Simple Embedding-Based
Word Sense Disambiguation

Dieke Oele
Rijksuniversiteit Groningen,
Groningen,
`d.oele@rug.nl`

Gertjan van Noord
Rijksuniversiteit Groningen,
Groningen,
`g.j.m.van.noord@rug.nl`

Abstract

We present a simple knowledge-based WSD method that uses word and sense embeddings to compute the similarity between the gloss of a sense and the context of the word. Our method is inspired by the Lesk algorithm as it exploits both the context of the words and the definitions of the senses. It only requires large unlabeled corpora and a sense inventory such as WordNet, and therefore does not rely on annotated data. We explore whether additional extensions to Lesk are compatible with our method. The results of our experiments show that by lexically extending the amount of words in the gloss and context, although it works well for other implementations of Lesk, harms our method. Using a lexical selection method on the context words, on the other hand, improves it. The combination of our method with lexical selection enables our method to outperform state-of the art knowledge-based systems.

1 Introduction

The quest of automatically finding the correct meaning of a word in context, also known as Word Sense Disambiguation (WSD), is an important topic in Natural Language Processing (NLP). WSD systems that are based on supervised learning methods gain best results (Snyder and Palmer, 2004; Pradhan et al., 2007; Navigli and Lapata, 2007; Navigli, 2009; Zhong and Ng, 2010). However, they require a large amount of manually annotated data for training. Also, even if such a supervised system obtains good results in a certain domain, it is not readily portable to other domains (Escudero et al., 2000).

As an alternative to supervised systems, knowledge-based systems do not require manually tagged data and have proven to be applicable to new domains (Agirre et al., 2009). An example of such a system is the Lesk algorithm (Lesk, 1986) that exploits the idea that the overlap between the definition of a word and the definitions of the words in its context can provide information about its meaning. It only requires two types of information: a set of dictionary entries with definitions (hereafter referred to as glosses) for each possible word meaning, and the context in which the word occurs. A popular variant of the algorithm is the "simplified" Lesk algorithm (Kilgarriff and Rosenzweig, 2000), which disambiguates one word at a time by comparing each of its glosses to the context in which the word is found. This variant avoids the combinatorial explosion of word sense combinations the original version suffers from when trying to disambiguate multiple words in a text.

A problem with the aforementioned method, however, is that, when a gloss is matched against the context of a word, in most cases the lexical overlap is very small. As a solution to this problem, we use a WSD-method that, instead of counting the number of words that overlap, takes embeddings as input to compute the similarity between the gloss of a sense and the context of the word. Although our method works well on its own, its simplicity allows us to explore whether other extensions to the Lesk algorithm that have proven to be successful can improve it further.

As both the Lesk algorithm and our extension rely on the definition of the words and the words that surround it, it is interesting to see whether adapting both sources of information would improve either of them. In this light, there are two possibilities: expansion or reduction. For the first option, the existing words of the context and glosses can be expanded with additional words that have similar meanings. For example, Miller

et al. (2012) use a distributional thesaurus, that is computed from a large parsed corpus to lexically expand the context and sense information. They show that, using these expanded context and glosses, improves two variants of Lesk. When reducing the amount of words in either the context or the target words' sense, methods are required that prohibit the loss of informative words. Vasilescu et al. (2004) shows that a pre-selection of words in the context of the target word improves Simplified Lesk. In this paper we describe experiments where both methods are used in combination with out method that is based on word- and sense embeddings.

2 Related work

In the past few years, much progress has been made on learning word embeddings from unlabeled data that represent the meanings of words as contextual feature vectors. A major advantage of these word embeddings is that they exhibit certain algebraic relations and can, therefore, be used for meaningful semantic operations such as computing word similarity (Turney, 2006), and capturing lexical relationships (Mikolov et al., 2013b).

A disadvantage of word embeddings is that they assign a single embedding to each word, thus ignoring the possibility that words may have more than one meaning. This problem can be addressed by associating each word with a number of sense-specific embeddings. For this, several methods have been proposed in recent work. For example, in Reisinger and Mooney (2010) and Huang et al. (2012), a fixed number of senses is learned for each word that has multiple meanings by first clustering the contexts of each token, and subsequently relabeling each word token with the clustered sense before learning embeddings.

Although such sense embedding methods have demonstrated good performance, they use automatically induced senses. They are, therefore, not readily applicable for applications that rely on WordNet-based senses, such as machine translation and information retrieval and extraction systems (see Morato et al. (2004) for examples of such systems). Recently, features based on sense-specific embeddings learned using a combination of large corpora and a sense inventory have been shown to achieve state-of-the-art results for supervised WSD Rothe and Schütze (2015; Jauhar et al. (2015; Taghipour and Ng (2015).

