ICL-HD at SemEval-2016 Task 10: Improving the Detection of Minimal Semantic Units and their Meanings with an Ontology and Word Embeddings

Angelika Kirilin and **Felix Krauss** and **Yannick Versley**

Institute for Computational Linguistics
Ruprecht-Karls-University
Heidelberg

{kirilin,krauss,versley}@cl.uni-heidelberg.de

Abstract

This paper presents our system submitted for SemEval 2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM; Schneider, Hovy, et al., 2016). We extend AMALGrAM (Schneider and Smith, 2015) by tapping two additional information sources. The first information source uses a semantic knowledge base (YAGO3; Suchanek et al., 2007) to improve supersense tagging (SST) for named entities. The second information source employs word embeddings (GloVe; Pennington et al., 2014) to capture fine-grained latent semantics and therefore improving the supersense identification for both nouns and verbs. We conduct a detailed evaluation and error analysis for our features and come to the conclusion that both our extensions lead to an improved detection for SST.

1 Introduction

The SemEval 2016 Task 10 on Detecting Minimal Semantic Units of Meaning (DiMSUM) is concerned with the identification of semantic classes called supersenses for single words as well as multiword expressions (MWEs).

Identifying supersenses in text allows for abstractions that characterize word meanings beyond superficial orthography (Schneider and Smith, 2015) as well as inferring representations that move towards language independence (Schneider, Mohit, et al., 2013). It has been used to extend named entity recognition (Ciaramita and Johnson, 2003) and to support supervised word sense disambiguation as it provides partial disambiguation (Ciaramita and Altun, 2006; Ciaramita and Johnson, 2003) as well as in syntactic parse re-ranking (Koo et al., 2005) as latent semantic features.

The addition of MWEs - idiosyncratic interpretations that cross word boundaries (Sag et al., 2002) - takes into account that the supersense of a MWE is usually not predictable from the meaning of the individual lexemes. The Task moreover distinguishes between continuous MWEs like "$high\_school_{\text{n.group}}$" and discontinuous (*gappy*) MWEs like "*track* people $down_{\text{v.social}}$". The inventory of supersenses used for the task - 41 supersense classes, consisting of 26 noun and 15 verb supersenses - is derived from WordNet's top-level hypernyms in the taxonomy. They are designed to be broad enough to encompass all nouns and verbs (Miller, 1990; Fellbaum, 1990).

2 Related Work

Schneider and Smith (2015) are the first to approach SST and MWE detection jointly with a discriminative model. Most of the previous work focuses on each of the tasks in separate. Sag et al. (2002) tried to raise attention for the issue of MWEs in general and analyzed different types of MWE. Baldwin et al. (2003) employed latent semantic analysis to determine the decomposability of MWEs. Finally many MWE lexicons have been built for different purposes which is why Schneider et al. picked up the issue to address MWE annotation for general purposes (Schneider, Danchik, et al., 2014; Schneider, Onuffer, et al., 2014).

Ciaramita and Johnson (2003) first trained and tested a discriminative model for SST of unambigu-

Proceedings of SemEval-2016, pages 937–945,
San Diego, California, June 16-17, 2016.

ous nouns on data extracted from different versions of WordNet and achieved an accuracy of slightly over 52%. Curran (2005) applied an unsupervised approach based on vector-space word similarity and achieved 63% accuracy on the same data used by Ciaramita and Johnson (2003). When revisiting the task Ciaramita and Altun (2006) achieved between 70% and 77% F-score using a HMM sequence tagger.

3 System Description

3.1 Baseline System

We use AMALGrAM 2.0 (Schneider and Smith, 2015) as our baseline system. The model uses a first-order structured perceptron (Collins, 2002) with averaging. It involves a linear scoring function, a Viterbi algorithm that chooses the highest-scoring valid output tag sequence and an online learning algorithm that determines the best tagging given the current model.

We contrast four feature sets for full SST: Schneider and Smith (2015) most effective feature set which is further described in section 3.2, those baseline features plus YAGO feature (3.3), baseline features plus GloVe word embeddings (3.4), and finally a feature set that combines all the above mentioned.

