalov, Alexander Panchenko, and Chris Biemann. 2017. Watset: Automatic induction of synsets from a graph of synonyms. *CoRR* abs/1704.07157. http://arxiv.org/abs/1704.07157.

Jean Véronis. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language* 18(3):223–252. https://doi.org/10.1016/j.csl.2004.05.002.

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442. https://doi.org/10.1038/30918.