

# 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<sup>1</sup> 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 with a more dynamic feature set as well as recent methods, previously used mostly for sentiment analysis.

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 as set by the original task participants.

<sup>1</sup> 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: Meta parameters for building a cooccurrence graph for the analysed systems.

## 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 the properties of small world graphs for sense disambiguation has been shown previously in Newman (2003). The term 'Small World' was introduced by Travers and Milgram, who used 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) and we will see later on that the graph we are using is indeed a small world graph, as the words are connected in a similar way.

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 explained later on. Why this weighting algorithm is chosen over something like Dice weights is not further explained, but we expect an algorithm using Dice weights to 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 nodes of the cooccurrence graph, given the following criteria:

1. The candidate node has at least six neighbours, excluding root hubs and neighbours of root hubs.
2. The six most frequent neighbours without root hubs and neighbours of root hubs have a mean weight under 0.8.
3. The candidate is not a neighbour of any previously chosen root hub.

The underlying algorithm to fulfil these criteria is explained further into the documentation.

Before building the minimum spanning tree, the target string is inserted back into the graph with a distance of 0 to each root hub.

For disambiguation, Véronis iterates over each node  $v$  in the minimum spanning tree and assigns each a weight vector  $\omega$ :

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

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 graph propagation (Hamilton et al., 2016).

### 3 Task Setup

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 word, that we will consider is a topic. For every topic there is a list of the first 100 Internet search results, containing information on the found web page, namely the URL, title and a text snippet.

### 3.1 Corpus

Our system uses an unordered plain-text Wikipedia corpus 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

As noted before, the graphs we will build are so called 'small world graphs'. The connection topography of a small world graph, as described in Watts and Strogatz (1998) lies between a complete random and a complete ordered one. 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 two nodes in the graph and 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 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

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.

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 same size.

The random measures are calculated as follows:

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

$$C_{rand} \sim 2k/N.$$

As can be seen in table 2, our graphs resemble small world graphs, because they have a short Average Path Length, but a significantly higher Clustering Coefficient than a random graph would have.

Véronis used these properties for root hub detection. We included another system for disambiguation, namely a graph propagation algorithm, that again builds on these properties of the graph. Because our corpus is much less balanced than Véronis (2004) and our task is more varied<sup>2</sup>, we use a more flexible set of parameters and methods. This task setup 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 the euclidean mean offers a solution to the problem of sparse graphs for less represented terms, more frequent terms seem to overgenerate 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 Systems

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

<sup>2</sup>Véronis mostly disambiguates highly polysemous terms and no proper names.

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.

### 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 sub-corpus 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 4 relevant tokens is discarded.

Every node or edge, which 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 overgeneration of root hubs and our sub-corpora vary in size too considerably to allow heuristic senses without undergenerating root hubs for less frequent targets. The graph is weight using the following method:

$$\omega_{a,b} = 1 - \max[p(A|B), p(B|A)], \quad \text{with}$$

$$p(A|B) = f_{A,B}/f_B \quad \text{and}$$

$$p(B|A) = f_{A,B}/f_A.$$

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. To collect root hubs, we use the algorithm shown in Véronis (2004), iteratively choosing root hubs by their degree and average weight with their most frequent neighbours (See table 3). We then delete

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.

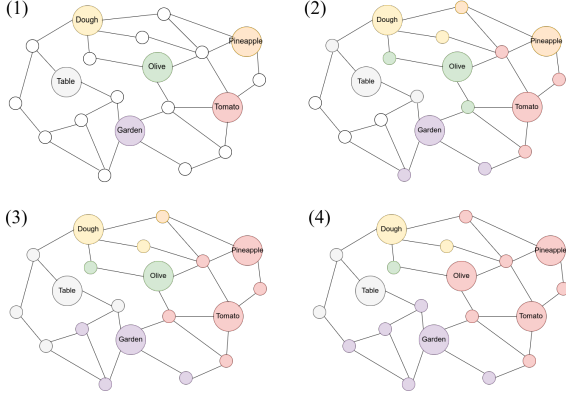


Figure 1: Example of Propagation for the target 'Pizza'.

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<sup>3</sup> 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

<sup>3</sup>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).

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 overgenerate root hubs in earlier steps without much repercussion.

### Algorithm 1 Graph Labelling

---

```

1: procedure LABEL_GRAPH
2:    $G \leftarrow$  cooccurrence graph
3:    $H \leftarrow$  list of root hubs
4:    $stable \leftarrow False$ 
5:   for node  $\in G$  do
6:      $node.\omega \leftarrow (\omega_1 \dots \omega_n)$ 
7:      $\omega_1^0 \dots \omega_n^0 \leftarrow 0$ 
8:     if node  $= h \in H$  then
9:        $\omega_h^0 \leftarrow 1$ 
10:     $i \leftarrow 1$ 
11:    while  $stable = False$  do
12:       $stable = True$ 
13:      for node  $\in G, h \in H$  do
14:        for nbr  $\in$  neighbours do
15:          if  $h = \text{argmax}(nbr.\omega)$  then
16:             $\omega_h^i \leftarrow \omega_h^i + (1 - d(node, nbr))$ 
17:           $node.\omega \leftarrow \frac{1}{i+1} \sum_{j=0}^i \omega^j$ 
18:          if  $\text{argmax}(\omega) \neq \text{argmax}(\frac{1}{i} \sum_{j=0}^{i-1} \omega^j)$  then
19:             $stable = False$ 
20:           $i \leftarrow i + 1$ 
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<sup>4</sup> ( $d$ ) 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.

<sup>4</sup>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

---