3.2 Baseline Features

The feature set incorporates three components: the first is Schneider, Danchik, et al. (2014) basic MWE features, second is Brown clusters and the last component is WordNet synset features (Schneider and Smith, 2015). Basic MWE features analyze word n-grams, character pre- and suffixes, and POS tags, as well as lexicon entries that match lemmas of MWE in the sentence. We use four of the ten available lookup lexicons: *semcor_mwes* including all MWEs in SemCor (Miller et al., 1993), *WordNet_mwes* containing all MWEs from WordNet (Fellbaum, 1998), *phrases_dot_net* which is a phrase idiom lexicon[1], *wikimwe* which is mined from English Wikipedia (Hartmann et al., 2012), and *enwikt* consisting of all MWE entries in English Wiktionary. The second component of the baseline feature set provides unsupervised distributional word clusters in the form of Brown clusters (Brown et al., 1992). Those clusters reflect lexical generalizations that are useful for syntactic and semantic analysis tasks and are therefore suitable for our task. The last component of the feature set helps predicting supersenses by creating possible supersense candidates from WordNet synsets.

3.3 YAGO Feature

We implement a YAGO lookup component that provides semantic information as an additional feature for the AMALGrAM system with the purpose of improving SST for named entities. YAGO (Yet Another Great Anthology; Suchanek et al., 2007) is a knowledge resource that combines the structural benefits from the WordNet taxonomy with the richness of Wikipedia's categories system. This renders it ideal for our task by enabling us to retrieve WordNet hypernyms for many different named entities. The following part describes how we look up potential concepts in YAGO, how we find potential names of YAGO concepts within the text and how we encode the returned results as a feature for AMALGrAM.

There is no guarantee that the named entities contained in training or test data appear in the same surface form as they are stored in YAGO (e.g. "*Elvis*" vs. "*Elvis_Presley*"). We apply some heuristics should an initial query with a potential named entity not yield a result:

1. Capitalize the initial character of each token [2]
2. Try to retrieve the exact entry via Wikipedias "redirectedFrom" links that are included in YAGO (e.g. $\langle Elvis\_Presley \rangle$, $\langle redirectedFrom \rangle$, $\langle \text{“}Elvis\text{”}@eng \rangle$)
3. Drop all tokens except the first and repeat the previous steps (only applicable for sequences)

For the detection of named entities during feature extraction we rely on the gold POS tag annotation in the provided data. Whenever a token appears with a "PROPN" tag (e.g. Germany $_{\text{PROPN}}$) we query it on YAGO. If the token is followed by a continuous sequence of further "PROPN" tagged tokens (e.g.

[1] http://www.phrases.net/

[2] We found that we get the best results when capitalizing every candidate expression.

athlete	fullback	physical_entity
back	living_thing	player
causal_agent	object	preserver
contestant	organism	running_back
defender	person	whole
football_player		

Table 1: WordNet hypernyms contained in the YAGO-Entry: "Jérôme_Boateng".

FC_{PROPN} $\text{Bayern}_{\text{PROPN}}$ $\text{Munich}_{\text{PROPN}}$) the whole sequence linked with underscores is used as the search query.

The feature extraction is done in the following way: In a first step we iterate over all supersense bearing singleton and MWE nouns in our training data and try to query them in YAGO. If we have a match we extract the WordNet hypernyms for the found entity. An example for a successful query can be seen in Table 1. We accumulate a count of observed WordNet hypernyms for each supersense. From the count we calculate a tf-idf whereby the supersenses are seen as the "documents" and the WordNet hypernyms as "words". What we obtain from this procedure is a tf-idf index that tells us the significance of a WordNet hypernym for a given supersense based on our training data. We require that a WordNet hypernym has to appear at least three times with a supersense otherwise its tf-idf is set to zero for this supersense. During feature extraction we provide the information from extracted WordNet hypernyms in two ways: The straightforward way is to provide each WordNet hypernym as is. With the second feature we make use of our precomputed tf-idf index by calculating a supersense ranking whenever we find a candidate entity in YAGO. For each supersense we add up the tf-idf values of the WordNet hypernyms found with the current candidate entity. If the entity is linked to many WordNet hypernyms with high tf-idf values for a certain supersense the respective supersense will receive a high rank. [3]

