Syntax-sensitive models for distributional semantics

Simon Buijs, Ilse Griffioen, Konstantinos Kogkalidis, Jasper Munnichs, Steven Veerbeek
student numbers: 6311733, 6304354, 6230067, 3840530, 3923363

Abstract

A recently emerging approach to natural language processing is distributional semantics. Distributional semantics are an application of the notion that a words meaning can be derived from its context. While largely successful, most established state-of-the-art models assume the same, coarse way of determining context; through positional distance. We look into a different way of determining context, namely by using dependency trees and show that this approach can potentially outperform models based on positional distance in defining various semantic relations.

Keywords: natural language processing; distributional semantic models; syntactic word embeddings

Introduction

Natural language processing is an important domain of Artificial Intelligence. The increased availability of cheap computational resources and larger corpora, combined with the recent bloom in machine learning techniques has led to a rapidly growing interest in new methods of text processing. One particular approach, which has been introduced and subsequently established as common practice in recent years, are distributional semantic models (DSM). Such models attempt to model the meaning of words by representing them as vectors in a high dimensional space. Such representations are easy to perform computations on, thus enabling complex tasks such as question answering and machine translation (Molino, Basile, Caputo, Lops, & Semeraro, 2012; Turney & Pantel, 2010).

DSMs rely on the distributional hypothesis; words that occur in similar contexts tend to have similar meanings (Clark, 2012). Under this notion, they use the frequency statistics of words to abstract over to word features. Based on the contexts, a co-occurrence matrix is created, which counts the number of times every word appears together with every other word. The meaning of a word is thus represented by the resulting vector of co-occurrences in semantic space. This matrix can then be reduced to denser, more abstract features through a multitude of techniques borrowed from linear algebra and statistical learningthe resulting representations are called word embeddings. Words can finally be compared using the measure of cosine distance in semantic space; small cosine distance equates to similar meanings (Singhal, 2001).

Such methods of constructing vectors quickly rose to prominence, proving extremely efficient in capturing information under minimal to no supervision. A number of possible alterations in the process have been proposed (Pennington, Socher, & Manning, 2014; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), with the state of the art being under constant evolution during the last few years (Ruder, 2017). Despite their differences, one common element in these approaches has been the way of constructing the context; the vast majority of them create context by means of positional distance (i.e. how far away words are from one another in the sentence), thereby completely discarding any structural information contained within the sentence.

Given the presumed naivety of this approach, it is perhaps surprising how easily such models were established and popularized as the go-to method for word embeddings. Some criticisms have been raised (Sidorov, Velasquez, Stamatatos, Gelbukh, & Chanona-Hernández, 2014; Levy & Goldberg, 2014), amongst others, in which alternatives that take into account the syntax of each particular sentence have been proposed.

Motivated by this, we formulate our research question as how do syntax-sensitive DSMs compare against syntax-insensitive DSMs in representing semantic relations. In an attempt to answer this, we discuss in our paper three models which implement the distributional hypothesis and construct co-occurrence matrices in different ways. The first model adopts the positional distance approach by sliding a window of varying size along the words across a sentence and adding all other words occurring within that window to the inspected words context. The other two models construct the context examining purely the syntax tree of the sentence; the first is a variation of the one Levy and Golberg propose, whereas the second is a novel method that we propose. These models are tested using the WBST (Freitag et al., 2005) and BLESS tests (Baroni & Lenci, 2011).

A detailed description of our models can be found in section 2. Section 3 elaborates on the tests we use and our experimental setup. In section 4 we posit the results we obtained, and in section 5 we discuss our findings. A brief conclusion is presented in section 6.

Models

In this section, we describe the models that we created for this research. We go into detail on how these models operate and how they construct their co-occurrence matrix.

Window Model

The first model functions as a baseline to compare the syntactic models against. This model counts, for every word, how many times it appears within a specified positional distance together with every other word that is present in the corpus; this distance is called the window size. For each word this results in an n-dimensional vector, where n is the number of words in the corpus. All these vectors together represent an n n matrix, called the co-occurrence matrix. We test with window size three and six. An example of the models functionality is presented in figure 1.

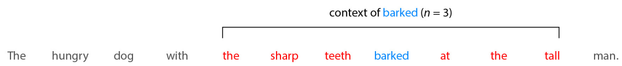


Figure 1: Context of barked in a 3-word-window model.

