Supervised Sense Tagging using Support Vector Machines

Clara Cabezas, Philip Resnik, and **Jessica Stevens**
Dept. of Linguistics and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742 USA
{clarac,resnik,stevenjc}@umiacs.umd.edu

Abstract

We describe the University of Maryland's supervised sense tagger, which participated in the SENSEVAL-2 lexical sample evaluations for English, Spanish, and Swedish; we also present unofficial results for Basque. We designed a highly modular combination of language-independent feature extraction and supervised learning using support vector machines in order to permit rapid ramp-up, language independence, and capability for future expansion.

1 Introduction

The SENSEVAL-2 exercise provided an unprecedented opportunity to explore word sense disambiguation (WSD) in a common evaluation framework for a large number of languages. In past work, we have focused on unsupervised methods for English, taking advantage of the WordNet hierarchy and sometimes also selectional preferences between predicates and arguments (Resnik, 1997; Resnik, 1999). In the current exercise, however, WordNet-like sense hierarchies were not necessarily going to be available for all languages, and the predominance of lexical selection tasks (rather than all-words tasks) suggested adopting a disambiguation approach capable of exploiting manually annotated training data. These considerations motivated a system design based on supervised learning, where senses to be predicted did not need to be treated as part of a semantic hierarchy.

Our design was also motivated by the role of semantic selection techniques in our longer term research agenda. In the context of our group's work on cross-language information retrieval and machine translation applications (Resnik et al., 2001; Cabezas et al., 2001), lexical selection — that is, choosing the right target-language word given a source-language word in context — is a crucial task. Because the lexical selection problem is extremely similar to sense selection, and because this was our first foray into supervised methods, we took advantage of the opportunity to construct an architecture that will support both tasks.

In the sections that follow, we lay out our system architecture, briefly summarize our SENSEVAL-2 results, and discuss our plans for future work.

2 System Architecture

UMD's system follows the classic supervised learning paradigm that, for WSD, is perhaps best exemplified by Yarowsky's (1993) work. Each word in the vocabulary is considered an independent classification problem. First, annotated training instances for the ambiguous word are analyzed so that each instance can be represented as a collection of feature-value pairs labeled with the correct category. Then, these data are used for parameter estimation within a supervised learning framework in order to produce a trained classifier. Finally, the trained classifier is given previously unseen test instances and for each instance it predicts what the appropriate category label should be.

2.1 Contextual Features

We began by tokenizing all the training instances using a simple language-specific tokenizer. Features were then defined in terms of the presence of tokens either within a wide context or at a certain position to the right or left of the word being disambiguated.

In detail, let $\mathcal{T}$ be the set of unique tokens found in the full set of training data (all training instances), plus the special token UNKNOWN, which replaces any token in test data that was never seen during training. Define $\mathcal{F}_{\text{wide}} = \mathcal{T}$. A feature $f \in \mathcal{F}_{\text{wide}}$ will be considered present and have a non-zero value if f appears anywhere in the wide context of the word being disambiguated. For example, if we were disambiguating the word *training* that appears in the first sentence of this paragraph, using the entire paragraph as the wide context, then there would be non-zero values for features WE, BEGAN, and every other word in the paragraph. That is, features correspond to surrounding words.[1]

Let $\mathcal{L} = \{L_3, L_2, L_1, R_1, R_2, R_3\}$, signifying the locations "three tokens to the left", "two tokens to the left", ..., "three tokens to the right", and define $\mathcal{F}_{\text{colloc}} = \{\text{l:t} \mid l \in \mathcal{L} \text{ and } t \in \mathcal{T}\}$. A feature l:t $\in \mathcal{F}_{\text{colloc}}$ will be considered present and have a non-zero value if token t appears at position l relative to the word being disambiguated. For example, if we were disambiguating the word *training* that appears in the first sentence of this section, there would be non-zero values for the features L_3:`tokenizing`, L_2:`all`, L_1:`the`, L_1:`instances`, L_2:`using`, and L_3:`a`.

2.2 Feature Weights

The value associated with each feature is a weight indicating how useful the feature is likely to be in disambiguation, analogous to the term weights used in representing documents as feature vectors for information retrieval.