3.4 GloVe Feature

The second novel feature we provide is based on word embeddings which are representations of the meaning of words in terms of real-valued vectors in a low-dimensional vector space. Because such embeddings provide generalizations over the meaning of words, including words that do not occur in the training data, they are often used as a general way to improve accuracy in the form of extra word features (Turian et al., 2010). Two of the most popular methods to create such a mapping include: global matrix factorization and the local context window method. Pennington et al. (2014) combine both methods in the GloVe word embeddings. The available word vectors are derived from a 2014 Wikipedia[4] dump and the Gigaword 5 corpus[5]. Both sources together comprise 400,000 word types and there are four versions available that differ in the size of their dimensions: 50, 100, 200 and 300. GloVe word embeddings capture fine-grained semantic and syntactic regularities using a global log-bilinear regression model with a weighted least-squares objective. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence.

We incorporate our feature set using a similar method to the lookup lexicons. GloVe word embeddings are essentially a dictionary where each entry consists of a word type and their word vector. If a given lowercase token matches a GloVe lexicon entry then we extend the feature vector of that token with its corresponding word embeddings. Turian et al. (2010) mention that the weights in word embeddings are not necessarily in a bounded range. If the range of the word embeddings is too large, they will exert more influence than the remaining features. To prevent the word embeddings from exerting too much influence on the prediction when they consist of an unbounded range of real numbers we adopt Turian et al.'s (2010) method of scaling the word vectors. Assuming that all word embeddings are represented in a matrix E: Each row E_i contains the word embeddings of a token, each column E_j represents one dimension of the word embeddings and each cell contains a word embedding E_{ij}. We scale each word vector dimension by a scaling constant σ and the inverse of the standard deviation over

[3] We achieve the best results when using the top ranked supersense as well as all additional supersenses that receive at least half of the top ranked supersense's tf-idf score.

[4] http://dumps.wikimedia.org/enwiki/20140102/

[5] https://catalog.ldc.upenn.edu/LDC2011T07

the values across all words:

$$w_{new} = \frac{\sigma * E_{ij}}{stdev(E_j)} \quad (1)$$

4 Experiments

In the following section we present the data we used, then the tools and parameters and finally the results of our experiments.

4.1 Data

Our training and test data consists of the data made available for SemEval 2016 Task 10. The training data includes three harmonized data-sets: STREUSLE 2.1 (Schneider and Smith, 2015), Ritter and Lowlands Twitter dataset (Johannsen et al., 2014). The test set also consists of three sources: online reviews from the TrustPilot corpus (Hovy et al., 2015), tweets from the Tweebank corpus (Kong et al., 2014) and TED talk transcripts (Cettolo et al., 2012; Neubig et al., 2014). All datasets use the 17 Universal POS categories and the extended BIO scheme from Schneider and Smith, 2015. For feature development we used a shuffled held-out portion of the train set.

4.2 Experimental Setup

We conduct the experiments on the aforementioned DiMSUM data sets. We use the AMALGrAM 2.0 tagger as baseline system with the following parameters:[6] four training iterations, features that appeared less than five times were cutoff, a constraint for the decoding process that asserts that the "O" label is never followed by an "I", including loss term and a cost penalty of 100 for errors against recall. Furthermore we use brown clusters and five MWE lexica mentioned in section 3.2. Additionally we create a new tagset that suits the DiMSUM tags. We only consider tags occurring in the DiMSUM training set, yielding $|Y| = 170$ tags.

$$\underbrace{|\{BbOo\}|}_{4} \times \underbrace{(|N| + |V| + |\emptyset|)}_{26+15+1} + \underbrace{|\{Ii\}|}_{2} = 170$$

All evaluation scores where obtained using the evaluation script for SemEval 2016 Task 10.

[6]We have been orienting ourselves towards the parameters used in Schneider and Smith (2015).

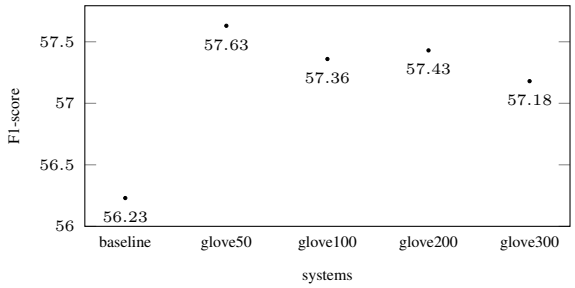

Figure 1: Evaluation of the influence of high dimensional word embeddings on the performance of the baseline system using combined F1-scores.