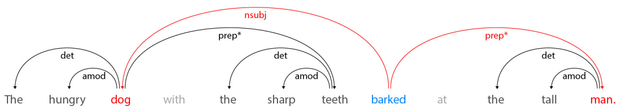


Figure 2: Dependency tree of the sentence The hungry dog with the sharp teeth barked at the tall man.

Two implementation details are worth noting; one is that the occurrence statistics for words with the same lemma are aggregated in the final matrix in order to reduce sparsity and matrix size. We apply this procedure for all three models. The other is that we disallow contexts from crossing sentence borders, following Lison and Kutuzovs findings that this hurts performance (2017).

Syntactic Models

As figure 1 shows, the baseline model tends to introduce a significant amount of noise into the context, as it discards any structural information contained in the sentence. Syntactic models bypass this issue by operating on dependency trees; these map dependency relations between different words and phrases in a tree-like structure. We obtain these dependency trees using Honnibal and Johnsons algorithm (2015)[1], which achieves performance surpassing 90% accuracy while maintaining low computational complexity.

We test two different methods; one is based on the work of Levy and Golberg (2014), while also proposing a novel method which operates on noun phrases as the base syntactic unit.

Head-Children Model Levy and Goldberg consider the context of each word as the set of its parent (head) and children in the dependency tree. Their model also involves skipping prepositions by establishing a head-child relation directly between the head and the children of each preposition. We extend this model by considering the words in conjunct and disjunct phrases to be of distance zero with one another. The reasoning for this is that all elements of a conjunction or disjunction can be listed in any order. As such they are relevant not only to one another but also to the head of the first element. Using this approach, figure 2 shows that a smaller but more relevant context can be constructed, improving information and reducing noise.

Noun-Chunk Model As an alternative, we came up with a second syntactic model, which considers noun phrases; these are phrases which fulfill a single grammatical role in a sentence. This model operates by collapsing noun phrases into a single entity, the elements of which have distance zero with one another. This is based on the belief that noun phrases include valuable contextual information, that is largely ignored when a model is only allowed a single step in the syntax tree.

Test Methods

In this section we will discuss the tests that we used to evaluate our models and briefly discuss the corpus that was used.

Corpus

For this research, we selected the Open American National Corpus (OANC)[2]. It consists of both written and spoken text on different subjects, and features about 15 million words of American English spread over 9,000 sample texts. We used this corpus because it is of adequate size and the topics covered are very diverse and common, allowing for easier testing.

A few limitations were necessary in order to improve the quality of the data extracted from the corpus. To begin with, all spoken transcripts were removed; these naturally contain a high number of fillers such as you know and uh, which corrupt the co-occurrence matrix while making parsing harder due to non-formal grammar. Additionally, words with either too high or too low term-document frequency were discarded; the first appear too often to contain any useful information, whereas the latter are either very domain-specific, typographical errors or post-processing artifacts. We test with different term-frequency lower boundaries, below which a words meaning vector is considered too sparse and distorted to draw meaning from. After these processing operations, the working vocabulary is reduced to about 20,000 unique lemmata, down from an original size of over 200,000.

Wordnet Based Synonymy Test

Previous studies have used different tests to measure the performance of their models. This section discusses the WordNet Based Synonymy Test (WBST) (Freitag et al., 2005) and justifies our choice. WBST is based on the TOEFL test (Landauer & Dumais, 1997) and contains 23570 multiple-choice questions, instead of TOEFLs 80. The models are given one target word and four possible answers. The models need to pick which of the four words is most synonymous to the target word. Their performance is measured by the number of correct answers.

We selected this test because we regard it as being the most objective. Alternatives to it are word pair tests, such as the MEN test (Bruni, Tran, & Baroni, 2014) and WordSim-353 (Finkelstein et al., 2001), 2001), and tests based on thesauri, such as Rogets Thesaurus (Jarmasz & Szpakowicz, 2004), the MacQuarie Thesaurus (Bernard, 1990) or WordNet (Fellbaum, 1998).