In detail, let us designate the full feature set as $\mathcal{F} = \mathcal{F}_{\text{wide}} \cup \mathcal{F}_{\text{colloc}}$, and let $N_{\mathcal{F}} = |\mathcal{F}|$. Clearly some features are more useful than others. For example, the feature `into` (word *into* appearing anywhere in the context) is unlikely to help distinguish among senses, although the feature R_1:`into` (word *into* appearing one word to the right) might be useful for disambiguating among the senses of some verbs. In order to assign weights to features based on their likely utility, we follow a strategy similar to what is done in information retrieval, defining inverse category frequency (ICF), by analogy with inverse document frequency (IDF), as a function of how many distinct categories a feature appears with in training data.

Specifically, if we are disambiguating a word w with senses $\mathcal{S} = \{s_1, s_2, \ldots, s_{N_w}\}$, then we define $\text{ICF}_w(f) = -\log(N_w^f/N_w)$ where N_w^f is the number of distinct elements of $\mathcal{S}$ that ever cooccur with feature f in the training data for word w. For example, if a word has five senses, and the feature L_1:`the` appears in some training instance for each of the five senses, then $\text{ICF}_w(L_1\texttt{:the}) = -\log(5/5) = 0$, correctly indicating that this feature is not at all useful for disambiguating among the five senses of this word. The lower N_w^f is, the greater the value of the $\text{ICF}_w(f)$ value and hence the greater weight accorded this feature.

Training and test instances are represented as $N_{\mathcal{F}}$-ary feature vectors: given a training or test instance for a word w, the vector representation is defined by $v_w[f] = \text{ICF}_w(f)$ if $f \in \mathcal{F}$ is present, and zero otherwise.

2.3 Learning Framework

Once training and test instances are represented as feature vectors, it becomes possible to exploit any number of existing supervised learning algorithms. In general, such algorithms take a set $\{\langle v_1, c_1\rangle, \langle v_2, c_2\rangle, \ldots, \langle v_N, c_N\rangle\}$ of training instances, and produce a classifier that takes a feature vector v as input and return a distribution or confidence function over the possible categories.

For SENSEVAL-2, we selected support vector machines (SVMs) as the supervised learning framework. We were motivated by the fact that SVMs have been shown to achieve high performance and work efficiently in environments where there are very large numbers of features, and also by the existence of a good off-the-shelf implementation, SVM-Light, available for research purposes (Joachims, 1999; Joachims, 1998).[2]

SVM learning is appropriate for binary classification tasks, rather than the multi-way classification needed for disambiguating among n senses. For each word in the lexical sample tasks, therefore, we constructed a family of SVM classifiers, one for each of the word's N_w senses. All positive training examples for a

[1] For SENSEVAL-2, we defined the surrounding context for wide contexts as being anywhere within the test instance, because instances comprised only a sentence or two. In a more general setting the context could be defined as a window of ±50 words, ±100 words, the entire document, etc.

[2] Hearst (1998) presents a collection of brief and illuminating discussions of SVMs; see http://www.computer.org/intelligent/ex1998/pdf/x4018.pdf. SVM-Light is available at http://www-ai.cs.uni-dortmund.de/svm_light.

Language	Precision (%)	Recall (%)
English (coarse)	64.3	64.3
English (fine)	56.8	56.8
Spanish (fine)	62.7	62.7
Swedish (mixed)	65.6	65.6
Swedish (fine)	61.1	61.1
Basque (fine)	70.3	70.3

Table 1: UMD-SST lexical sample results

sense s_i of w were treated as negative training examples for all the other senses s_j, $j \neq i$.

In the testing phase, we convert test instances for word w into feature vectors, and we then we run these vectors through the SVM classifiers for $\{s_1, s_2, \ldots, s_{N_w}\}$. For each instance, we select the sense for which the SVM classifier's response is most strongly "yes" (or, equivalently, most weakly "no").

3 SENSEVAL-2 Results

Table 1 shows the performance of UMD's supervised sense tagger (UMD-SST) for the lexical sample tasks in four languages. The figures for English, Spanish, and Swedish are official SENSEVAL-2 results; the figures for Basque are unofficial results kindly computed by the Basque task organizers after SENSEVAL-2 because our Basque responses were not submitted in time for official evaluation.

In general, we were quite pleased with the results, particularly since this was our first time participating in SENSEVAL. UMD-SST turned in a solid performance in comparison with the baselines and other systems, with essentially no language-specific alterations necessary other than those required for tokenization. This enabled us to participate in system evaluation for more languages than any site except JHU. We consider this a good starting point for our further investigations, which we now briefly describe.