4.3 YAGO Feature Experiments

The effects of the YAGO feature on AMALGrAM can be seen in Table 5. Compared to the baseline (BL) the WordNet hypernyms have almost no effect on MWE detection (+0.02) while improving SST (+0.64). The supersense rankings improve MWE detection (+0.49) as well as SST (+0.62). Combining the features further improves the detection for MWEs (+0.59) and SST (+1.13).

4.4 GloVe Feature Experiments

The first experiment involves the evaluation of different word embedding dimensions. Figure 1 shows a comparison between the baseline system (BL) which uses no word embeddings and our system adopting GloVe word embeddings with four dimension sizes: 50, 100, 200 and 300. This experiment uses word embeddings in their given real-valued form without scaling them. All systems using word embeddings show improved performance and are approximately one percent higher than the baseline. In this experiment the system performs best with 50 dimensional word embeddings, after which the performance shows a slight decrease with growing vector dimensionality.

The second experiment evaluates the influence of unscaled versus scaled word embeddings using the method of Turian et al. (2010) which we described in section 3.4. Table 2 compares the F1-scores of four systems: the first - *Glove50* - uses unscaled word embedding whereas the remaining three systems scale those word embeddings with varying σ-values (0.01, 0.1 and 1). According to Turian et al. (2010) their method works best with a σ-value that scales the standard derivation to 0.1. With our data

	Glove50	σ=0.01	σ=0.1	σ=1
MWE	57.52	**58.49**	57.71	58.38
SST	57.65	57.14	57.04	**57.98**
combined	57.63	57.36	57.15	**58.05**

Table 2: Influence of unscaled (Glove50) versus scaled word embeddings for MWE, SST and both tags (combined) using F1-score. Last three columns use Turian et al.'s 2010 scaling method with varying σ-values.

	-brown		+brown	
	BL	**Best**	**BL**	**Best**
MWE	58.13	57.69	58.10	**58.38**
SST	56.28	57.43	55.87	**57.98**
combined	56.59	57.47	56.23	**58.05**

Table 3: Comparison of baseline (BL) and our most successful system (Best) with (+brown) and without (-brown) Brown cluster.

that value is $\sigma = 0.1$. However our results contradict Turian et al. (2010) as our experiment shows that $\sigma = 0.01$ is more successful in predicting MWEs whereas $\sigma = 1$ is more suitable for the detection of supersenses as well as both combined.

The third experiment evolves around the interaction between Brown clusters and word embeddings. As both methods have a similar aim - capturing the semantic representation of words - it is of interest to distinguish their influence on the system performance. To accomplish this we train the baseline system (BL) and our most successful system (Glove50, $\sigma = 1$) with (+brown) and without using Brown clusters (-brown). Table 3 represents the resulting F1-scores. Our system profits from the Brown clusters as the F1-scores for all categories (MWE, SST and combined) improves.

4.5 Final System Comparison

Finally we evaluate the combined impact of our features on the performance of the baseline system. Table 5 compares the baseline system plus YAGO, baseline plus GloVe and our combined system using both YAGO and GloVe features (final)[7] with various measurements. Our GloVe feature produces the overall best results except for the detection of MWEs due to a decrease in recall. In this case the YAGO feature improves all measures which leads to the highest performance. Although the combination of both features (final) results in an improvement of the baseline the performance mostly ranges in between the results for the individual GloVe and YAGO features.

	system	Acc	P	R	F1
MWE	YAGO	91.70	70.43	**50.31**	**58.69**
	GloVe	**91.99**	**73.47**	48.43	58.38
	final	91.70	71.74	47.35	57.05
SST	YAGO	85.02	55.96	58.08	57.00
	GloVe	**85.34**	**56.82**	**59.20**	**57.98**
	final	85.15	56.52	59.09	57.78
combined	YAGO	81.29	57.98	56.60	57.28
	GloVe	**81.81**	**58.97**	**57.15**	**58.05**
	final	81.57	58.49	56.86	57.66

Table 4: Accuracy (Acc), precision (P), recall (R) and F1-score (F1) for MWE detection, SST and both (combined) for the baseline system (BL) + YAGO, BL + GloVe and BL + YAGO + GloVe (final).