Our system makes use of a combination of sense embeddings, context embeddings, and gloss embeddings. Similar approaches have been proposed by Chen et al. (2014) and Pelevina et al. (2016). The main difference to our approach is that they automatically induce sense embeddings and find the best sense by comparing them to context embeddings, while we add gloss embeddings for better performance. Inkpen and Hirst (2003) apply gloss- and context vectors to the disambiguation of near-synonyms in dictionary entries. Also Basile et al. (2014) use a distributional approach to representing definitions and the context of the target word. They create semantic vectors for glosses and contexts to compute similarity of the gloss and the context of a target word, while we also compute the similarity of a sense and its context directly using sense embeddings.

3 Lesk++

Our WSD algorithm takes sentences as input and outputs a preferred sense for each polysemous word. Given a sentence $w_1 \ldots w_i$ of i words, we retrieve a set of word senses from the sense inventory for each word w. Then, for each sense s of each word w, we consider the similarity of its lexeme (the combination of a word and one of its senses (Rothe and Schütze, 2015)) with the context and the similarity of the gloss with the context.

For each potential sense s of word w, the cosine similarity is computed between its gloss vector G_s and its context vector C_w and between the context vector C_w and the lexeme vector $L_{s,w}$. The score of a given word w and sense s is thus defined as follows:

$$\text{Score}(s,w) = cos(G_s, C_w) + cos(L_{s,w}, C_w) \quad (1)$$

The sense with the highest score is chosen. When no gloss is found for a given sense, only the second part of the equation is used.

Prior to the disambiguation itself, we sort the words by the number of senses it has, in order that the word with the fewest senses will be considered first. The idea behind this is that words that have fewer senses are easier to disambiguate (Chen et al., 2014). The algorithm relies on the words in the context which may themselves be ambiguous. If words in the context have been disambiguated already, this information can be used for the ambiguous words that follow. We, therefore, use the

resulting sense of each word for the disambiguation of the following words starting with the "easiest" words.

Our method requires lexeme embeddings $L_{s,w}$ for each sense s. For this, we use AutoExtend (Rothe and Schütze, 2015) to create additional embeddings for senses from WordNet on the basis of word embeddings. AutoExtend is an auto-encoder that relies on the relations present in WordNet to learn embeddings for senses and lexemes. To create these embeddings, a neural network containing lexemes and sense layers is built, while the WordNet relations are used to create links between each layer. The advantage of their method is that it is flexible: it can take any set of word embeddings and any lexical database as input and produces embeddings of senses and lexemes, without requiring any extra training data.

For each word w we need a vector for the context C_w, and for each sense s of word w we need a gloss vector G_s. The context vector C_w is defined as the mean of all the content word representations in the sentence: if a word in the context has already been disambiguated, we use the corresponding sense embedding; otherwise, we use the word embedding. For each sense s, we take its gloss as provided in WordNet. In line with Banerjee and Pedersen (2002), we expand this gloss with the glosses of related meanings, excluding antonyms. Similar to the creation of the context vectors, the gloss vector G_s is created by averaging the word embeddings of all the content words in the gloss.

4 Lexical expansion and lexical selection

We use the method of Miller et al. (2012) to expand the glosses and the contexts of the target words before using our adaptation of the Lesk system.[1] For each content word we retrieve the 30 most similar terms from the distributional thesaurus and add them to the context or gloss while occurrences of the target word are removed.

For the selection of context words, we use the lexical chaining technique as applied in Vasilescu et al. (2004) that use the idea of creating lexical chains from Hirst and St-Onge (1998). Lexical chains are sequences of words that are semantically related. Similar to Vasilescu et al. (2004), we use the synonymy and hypernymy relations in

WordNet in combination with a similarity measure (Jaccard formula (Manning and Schütze, 1999)), to verify whether a context word is a member of such a lexical chain. For both the target word w and each context word c in its context, we retrieve a set of sense definitions of all the synonyms and hypernyms of w according to the WordNet hierarchy. A context word is added to the context if the similarity score for the set of w and the set of c is greater than an experimental threshold.

5 Experiments

We test our method on both Dutch and English data. We build 300-dimensional word embeddings on the Dutch Sonar corpus (Oostdijk et al., 2013) using word2vec CBOW (Mikolov et al., 2013a), and create sense- and lexeme embeddings with AutoExtend. For English, we use the embeddings from Rothe and Schütze (2015)[2]. They lie within the same vector space as the pre-trained word embeddings by Mikolov et al. (2013a)[3], trained on part of the Google News dataset, which contains about 100 billion words. This model (similar to the Dutch model) contains 300-dimensional vectors for 3 million words and phrases.

