Using Lexical Dependency and Ontological Knowledge to Improve a Detailed Syntactic and Semantic Tagger of English

Andrew Finch
NiCT[*]-ATR[†]
Kyoto, Japan
andrew.finch
@atr.jp

Ezra Black
Epimenides Corp.
New York, USA
ezra.black
@epimenides.com

Young-Sook Hwang
ETRI
Seoul, Korea
yshwang7
@etri.re.kr

Eiichiro Sumita
NiCT-ATR
Kyoto, Japan
eiichiro.sumita
@atr.jp

Abstract

This paper presents a detailed study of the integration of knowledge from both dependency parses and hierarchical word ontologies into a maximum-entropy-based tagging model that simultaneously labels words with both syntax and semantics. Our findings show that information from both these sources can lead to strong improvements in overall system accuracy: dependency knowledge improved performance over all classes of word, and knowledge of the position of a word in an ontological hierarchy increased accuracy for words not seen in the training data. The resulting tagger offers the highest reported tagging accuracy on this tagset to date.

1 Introduction

Part-of-speech (POS) tagging has been one of the fundamental areas of research in natural language processing for many years. Most of the prior research has focussed on the task of labeling text with tags that reflect the words' syntactic role in the sentence. In parallel to this, the task of word sense disambiguation (WSD), the process of deciding in which semantic sense the word is being used, has been actively researched. This paper addresses a combination of these two fields, that is: labeling running words with tags that comprise, in addition to their syntactic function, a broad semantic class that signifies the semantics of the word in the context of the sentence, but does not necessarily provide information that is sufficiently fine-grained as to disambiguate its sense. This differs from what is commonly meant by WSD in that although each word may have many "senses" (by senses here, we mean the set of semantic labels the word may take), these senses are not specific to the word itself but are drawn from a vocabulary applicable to the subset of all types in the corpus that may have the same semantics.

In order to perform this task, we draw on research from several related fields, and exploit publicly available linguistic resources, namely the WordNet database (Fellbaum, 1998). Our aim is to simultaneously disambiguate the semantics of the words being tagged while tagging their POS syntax. We treat the task as fundamentally a POS tagging task, with a larger, more ambiguous tag set. However, as we will show later, the 'n-gram' feature set traditionally employed to perform POS tagging, while basically competent, is not up to this challenge, and needs to be augmented by features specifically targeted at semantic disambiguation.

2 Related Work

Our work is a synthesis of POS tagging and WSD, and as such, research from both these fields is directly relevant here.

The basic engine used to perform the tagging in these experiments is a direct descendent of the maximum entropy (ME) tagger of (Ratnaparkhi, 1996) which in turn is related to the taggers of (Kupiec, 1992) and (Merialdo, 1994). The ME approach is well-suited to this kind of labeling because it allows the use of a wide variety of features without the necessity to explicitly model the interactions between them.

The literature on WSD is extensive. For a good overview we direct the reader to (Nancy and Jean, 1998). Typically, the local context around the

[*] National Institute of Information and Communications Technology

[†] ATR Spoken Language Communication Research Labs

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 215–222,
Sydney, July 2006. ©2006 Association for Computational Linguistics

word to be sense-tagged is used to disambiguate the sense (Yarowsky, 1993), and it is common for linguistic resources such as WordNet (Li et al., 1995; Mihalcea and Moldovan, 1998; Ramakrishnan and Prithviraj, 2004), or bilingual data (Li and Li, 2002) to be employed as well as more long-range context. An ME-system for WSD that operates on similar principles to our system (Suarez, 2002) was based on an array of local features that included the words/POS tags/lemmas occurring in a window of +/-3 words of the word being disambiguated. (Lamjiri et al., 2004) also developed an ME-based system that used a very simple set of features: the article before; the POS before and after; the preposition before and after, and the syntactic category before and after the word being labeled. The features used in both of these approaches resemble those present in the feature set of a standard n-gram tagger, such as the one used as the baseline for the experiments in this paper. The semantic tags we use can be seen as a form of semantic categorization acting in a similar manner to the semantic class of a word in the system of (Lamjiri et al., 2004). The major difference is that with a left-to-right beam-search tagger, labeled context to the right of the word being labeled is not available for use in the feature set.

Although POS tag information has been utilized in WSD techniques (e.g. (Suarez, 2002)), there has been relatively little work addressing the problem of assigning a part-of-speech tag to a word together with its semantics, despite the fact that the tasks involve a similar process of label disambiguation for a word in running text.

3 Experimental Data

The primary corpus used for the experiments presented in this paper is the ATR General English Treebank. This consists of 518,080 words (approximately 20 words per sentence, on average) of text annotated with a detailed semantic and syntactic tagset.