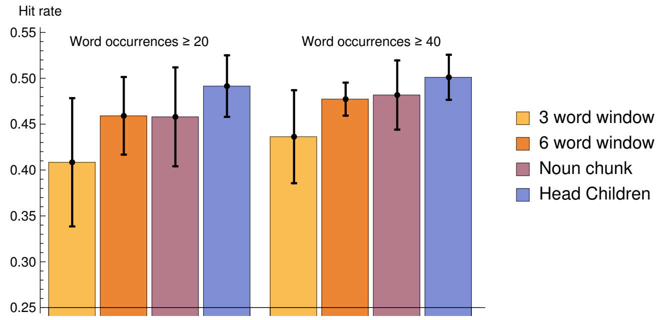
Examples of such studies are done by Hirst & St.-Onge (1998) and and Jarmasz and Szpakowicz (2004). Both of

[1] https://spacy.io

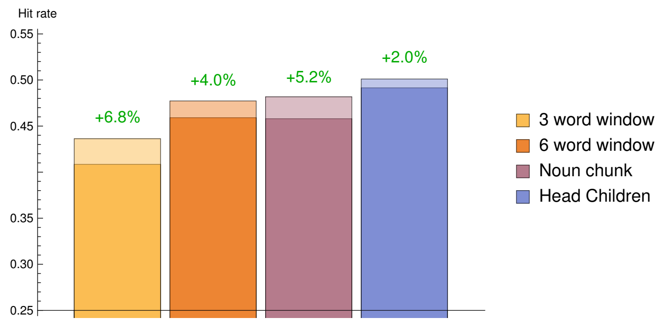
[2] http://www.anc.org/

		Categories				
		Coord	**Hyper**	**Mero**	**Attri**	**Event**
Concepts	**Alligator**	crocodile frog ...	animal predator ...	mouth tail ...	dangerous aquatic ...	breathe chase ...
	Bus	bike car ...	transport vehicle ...	brake color ...	fast heavy ...	arrive board ...
	Vest	blouse dress ...	apparel clothing ...	wool silk ...	black clean ...	protect keep ...

Figure 3: Indication of BLESS



(a) Hit rate of our models.



(b) Relative improvement.

Figure 4: Results of WBST.

these test methods rely more heavily on human judgment (Faruqui, Tsvetkov, Rastogi, & Dyer, 2016; Curran & Moens, 2002) than WBST. We therefore consider WBST to be more reliable.

Still, WBST has some drawbacks. As Baroni and Lenci (2011) point out, some of the questions can be ambiguous. Additionally, they argue that the test only gives a performance measure, but does not give insight into why one model outperforms the other.

For our experiment, we conduct the WBST for a minimum occurrence threshold of 20 and 40. To account for variance, we split the WBST questions into 10 mutually exclusive subsets, from which the mean and standard deviation of the performance is extracted.

BLESS Test

As we discussed, tests like the TOEFL test do not tell us why one model outperforms another. To address this issue, we also run the BLESS test (Baroni & Lenci, 2011). This test aims to reveal what types of semantic relations a DSM captures. The test consists of a list of 200 carefully selected nouns, from hereon referred to as concepts, of which we selected the 120 concepts which appeared at least 40 times in the corpus. For each of these concepts there are about 10 words in each of the eight categories of semantic relations, which are: coordinates (similar words), attributes, events (associated verbs), hypernyms, meronyms (components), and for reference three categories of random words, one for adjectives, verbs and nouns. A sample is shown in figure 3.

For each concept, the model selects the most similar word out of each category. In order to avoid discrepancies between the results, since more general words usually have higher overall similarities, the scores are normalized to zero mean and unit variance. For every model we create a boxplot, showing the relative scores. This way it shows which semantic relations a model captures well.

Results

In this section we report on the results of the WBST and the bless test.

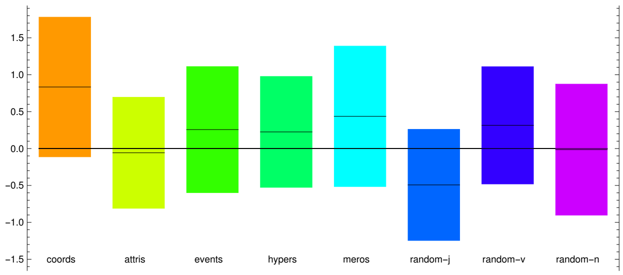
Wordnet Based Synonymy Test

The results of the WBST for word occurrence rate more than 20 and more than 40 are shown in figure 4a. The hit rate is shown from 0.25 onwards, which corresponds to random chance. In both cases the 3-word-window model (3WW) performs worst and the head-children model (HC), performs best. The 6-word-window model (6WW) and the noun chunk model (NC) perform equally well.