Our sense inventory for Dutch is Cornetto (Vossen et al., 2012) and for English, we use WordNet 1.7.1 (Fellbaum, 1998) as this version matches the AutoExtend embeddings. In Cornetto, 51.0% of the senses have glosses. In the Princeton WordNet, almost all of them do. The DutchSemCor corpus (Vossen et al., 2013b) is used for Dutch evaluation and, for English, we use SemCor (Fellbaum, 1998). A random subset of 5000 manually annotated sentences from each corpus was created. Additionally, we test on the Senseval-2 (SE-2) and Senseval-3 (SE-3) all-words datasets (Snyder and Palmer, 2004; Palmer et al., 2001).

We evaluate our method by comparing it with a random baseline and Simplified Lesk with expanded glosses (SE-Lesk) (Kilgarriff and Rosenzweig, 2000; Banerjee and Pedersen, 2002). We do not compare our system to the initial results of AutoExtend (Rothe and Schütze, 2015) as they tested it in a supervised setup using sense embeddings as features. However, as is customary in WSD evaluation, we do compare our system to the most frequent sense baseline, which is notoriously

---

[1] We use the distributional thesaurus downloaded from www.lt.informatik.tu-darmstadt.de/de/data/distributional-thesauri.

[2] http://www.cis.lmu.de/ sascha/AutoExtend/
[3] https://code.google.com/p/word2vec/

	DSC	SC	SE-2	SE-3		DSC	SC	SE-2	SE-3
SE-Lesk	28.1%	53.2	52.1%	50.1%	Lesk++	45.9%	55.1%	54.9%	59.3%
+LE	**29.6%**	**56.5%**	**51.0%**	**49.3%**	+LE	42.5%	47.8%	43.8%	46.2%
+LS	16.0%	40.7%	48.1%	54.3%	+LS	**47.3%**	**67.2%**	**58.4%**	**59.4%**
+LE,LS	25.2%	40.6%	46.2%	46.0%	+LE,LS	41.0%	66.9%	49.1%	43.5%

Table 1: Results for DutchSemCor (DSC), SemCor (SC), Senseval-2 (SE-2) and Senseval3 (SE-3) for Simplified Extended Lesk (SE-Lesk) and Lesk++. The following columns use lexical selection (LS), lexical extension (LE) and both extension and selection (LE,LS).

difficult to beat due to the highly skewed distribution of word senses (Agirre and Edmonds, 2007). As this baseline relies on manually annotated data, which our system aims to avoid, we consider this baseline to be semi-supervised.

Additionally, we compare our system to a state-of-the-art knowledge-based WSD system, UKB (Agirre and Soroa, 2009), that, similar to our method, does not require any manually tagged data. UKB can be used for graph-based WSD using a pre-existing knowledge base. It applies random walks, e.g. Personalized PageRank, on the Knowledge Base graph to rank the vertices according to the context. We use UKBs Personalized PageRank method word-by-word with WordNet 1.7 and eXtended WordNet for English, as this setup yielded the best results in Agirre and Soroa (2009). For Senseval-2 (SE-2) and Senseval-3 we use the WSD evaluation framework of Raganato et al. (2017), which provides evaluation datasets and output of other knowledge-based WSD systems. From those systems we report on the Extended Lesk version of Basile et al. (2014), (DSM)[4] which is most similar to our approach.

The manually annotated part of DutchSemCor is balanced *per sense* which means that an equal number of examples for each sense is annotated. It is therefore not a reliable source for computing the most frequent sense. Alternatively, similar to Vossen et al. (2013a), we derive sense frequencies by using the automatically annotated counts in DutchSemCor[5]. The most frequent sense baseline for Dutch is, therefore, lower as compared to the English, where the most frequent sense of a word is fully based on manual annotation.

6 Results

Table 1 shows the results of both SE-Lesk and our method (Lesk++) with lexically extended (LE) and selected (LS) context and gloss vectors. The use of word and sense embeddings yields overall better results as compared to SE-Lesk. Remarkably, lexical extension, that is very beneficial for SE-Lesk, does serious harm to our method. Selecting words in the context, on the other hand, improves our method and makes SE-Lesk perform worse.

Table 2 shows the results of the best performing combinations, SE-Lesk with lexical extension and Lesk++ with lexical selection, compared to three baselines. Our system, when used in combination with the lexical selection method, performs better than the other purely knowledge-based methods.