5 Feature Analysis

To get a deeper understanding of the impact and benefit of our features we conduct the following analyzes. First we compare the coverage of our YAGO lookup with NLTK's WordNet component and examine the accuracy of our ranking feature. Then we analyze the coverage of GloVe on the provided data set and give a detailed recall analysis for each tag.

5.1 YAGO coverage

For the coverage comparison we extract all supersense bearing nouns from the whole gold annotated DiMSUM data set. We query the extracted nouns and count the ones that are found exclusively with the lookup component of the YAGO feature and not by NLTK-WordNet. Figure 2 (see Appendix A) displays the results for each supersense that has at

[7]This is a revised version of our submitted system for SemEval 2016 where we used our YAGO combined and unscaled GloVe100 features. The results of our submitted system were slightly better (combined F1-score 57.77%), but could not be reproduced after revising our system.

	BL	Hyp	Rank	H+R
MWE	58.10	58.12	58.59	**58.69**
SST	55.87	56.61	56.49	**57.00**
combined	56.23	56.86	56.84	**57.28**

Table 5: Evaluation of the YAGO feature with F1-score comparing baseline (BL), baseline + WordNet hypernym features (Hyp), baseline + ranking features (Rank) and baseline + both additional features combined (H+R).

least 100 associated nouns. Looking at the results we observe a decreased coverage of the WordNet lookup for nouns associated to supersenses that tend to have an increased proportion of named entities, e.g. "n.location", "n.group" or "n.person". For the YAGO lookup this correlation is inverted resulting in an increased amount of additionally found supersense bearing nouns for the previously described supersenses.

We also investigate the YAGO feature's ranking component. First we extract a tf-idf index from the DiMSUM training data set. We use the index to generate a ranking for each expression that we are able to detect with our YAGO feature on the test data set. Figure 3 (see Appendix A) shows the relative distribution of gold supersenses for the first five ranking positions. In this evaluation, the correct supersense is present in the first two ranks for the majority of cases. This experiment also indicates how well the detection of supersense bearing nouns performs. E.g. out of 413 nouns in the DiMSUM test set that are marked with the "n.person" supersense, 146 were detected and received a rank.

5.2 GloVe errors

We conduct two experiments to assess our GloVe feature: The first experiment examines the number of tokens for both the train and test set for which there is a GloVe word vector. As described in section 3.4 we use lowercase tokens and match them to their corresponding GloVe word embeddings if possible. With this method we get a coverage of 97.35% for the train set and 94.32% for the test set. Which means that almost every word of the DiMSUM data set is represented by GloVe word embeddings.

To fully assess our GloVe feature we investigate the improvement and deterioration of each tag with our second experiment. To accomplish this we compare the difference between the F1-score of our system and the F1-score of the baseline (F1-score discrepancy) for each tag. Since there are two types of tags - MWE and supersenses - we conduct an analysis for each type. We provide separate evaluations for tokens that have already been seen in the training set (seen) and tokens that have never been seen before (unseen). Firstly we examine the F1-score discrepancy for all supersenses. Figure 4 (see Appendix A) shows the result for all supersenses that occur more than 50 times in the test data set having a F1-score discrepancy that is higher than $|0.5|$. Most supersenses - whether they have been previously seen in the training set or not - improve with the use of the GloVe feature. An exception is "n.event" - a supersense whose F1-score decreases by -8.2% for seen tokens. The average F1-score discrepancy of all noun and verb supersense tags is +0.59; this further confirms our claim that the majority of the supersenses improve with the adoption of the GloVe feature. Lastly we examine the F1-score discrepancy for all MWE tags that occur more than 50 times in the test set. The results can be seen in Figure 5 (see Appendix A). The detection of "I" on seen tokens decreases by 13.2%, whereas the detection of "B" improves for unseen tokens by +8.2%. Overall the GloVe feature has a slightly positive influence on MWE detection.