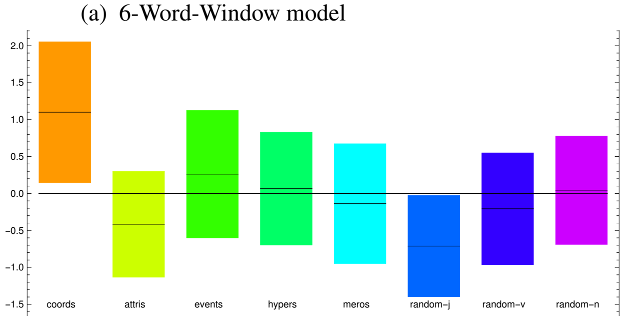
With an increase of the number of minimum occurrences all models improve their performance. Notably 3WW improves the most and HC improves the least, which is shown in figure 4b. With a minimal occurrence rate of 40 we observe a drop in standard deviation. As a result, with a minimum of 20 occurrences the only significant difference is found between HC and 3WW, while with a minimum of 40 occurrences, significant differences are found between 3WW and 6WW, 3WW and NC, and 6WW and HC.

BLESS Test

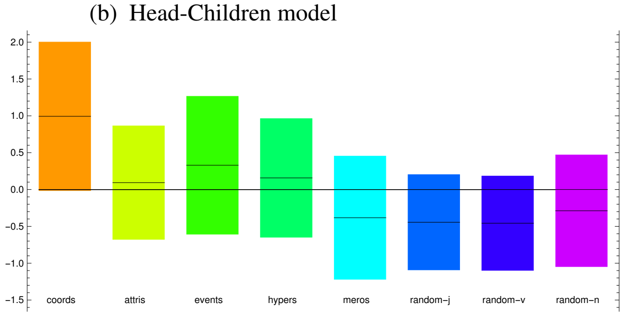
We performed the BLESS test on our three best performing models. The results are shown in figure 5, where the boxes signify one standard deviation. As could be expected, all three models score highly in the coordinate category. Despite being the highest performing method in the WBST test, HC does not achieve the highest performance on any category except for coordinates. NC correctly assigns very low values to the three random categories, while scoring highest in attributes. 6WW significantly outperforms the rest of the models on meronyms. The results on events and hypernyms



(a) 6-Word-Window model



(b) Head-Children model



(c) Noun Chunk model

Figure 5: Results of the BLESS test.

are varying and inconclusive.

Discussion

In this section we reflect on our research. In particular, we discuss how different choices and procedures could have affected our research.

Corpus

Our choice to use the OANC naturally affected out research in a considerable way. While we believe this choice is justified, different corpora would have lead to different results.

One disadvantage is the corpus size: compared to other corpora, OANC is very small. Given its generality, this leads to the ratio of total words to unique topics being rather poor. This argument is supported by the fact that our model proved highly capable at returning synonyms of biomedical terms; the corpus included a relatively high number of biomedical sample texts. A drawback, however, of a bigger corpus would be longer computation times and larger memory requirements. Our resources in that regard were limited, causing such corpora to be impractical.

As a way to save computation time, we could have used a treebank, such as the Universal Dependencies English Web Treebank[3]. However, because such corpora are manually parsed, they are usually smaller in size, and were therefore not suited for our research.

Matrix Reduction

The usability of a model is not just measured by its performance, but also by its portability and computational complexity. For this reason, a compression of the original representation into a lower dimensional space is required; common practice is the usage of a 300-dimensional feature vector instead of the co-occurrence statistics (Ruder, 2017). We made several attempts to achieve a compression with a high-bound loss, using a variety of popular algorithms such as principal component analysis, singular value decomposition and language-modeling variational autoencoders (Yang, Hu, Salakhutdinov, & Berg-Kirkpatrick, 2017). However, the extreme sparsity of the matrix, combined with the limited computational resources, made efforts to pursue matrix reduction fruitless. For this reason, the compression aspect was eventually skipped in order to avoid unnecessary noise in our results due to lossy compression.