To understand the nature of the task involved in the experiments presented in this paper, one needs some familiarity with the ATR General English Tagset. For detailed presentations, see (Black et al., 1996b; Black et al., 1996a; Black and Finch, 2001). An apercu can be gained, however, from Figure 1, which shows two sample sentences from the ATR Treebank (and originally from a Chinese take–out food flier), tagged with respect to the ATR General English Tagset. Each verb, noun, adjective and adverb in the ATR tagset includes a semantic label, chosen from 42 noun/adjective/adverb categories and 29 verb/verbal categories, some overlap existing between these category sets. Proper nouns, plus certain adjectives and certain numerical expressions, are further categorized via an additional 35 "proper–noun" categories. These semantic categories are intended for any "Standard–American–English" text, in any domain. Sample categories include: "physical.attribute" (nouns/adjectives/adverbs), "alter" (verbs/verbals), "interpersonal.act" (nouns/adjectives/adverbs/verbs/verbals), "orgname" (proper nouns), and "zipcode" (numericals). They were developed by the ATR grammarian and then proven and refined via day–in–day–out tagging for six months at ATR by two human "treebankers", then via four months of tagset–testing–only work at Lancaster University (UK) by five treebankers, with daily interactions among treebankers, and between the treebankers and the ATR grammarian. The semantic categorization is, of course, in addition to an extensive syntactic classification, involving some 165 basic syntactic tags.

The test corpus has been designed specifically to cope with the ambiguity of the tagset. It is possible to correctly assign any one of a number of 'allowable' tags to a word in context. For example, the tag of the word *battle* in the phrase "a legal battle" could be either NN1PROBLEM or NN1INTER-ACT, indicating that the semantics is either a problem, or an inter-personal action. The test corpus consists of 53,367 words sampled from the same domains as, and in approximately the same proportions as the training data, and labeled with a set of up to 6 allowable tags for each word. During testing, only if the predicted tag fails to match any of the allowed tags is it considered an error.

4 Tagging Model

4.1 ME Model

Our tagging framework is based on a maximum entropy model of the following form:

$$p(t, c) = \gamma \prod_{k=0}^{K} \alpha_k^{f_k(c,t)} p_0 \qquad (1)$$

where:

```
(_( Please_RRCONCESSIVE Mention_VVIVERBAL-ACT this_DD1 coupon_NN1DOCUMENT
when_CSWHEN ordering_VVGINTER-ACT

OR_CCOR ONE_MC1WORD FREE_JJMONEY FANTAIL_NN1ANIMAL SHRIMPS_NN1FOOD
```

Figure 1: Two ATR Treebank Sentences from a Take–Out Food Flier

- t is tag being predicted;
- c is the context of t;
- γ is a normalization coefficient that ensures: $\Sigma_{t=0}^{L} \gamma \prod_{k=0}^{K} \alpha_k^{f_k(c,t)} p_0 = 1$;
- K is the number of features in the model;
- L is the number of tags in our tag set;
- α_k is the weight of feature f_k;
- f_k are feature functions and $f_k \epsilon \{0, 1\}$;
- p_0 is the default tagging model (in our case, the uniform distribution, since all of the information in the model is specified using ME constraints).

Our baseline model contains the following feature predecate set:

$$\begin{array}{llll}
w_0 & t_{-1} & pos_0 & pref_1(w_0) \\
w_{-1} & t_{-2} & pos_{-1} & pref_2(w_0) \\
w_{-2} & & pos_{-2} & pref_3(w_0) \\
w_{+1} & & pos_{+1} & suff_1(w_0) \\
w_{+2} & & pos_{+2} & suff_2(w_0) \\
 & & & suff_3(w_0)
\end{array}$$

where:

- w_n is the word at offset n relative to the word whose tag is being predicted;
- t_n is the tag at offset n;
- pos_n is the syntax-only tag at offset n assigned by a syntax-only tagger;
- $pref_n(w_0)$ is the first n characters of w_0;
- $suff_n(w_0)$ is the last n characters of w_0;

This feature set contains a typical selection of n-gram and basic morphological features. When the tagger is trained in tested on the UPENN treebank (Marcus et al., 1994), its accuracy (excluding the pos_n features) is over 96%, close to the state of the art on this task. (Black et al., 1996b) adopted a two-stage approach to prediction, first predicting syntax, then semantics given the syntax, whereas in (Black et al., 1998) both syntax and semantics were predicted together in one step. In using syntactic tags as features, we take a softer approach to the two-stage process. The tagger has access to accurate syntactic information; however, it is not necessarily constrained to accept this choice of syntax. Rather, it is able to decide both syntax and semantics while taking semantic context into account. In order to find the most probable sequence of tags, we tag in a left-to-right manner using a beam-search algorithm.