6 Conclusion

Both YAGO and GloVe are effective in improving the performance for SST and MWE detection. Our experiments show that the GloVe word embeddings provide information in addition to Brown clusters that help the system further distinguish between tags. Unfortunately the benefits of our features don't add up but instead balance each other out. Learning the reason for this could be subject to future work. One could also replace the heuristics we employed for the detection of named entities in the text with more sophisticated named entity resolution techniques. Another possibility would be the comparison of the effect of different word embeddings on the performance. It might also be advantageous to further research the scaling of word embeddings.

References

Baldwin, Timothy, Colin Bannard, Takaaki Tanaka, and Dominic Widdows (2003). "An Empirical Model of Multiword Expression Decomposability". In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*. MWE '03. Sapporo, Japan: Association for Computational Linguistics, pp. 89–96.

Brown, Peter F, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai (1992). "Class-based n-gram models of natural language". In: *Computational linguistics* 18.4, pp. 467–479.

Cettolo, Mauro, Christian Girardi, and Marcello Federico (2012). "Wit3: Web inventory of transcribed and translated talks". In: *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pp. 261–268.

Ciaramita, Massimiliano and Yasemin Altun (2006). "Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 594–602.

Ciaramita, Massimiliano and Mark Johnson (2003). "Supersense tagging of unknown nouns in WordNet". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pp. 168–175.

Collins, Michael (2002). "Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 1–8.

Curran, James R (2005). "Supersense tagging of unknown nouns using semantic similarity". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 26–33.

Fellbaum, Christiane (1990). "English verbs as a semantic net". In: *International Journal of Lexicography* 3.4, pp. 278–301.

– (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Hartmann, Silvana, György Szarvas, and Iryna Gurevych (2012). "Mining multiword terms from Wikipedia". In: *Semi-Automatic Ontology Development: Processes and Resources*, pp. 226–258.

Hovy, Dirk, Anders Johannsen, and Anders Søgaard (2015). "User review sites as a resource for large-scale sociolinguistic studies". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 452–461.

Johannsen, Anders, Dirk Hovy, Héctor Martinez, Barbara Plank, and Anders Søgaard (2014). "More or less supervised super-sense tagging of Twitter". In: *The 3rd Joint Conference on Lexical and Computational Semantics (*SEM)*. Dublin, Ireland.

Kong, Lingpeng, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A Smith (2014). "A dependency parser for tweets". In: *Proceedings of Conference on Empirical Methods In Natural Language Processing (EMNLP)*, pp. 1001–1012.

Koo, Terry and Michael Collins (2005). "Hidden-variable models for discriminative reranking". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 507–514.

Miller, George A (1990). "Nouns in WordNet: a lexical inheritance system". In: *International journal of Lexicography* 3.4, pp. 245–264.

Miller, George A, Claudia Leacock, Randee Tengi, and Ross T Bunker (1993). "A semantic concordance". In: *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, pp. 303–308.

Neubig, Graham, Katsuhito Sudoh, Yusuke Oda, Kevin Duh, Hajime Tsukada, and Masaaki Nagata (2014). "The NAIST-NTT Ted Talk Treebank". In: *International Workshop on Spoken Language Translation*.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "GloVe: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14, pp. 1532–1543.

Sag, Ivan A, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger (2002). "Multiword expressions: A pain in the neck for NLP". In: *Computational Linguistics and Intelligent Text Processing*. Springer, pp. 1–15.

Schneider, Nathan, Emily Danchik, Chris Dyer, and Noah A Smith (2014). "Discriminative lexical semantic segmentation with gaps: running the MWE gamut". In: *Transactions of the Association for Computational Linguistics* 2, pp. 193–206.

Schneider, Nathan, Dirk Hovy, Anders Johannsen, and Marine Carpuat (2016). "SemEval 2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM)". In: *Proc. of SemEval*. San Diego, California, USA.

Schneider, Nathan, Behrang Mohit, Chris Dyer, Kemal Olflazer, and Noah A Smith (2013). "Supersense tagging for Arabic: the MT-in-the-middle attack". In: Association for Computational Linguistics.

Schneider, Nathan, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T Mordowanec, Henrietta Conrad, and Noah A Smith (2014). "Comprehensive annotation of multiword expressions in a social web corpus". In: *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Schneider, Nathan and Noah A Smith (2015). "A corpus and model integrating multiword expressions and supersenses". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1537–1547.

Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 697–706.

Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). "Word representations: a simple and general method for semi-supervised learning". In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pp. 384–394.