Noun-Chunk Model Performance

The noun-chunk model we created performed worse than expected in the WBST. In particular, it performed significantly worse than the head-children model and was on par with the positional distance model that uses a context window of size six. A possible explanation of this behavior is the tendency of the algorithm to occasionally connect unrelated words with one another when linking multiple noun phrases together. This hypothesis could be tested by only allowing the parse tree traversal method to examine the noun phrase the currently inspected word belongs to.

Despite this models disappointing performance on the WBST, it scored highly on the BLESS test. This is especially interesting, given that BLESS attempts to capture a variety of linguistically complex relations rather than just focusing on synonymy. Perhaps specialized methods of context-building which use syntactic information (such as this one) are better suited for linguistic tasks other than synonymy detection. Further testing is necessary before a concrete conclusion can be drawn.

Further Research

Using relatively simple methods of traversing the syntax tree, we have achieved significant improvement over the representational capacity of word vectors that use positional distance. This naturally leads to the question of how more sophisticated approaches could potentially further increase performance. In particular, it is interesting to consider how, borrowing methodology from graph theory, we could enhance our model to look up longer distances in the syntactic space. This could possibly be done by assigning a decaying weight the further

[3] https://github.com/universaldependencies/UD_English/tree/master

away the model diverts from each inspected word (thereby incrementing the co-occurrence matrix by floats rather than unit points). Such an approach could potentially solve the problem of matrix sparsity, since each token would now be given larger, more uniform contexts, resulting in vectors of more non-zero values that are simultaneously more fine-grained.

Conclusion

In this paper we have presented syntax-sensitive methods for distributional semantic models and compared their performance against that of a syntax-insensitive method. Our work has shown that even our simple syntactic models are able to build more informative vector representations. The head-children model significantly outperforms the other models in the synonymy detection task. The noun-chunk model performs on par with the window model in synonymy detection, while slightly surpassing the other models in capturing deeper linguistic relations. These results combined with the increasing performance of parsing algorithms lead us to believe that syntax-sensitive methods of constructing word embeddings are a fertile ground for novel research and industrial applications.

Acknowledgments

This project wouldnt have been made possible without the valuable feedback from dr. Michael Moortgat and mr. Gijs Wijnholds. Wed also like to thank GeekBot[4] for offering us free use of their rented server.

References

Baroni, M., & Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the gems 2011 workshop on geometrical models of natural language semantics.*

Bernard, J. (1990). *The macquarie encyclopedic thesaurus: The book of words*. Chatswood, NSW: Macquarie Library.

Bruni, E., Tran, N.-K., & Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, *49*(2014).

Clark, S. (2012). Vector space models of lexical meaning. *Handbook of Contemporary Semantics–second edition. Wiley-Blackwell*.

Curran, J. R., & Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the acl-02 workshop on unsupervised lexical acquisition-volume 9.*

Faruqui, M., Tsvetkov, Y., Rastogi, P., & Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Fellbaum, C. (1998). *Wordnet*. Wiley Online Library.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on world wide web.*

Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., & Wang, Z. (2005). New experiments in distributional representations of synonymy. In *Proceedings of the ninth conference on computational natural language learning.*

Hirst, G., St-Onge, D., et al. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, *305*.

Honnibal, M., Johnson, M., et al. (2015). An improved non-monotonic transition system for dependency parsing. In *Emnlp.*

Jarmasz, M., & Szpakowicz, S. (2004). Rogets thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, *2003*.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *Acl (2).*

Lison, P., & Kutuzov, A. (2017). Redefining context windows for word embedding models: An experimental study. *arXiv preprint arXiv:1704.05781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems.*

Molino, P., Basile, P., Caputo, A., Lops, P., & Semeraro, G. (2012). Exploiting distributional semantic models in question answering. In *Semantic computing (icsc), 2012 ieee sixth international conference on.*

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp).*

Ruder, S. (2017). Word embeddings in 2017: Trends and future directions. `http://ruder.io`.

Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, *41*(3), 853–860.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, *24*(4).

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*.

Yang, Z., Hu, Z., Salakhutdinov, R., & Berg-Kirkpatrick, T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*.