```

1: procedure DISAMBIGUATE
2:    $S \leftarrow$  context string
3:    $G \leftarrow$  labelled graph
4:    $H \leftarrow$  list of root hubs
5:    $v \leftarrow$  score vector with length  $H$ 
6:   for  $token \in S$  do
7:     if  $token \in G$  then
8:       for  $h \in H$  do
9:          $v_h \leftarrow v_h + token.\omega_h \cdot \frac{1}{1+d(token,h)}$ 
10:  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:

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

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. Because these Baselines aren't linguistically motivated, but mere clustering solutions we added another Baseline, which is a naïve approach to graph based word sense induction. 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:

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

## 6 Evaluation

We use a very small subset of the MORESQUE dataset, consisting of four 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	70.95	41.65
W/O MST	67.87	42.07
W/ HYPERLEX PARAMETERS	64.38	40.76
BASELINE	66.63	45.59
SINGLETONS	75.51	0.00
ALL-IN-ONE	66.09	69.28

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



System	RI	ARI
OUR SYSTEM	57.32	12.30
W/O MST	57.56	13.81
W/ HYPERLEX PARAMETERS	50.58	00.74
BASELINE	48.77	-3.57
SINGLETONS	30.72	-0.01
ALL-IN-ONE	69.28	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 (Hyperlex). It however is not a faithful recreation of the original system, as the corpus used is not extracted from the target URLs.

System	50	60	70	80
OUR SYSTEM	49.05	34.17	24.68	20.25
W/O MST	50.83	35.21	25.40	20.77
W/ HYPERLEX PARAMETERS	31.11	25.36	20.59	14.40
BASELINE	34.23	21.18	15.21	12.92

Table 6: S-precision@r

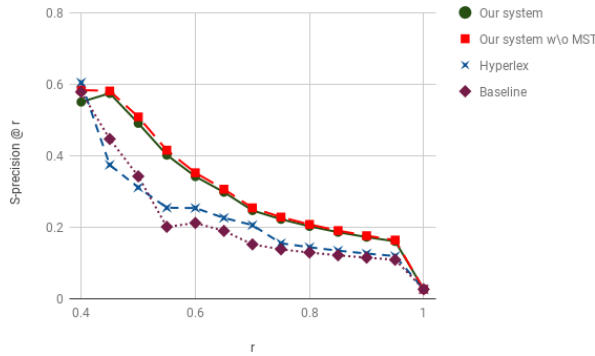


Figure 2: S-precision@r

Our system outperforms every baseline on the development data, as would be expected. Because of the small size of the development data and the heuristic parameter values, our recreation of the Hyperlex system ([Véronis, 2004](#)) did not create any clusters for cool\_water, which resulted in a significant drop in performance, as can be seen in figure 2, as well as in the results for our baseline using similar measures.

System	# cl	ACS
GOLD STANDARD	2.75	34.83
OUR SYSTEM	6.5	16.69
W/O MST	6	15.43
W/ HYPERLEX PARAMETERS	3	49.56
BASELINE	5.25	25.45

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

The gold standard features a very small 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 ([Diab et al., 2013](#)). We expect better performance for Rand Index and Adjusted Rand Index on a larger dataset.

Any difference between our system and our system without [Véronis'](#) components algorithm as backup is well within standard variance. The influence of the minimum spanning tree is only minimal, but it supports our system with a tried and tested approach, which easily outweighs any performance gain indicated on this admittedly small 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.

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.

Notable is also the difference in Adjusted Rand Index between the two systems with flexible node and edge number minimums (our system) and those without. This change could be attributed to the failure to induce cool\_water as well, but needs to be tested on a larger dataset.

System	1	2	4	8
OUR SYSTEM	39.58	45.83	58.33	91.67
W/O MST	39.58	45.83	66.67	91.67
W/ HYPERLEX PARAMETERS	31.25	39.58	62.50	79.17
BASILINE	39.58	54.17	54.17	70.83

Table 8: S-recall@K

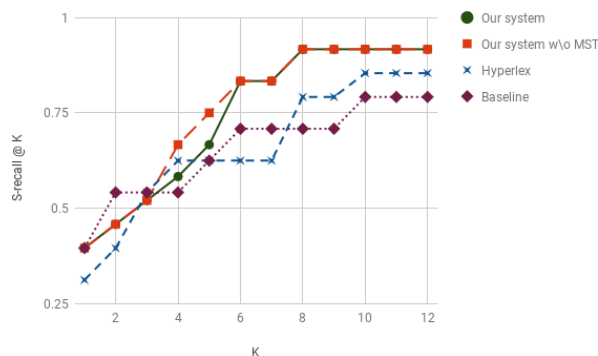


Figure 3: S-recall@K

## 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.

Working with such a limited dataset has shown to be quite challenging without implementing some systems to not only evaluate the finished output, but also smaller steps along the way. We would have liked to have more data available, but it forced us to get more in-depth with our system, which helped our project in the long run. A larger dataset would have allowed us to partition a held-out set for internal testing though, which would have been quite useful along the way.

Small world graphs, not really a native field of computational linguistic research, have proven themselves quite apt in modelling semantic rela-

tions, 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 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<sup>5</sup> 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.

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 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 for proof reading. And finally, Michael Staniek and Simon Will for their magic<sup>6</sup> 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.](#)

<sup>5</sup>We lowered it from 6 to 5 for our system based on limited tests on the development data.

<sup>6</sup>i.e. relentless support

*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.

Mona T. Diab, Timothy Baldwin, and Marco Baroni, editors. 2013. *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.  
<http://aclweb.org/anthology/S/S13/>.

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 (Diab et al., 2013), 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.

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>.