Appendix A

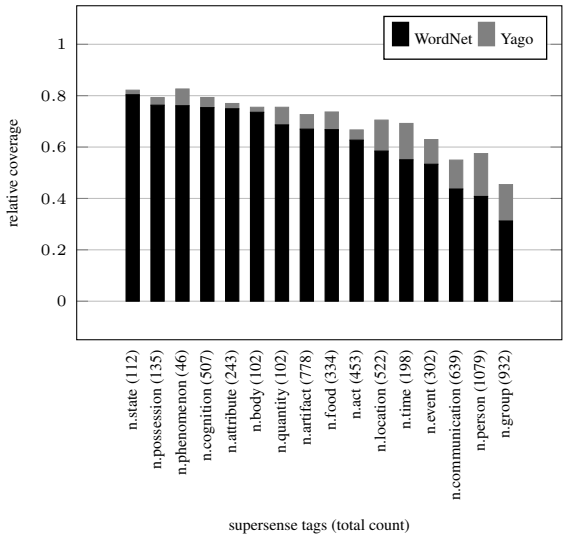

Figure 2: Additional nouns from the DiMSUM dataset found with the YAGO feature.

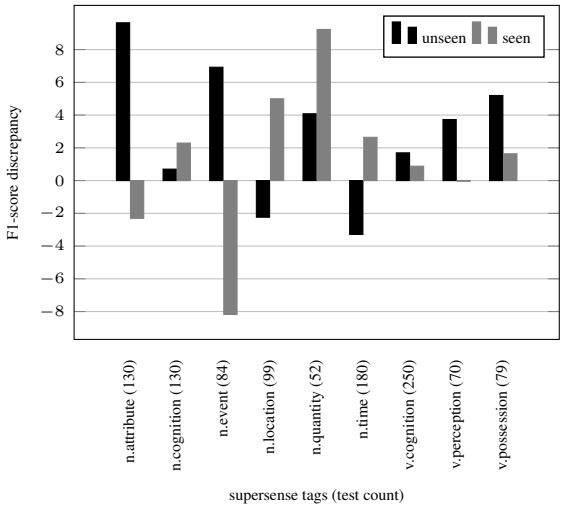

Figure 4: F1-score discrepancy higher than $|0.5|$ between baseline and best system using GloVe feature for supersense tags that occur more than 50 times in the test set distinguishing between seen and unseen tokens.

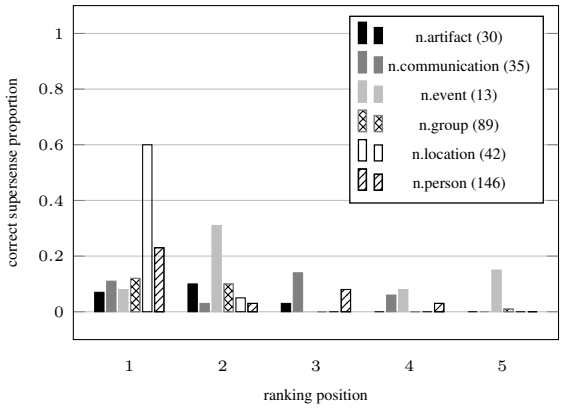

Figure 3: Normalized distribution of gold supersenses over first five ranks on test set.

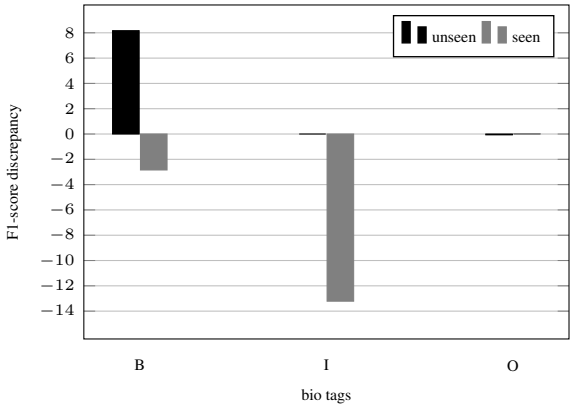

Figure 5: F1-score discrepancy between baseline and best system using GloVe feature for BIO tags that occur more than 50 times in the test set distinguishing between seen and unseen tokens.