Appendix I: Code

Most of the code that was written to enable our methods and experiments, as well as its future updates can be found at https://github.com/konstantinosKokos/syntax-driven-embeddings , available under the GPL.

[4] https://geekbot.io/

Appendix II: Reflection

In this section we discuss how each group member contributed to the project. We also consider the initial task distribution and the amount of work everyone ended up putting into the project. We end with a short note on the group process.

Konstantinos Konstantinos worked on creating a model, and helped Steven in getting started with his model. Konstantinos also provided valuable introductory sources to help the rest of the group get up to speed and become more familiar with the topic. Finally, he invested a lot of time in the most advanced part of this project: running different matrix reduction techniques in order to improve the quality of our data.

Steven Together with Konstantinos, Steven worked on creating the models for this project. After getting a little help getting started from Konstantinos, Steven was able to independently work on and deliver the more advanced of the two models. He also tried to improve the quality of the model further with multiple more specific implementations. Finally, he presented our work together with Ilse and created the presentation with her.

Ilse Together with Simon, Ilse was responsible for finding a suitable test to test both models on. With Jasper she researched statistical methods to analyze our results and wrote on the paper, while taking care of all presentation-related tasks together with Steven.

Simon Simon was responsible for finding a suitable test method together with Ilse. He also took over Ilses job of writing the paper with Jasper and functioned as the groups makeshift project manager. Lastly, he worked on identifying questions from the original WBST that could actually be used given our vocabulary.

Jasper Initially Jasper helped out with finding suitable test methods and later shifted his attention to make a start with our paper. He continued working on the paper, being aided by first Ilse and later Simon and did research on statistical methods to analyze our results.

Appendix III: Work Log

See below.