	DSC	SC	SE-2	SE-3
Lesk++LS	**47.3%**	**67.2%**	**58.4%**	**59.4%**
SE-Lesk,LE	29.6%	56.5%	51.0%	49.3%
UKB	38.9%	57.6%	56.0%	51.8%
DSM	-	-	51.2%	42.3%
Random	26.5%	33.6%	39.9%	34.9%
MFS	*36.0%*	*70.9%*	*65.6%*	*66.2%*

Table 2: Results for Simplified Extended Lesk (SE-Lesk) with lexical extension (LE) and Lesk++ with lexical selection (LS), UKB, DSM, a random and a most frequent sense baseline

7 Discussion

The difference in results for Dutch and English can be explained by the coverage of the datasets. The Cornetto coverage is about 60%, compared to Princeton Wordnet, with an average polysemy of 1.07 for nouns, 1.56 for verbs and 1.05 for adjectives while, for English it is 1.24 for nouns, 2.17 for verbs and 1.40 for adjectives. Also, not all Dutch senses have corresponding glosses while

---

[4] We use https://github.com/pippokill/lesk-wsd-dsm without sense frequency for comparability

[5] In DutchSemCor senses are annotated with an SVM, trained on the manually annotated part of the corpus, see Vossen et al. (2013a) for more details.

most of the English ones do. As our method relies greatly on gloss vectors, this could affect its performance.

The different performance of both extensions to SE-Lesk and Lesk++ shows that both algorithms capture different types of information and therefore require a different type of input. As SE-Lesk counts on the direct overlap of words, it depends highly on a larger amount of words. Lesk++ on the other hand, already overcomes this problem and clearly benefits from more "quality" information in the contexts.

In future work we would like to try other vector types such as Melamud et al. (2016) that represents contexts that outperform the context representation of averaged word embeddings. Also, it would be nice to see whether other Knowledge-based sense embeddings, such as the ones from Camacho-Collados et al. (2016), could improve our results.

8 Conclusions

We compared several extensions to the Lesk algorithm with an adaption which uses sense, gloss and context embeddings to compute the similarity of word senses to the context in which the words occur. We try two different methods that could improve ours, one that further extends the information in both the context and the glosses by utilizing Distributional thesauri (Miller et al., 2012) and one that pre-selects context words using the WordNet hierarchy (Vasilescu et al., 2004). Although our approach is a straightforward extension to the Lesk algorithm, it achieves better performance compared to Lesk and a random baseline. When using a selection scheme before creating context vectors, its performance is better than our knowledge based baselines. The main advantage of our method is its simplicity which makes it fast and easy to apply to other languages. It furthermore only requires unlabeled text and the definitions of senses, and does not rely on any manually annotated data, which makes our system an attractive alternative for supervised WSD.

References

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 33–41.

Eneko Agirre, Oier Lopez De Lacalle, and Aitor Soroa. 2009. Knowledge-based WSD on specific domains: Performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1501–1506.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136–145.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1591–1600.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artif. Intell.*, 240:36–64.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1025–1035.

Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 172–180.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 873–882.

Diana Zaiu Inkpen and Graeme Hirst. 2003. Automatic sense disambiguation of the near-synonyms in a dictionary entry. In *Computational Linguistics and Intelligent Text Processing, 4th International Conference*, pages 258–267.

Sujay Kumar Jauhar, Chris Dyer, and Eduard H. Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 683–693.

Adam Kilgarriff and Joseph Rosenzweig. 2000. English senseval: report and results. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece, May. European Language Resources Association (ELRA).

Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 51–61.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pages 746–751.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING 2012*, pages 1781–1796.

Jorge Morato, Miguel Ngel Marzal, Juan Llorns, and Jos Moreiro. 2004. Wordnet applications. In *Proceeding of the Second Global Wordnet Conference*.

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1683–1688.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, February.

Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman, 2013. *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*, pages 219–247. Springer Berlin Heidelberg, Berlin, Heidelberg.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24.

Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110. Association for Computational Linguistics.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1793–1803.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, July.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–323, May–June.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics, Volume 32, Number 3, September 2006*.

Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. 2004. Evaluating variants of the Lesk approach for disambiguating words. In *Proceedings*

of the Fourth International Conference on Language Resources and Evaluation.

Piek Vossen, Attila Görög, Rubén Izquierdo, and Antal van den Bosch. 2012. Dutchsemcor: Targeting the ideal sense-tagged corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pages 584–589, may.

Piek Vossen, Rubén Izquierdo, and Attila Görög. 2013a. Dutchsemcor: in quest of the ideal sense-tagged corpus. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov, editors, *Recent Advances in Natural Language Processing*, pages 710–718.

Piek Vossen, Isa Maks, Roxane Segers, Hennie der van Vliet, Marie-Francine Moens, Katja Hofmann, Erik Tjong Kim Sang, and Maarten de Rijke, 2013b. *Cornetto: A Combinatorial Lexical Semantic Database for Dutch*, pages 165–184. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 78–83.