4 Future Work

Using the current system as a starting point, we are engaged in three lines of further investigation: linguistically richer contextual features, corpus-dependent expansion of feature vectors, and lexical selection via supervised learning.

In our preliminary tests using training and development data, we experimented first with using $\mathcal{F}_{\text{wide}}$ as the feature set, and obtained significant improvements when we added $\mathcal{F}_{\text{colloc}}$ in order to capture collocations and other local contextual features. In our follow-up efforts we plan to use broad-coverage parsing to create a set of features augmented further by grammatical relations, thus capturing collocations mediated by syntactic structure. For example, although our current feature vectors could not represent the presence of the word *tagger* as a nearby collocate of the word *describe* in the abstract of this paper, syntactically richer representations of this context for the verb *describe* would include the feature `object='tagger'`. Use of syntactic collocates will require broad-coverage parsing in all the languages of interest in order to identify grammatical relations; for this we will take advantage of our other work at Maryland on bootstrapping stochastic parsers for new languages using parallel corpora (Cabezas et al., 2001).

In our preliminary efforts we were not surprised to find that sparseness of data was a problem. Although we expect that some improvements may be obtained by collapsing across word variants — e.g. via morphological equivalence classes or stemming — we also plan to focus our efforts on semantic expansion, using document expansion techniques we have developed in our research on cross-language information retrieval (Levow et al., 2001). We have implemented a variant of the architecture in which training contexts are used as queries to a comparable corpus in order to retrieve related documents. The features from these documents are then added to the context representations, providing semantically enhanced feature vectors. Evaluation of this approach using SENSEVAL data is in progress.

Our third avenue of investigation focuses on the use of our supervised WSD infrastructure to address problems of lexical selection in machine translation. Empirically, there is a close relationship between sense distinctions and patterns of lexicalization across languages (Resnik and Yarowsky, 1999). And operationally, there is no real difference between labeling a word with a sense tag from a monolingual dictionary and labeling that word with a translation from a bilingual dictionary. Using WSD techniques for lexical selection primarily requires solving two problems. The first problem is acquisition of annotated training data, and in this case large corpora of translation-labeled words in context can be created by obtaining parallel corpora, performing word-level alignment, and labeling each word with its correspondent in the other language; this problem is already solved as part of our infrastructure for research on statistical machine translation (Cabezas et al., 2001). The second problem is one of scalability: the approach we have described requires a separate classifer for every sense (or, now, every possible word-level translation) of every source language word. This remains an open issue, but we are optimistic about rapid developments in this area since scaling up to large vocabularies is a problem shared by everybody who wishes to use supervised WSD techniques in a broad-coverage setting.

5 Conclusions

University of Maryland's sense tagger represents a classic instance of the supervised learning approach. At the same time, we have made architectural choices that promote language independence, modularity, extensibility, and scalability, and in a relatively short time period we succeeded in putting together an implementation that performs quite credibly among an impressive collection of competitors. We are encouraged by the results and we look forward to participating in further SENSEVAL exercises.

Acknowledgements

This work was supported in part by Department of Defense contract MDA90496C1250 and DARPA/ITO Cooperative Agreement N660010028910. We're very grateful to all the SENSEVAL-2 organizers and task organizers for their hard work, to Thorsten Joachims for making SVM-Light available, and to David Martinez for computing our results for Basque.

References

Clara Cabezas, Bonnie Dorr, and Philip Resnik. 2001. Spanish language processing at University of Maryland: Building infrastructure for multilingual applications. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*, Jaen, Spain, September.

Marti A. Hearst. 1998. Trends and controversies: Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*. Springer.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press.

Gina-Anne Levow, Douglas Oard, and Philip Resnik. 2001. Rapidly retargetable interactive translingual retrieval. In *Human Language Technology Conference (HLT-2001)*, San Diego, CA, March.

Philip Resnik and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

Philip Resnik, Douglas Oard, and Gina Levow. 2001. Improved cross-language retrieval using backoff translation. In *Human Language Technology Conference (HLT-2001)*, San Diego, March.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *ANLP Workshop on Tagging Text with Lexical Semantics*, Washington, D.C., April.

Philip Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR)*, 11:95–130.

David Yarowsky. 1993. One sense per collocation. ARPA Workshop on Human Language Technology, March. Princeton.