Who?	Activity	Time Spent	Date		Team Total:	29,5
Kokos	Reading papers	6			Ilse Total:	73,5
Team	First team meeting	1			Jasper Total:	78,8
Team	Team meeting class	2			Simon Total:	72
Jasper	Reading paper Lambek vs Lambek	2,5			Steven Total:	74,5
Jasper	Reading Vector Space Models..	3			Kokos Total:	103,5
Simon	Reading Vector Space Models..	2,5				
Simon	Brushing up on Linear Algebra	0,5				
Steven	Reading GloVe paper	1,5				
Steven	Reading Vector Space Models..	2				
Simon	Reading Categorial Framework	1				
Ilse	Reading Vector Space Models..	4	22/09			
Ilse	Reading GloVe paper	2	25/09			
Team	Team meeting - discussing papers, brainstorm session	2	26/09			
Team	Team meeting - discussing ideas, kick-off with Moortgat	3	28/09			
Steven	Type based embeddings implementation	3	29/09			
Kokos	Type based embeddings implementation	4,5	29/09			
Simon	Reading papers on text simplification	2	30/09			
Ilse	Reading papers on entailment	1	30/09			
Steven	Exploring SpaCy possibilities for dep. tree-based context	1	02/10			
Team	Team meeting - discussing prework	2	02/10			
Kokos	Bash scripts for crawler and pdftotext	1,5	02/10			
Jasper	Entailment plan	3				
Team	Meeting final plan	2,5	4/10			
Jasper	Create plan	2	3/10			
Simon	Create Plan	2	3/10			
Team	Meeting	1	6/10			
Kokos	Downloading and processing data, building vocabulary	6	6/10			
Steven	Constructing methods of dependency tree-based context creation	6	7/10			
Kokos	Optimizations and utility functions, building cooccurrence matrix	4	7/10			
Kokos	Optimizations	2	8/10			
Kokos	Matrix reduction (filtering), statistical fitting, code restructure	6	14/10			
Kokos	Vocabulary compatibility for future tests, PCA prework	6	15/10			
Ilse	Reading papers about word similarity evaluation	2	15/10			
Team	Team meeting - discussing shortcomings of the project	2	16/10			
Steven	not present	-2	16/10			
Jasper	BLESS	2,5	16/10			
Kokos	Reading papers on coresets for sparse PCA	2	16/10			
Ilse	Reading papers on test methods	1	16/10			
Simon	Reading papers on test methods	2	16/10			
Steven	Constructing methods of dependency tree-based context creation	3	16/10			
Steven	Reading sn-grams paper	1	17-10			
Kokos	customized VAE for dimensionality drop and analysis	4	17/10			
Simon	Reading papers on test methods	2,5	17/10			
Ilse	Reading papers on test methods	2,5	17/10			
Simon	Discussing different test methods with OG Jasper (a.k.a. not-Steven) and the Inc	1,5	19/10			
Steven	Working on dependency tree methods	1,5	19/10			
Kokos	LSTM VAE for dimensionality drop, optimizations	2	19/10			
Simon	WBST Analysis	1,5	19/10			
Ilse	Looking up pro's and con's of different test methods and discussed them	3	19/10			
Team	Team Meeting	2	20/10			
Steven	not present	-2	20/10			
Simon	WBST Analysis tool	1	20/10			
Simon	Digitalize planning last two weeks	1	20/10			
Jasper	Intro	2,7	20/10			
Kokos	Similarity pairs (giving up on compression)	2,5	21/10			
Jasper	Intro + framework	2,3	21/10			
Ilse	Write about TOEFL test	2	21/10			
Kokos	Similarity tools, maintainance	2	22/10			
Simon	Extracting valid questions, feedback on intro, WBST analysis tool	1,5	22/10			
Steven	noun chunck-based context	7	22/10			
Jasper	Rewrite intro + comment on experiment	2	23/10			
Ilse	Write experiment part, looking for references, find statistical test	3	23/10			
Steven	Another day of programming	6	23/10			
Kokos	fun debugging session	3	23/10			
Team	Meeting Moortgat and discussing models	2	24/10			
Jasper	Writing on experiment	1,5	24/10			
Steven	Working on conjunction method thing	2	24/10			
Simon	Question Extraction tool is now independent executable and outputs .csv	1,5	24/10			
Steven	Working on and testing conjunction method thing	4	25/10			
Ilse	Add references, put them into right format, editing paper	2	25/10			
Kokos	Final evaluation scripts & measurements (naive-5, hc)	2	25/10			
Kokos	Final evaluation measurements (naive-3), server maintainance, running final co-c	4	26/10			
Steven	programming/debugging	3	26/10			
Team	Meeting Moortgat/Gijs + team meeting	2	27/10			
Steven	not present	-2	27/10			
Simon	Planning last week	0,5	27/10			
Simon	Writing on paper	1,5	27/10			
Ilse	Figuring out how to analyse results	2	27/10			
Steven	Adding content to paper outlines	1	27/10			
Jasper	Statistical analysis, making figure of results	4,5	27/10			

Kokos	bless	4	27/10			
Kokos	bless	3	28/10			
Simon	Writing on paper	1,5	28/10			
Kokos	Last minute changes, bless	1	29/10			
Ilse	Making presentation	3	29/10			
Steven	Presentation	3,5	30/10			
Ilse	Making presentation	3	30/10			
Jasper	Bless + statistics	4	30/10			
Jasper	Statistics and results	2,3	31/10			
Simon	Writing on paper	2	31/10			
Jasper	Figures and Discussion	1,5	31/10			
Ilse	Making presentation	6	1/11			
Simon	Writing on paper	4	1/11			
Simon	Practicing Presentation	1,5	1/11			
Steven	Practicing Presentation	1,5	1/11			
Ilse	Practicing Presentation	1,5	1/11			
Simon	Writing paper	2	2/11			
Jasper	Paper + statistics	3,3	2/11			
Team	presentation rehearsal + final notes	4	2/11			
Jasper	not present	-4	2/11			
Jasper	Commenting and rewriting	2,5	2/11			
Team	Give presentation, answer questions, attend other presentations	4	3/11			
Simon	Write paper	3,5	3/11			
Jasper	Writing results	2,3	3/11			
Kokos	Paper	3	3/11			
Jasper	Writing	2,2	3/11			
Ilse	Comenting on paper	1	3/11			
Jasper	Starting on LateX	2,7	4/11			
Kokos	Writing on paper	5,5	4/11			
Simon	Writing on paper	5,5	4/11			
Ilse	Commenting and rewriting on paper	5	4/11			
Steven	Proofreading paper	4	4/11			
Jasper	Latex	4,5	4/11			
Jasper	Latex	2	4/11			