4.2 Feature selection

For reasons of practicability, it is not always possible to use the full set of features in a model: often it is necessary to control the number of features to reduce resource requirements during training. We use mutual information (MI) to select the most useful feature predicates (for more details, see (Rosenfeld, 1996)). It can be viewed as a means of determining how much information a given predicate provides when used to predict an outcome.

That is, we use the following formula to gauge a feature's usefulness to the model:

$$I(f;T) = \sum_{f \in \{0,1\}} \sum_{t \in T} p(f,t) log \frac{p(f,t)}{p(f)p(t)} \qquad (2)$$

where:

- $t \in T$ is a tag in the tagset;
- $f \in \{0, 1\}$ is the value of any kind of predicate feature.

Using mutual information is not without its shortcomings. It does not take into account any of the interactions between features. It is possible for a feature to be pronounced useful by this procedure, whereas in fact it is merely giving the same information as another feature but in different form. Nonetheless this technique is invaluable in practice. It is possible to eliminate features which provide little or no benefit to the model, thus speeding up the training. In some cases it even allows a model to be trained where it would not otherwise be possible to train one. For the purposes of our experiments, we use the top 50,000 predicates for each model to form the feature set.

5 External Knowledge Sources

5.1 Lexical Dependencies

Features derived from n-grams of words and tags in the immediate vicinity of the word being tagged have underpinned the world of POS tagging for many years (Kupiec, 1992; Merialdo, 1994; Ratnaparkhi, 1996), and have proven to be useful features in WSD (Yarowsky, 1993). Lower-order n-grams which are closer to word being tagged offer the greatest predictive power (Black et al., 1998). However, in the field of WSD, relational information extracted from grammatical analysis of the sentence has been employed to good effect, and in particular, subject-object relationships between verbs and nouns have been shown be effective in disambiguating semantics (Nancy and Jean, 1998). We take the broader view that dependency relationships in general between any classes of words may help, and use the ME training process to weed out the irrelevant relationships. The principle is exactly the same as when using a word in the local context as a feature, except that the word in this case has a grammatical relationship with the word being tagged, and can be outside the local neighborhood of the word being tagged. For both types of dependency, we encoded the model constraints $f_{stl}(d)$ as boolean functions of the form:

$$f_{stl}(d) = \begin{cases} 1 & \text{if } d.s = s \wedge d.t = t \wedge d.l = l \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where:

- d is a lexical dependency, consisting of a source word (the word being tagged) $d.s$, a target word $d.t$ and a label $d.l$
- s and t (words), and l (link label) are specific to the feature

We generated two distinct features for each dependency. The source and target were exchanged to create these features. This was to allow the models to capture the bidirectional nature of the dependencies. For example, when tagging a verb, the model should be aware of the dependent object, and conversely when tagging that object, the model should have a feature imposing a constraint arising from the identity of the dependent verb.

5.1.1 Dependencies from the CMU Link Grammar

We parsed our corpus using the parser detailed in (Grinberg et al., 1995). The dependencies output by this parser are labeled with the type of dependency (connector) involved. For example, subjects (connector type S) and direct objects of verbs (O) are explicitly marked by the process (a full list of connectors is provided in the paper). We used all of the dependencies output by the parser as features in the models.

5.1.2 Dependencies from Phrasal Structure

It is possible to extract lexical dependencies from a phrase-structure parse. The procedure is explained in detail in (Collins, 1996). In essence, each non-terminal node in the parse tree is assigned a head word, which is the head of one of its children denoted the 'head child'. Dependencies are established between this headword and the heads of each of the children (except for the head child). In these experiments we used the MXPOST tagger (Ratnaparkhi, 1996) combined with Collins' parser (Collins, 1996) to assign parse trees to the corpus. The parser had a 98.9% coverage of the sentences in our corpora. Again, all of the dependencies output by the parser were used as features in the models.

5.2 Hierarchical Word Ontologies

In this section we consider the effect of features derived from hierarchical sets of words. The primary advantage is that we are able to construct these hierarchies using knowledge from outside the training corpus of the tagger itself, and thereby glean knowledge about rare words. In these experiments we use the human annotated word taxonomy of hypernyms (IS-A relations) in the WordNet database, and an automatically acquired ontology made by clustering words in a large corpus of unannotated text.

We have chosen to use hierarchical schemes for both the automatic and manually acquired ontologies because this offers the opportunity to combat data-sparseness issues by allowing features derived from all levels of the hierarchy to be used. The process of training the model is able to de-

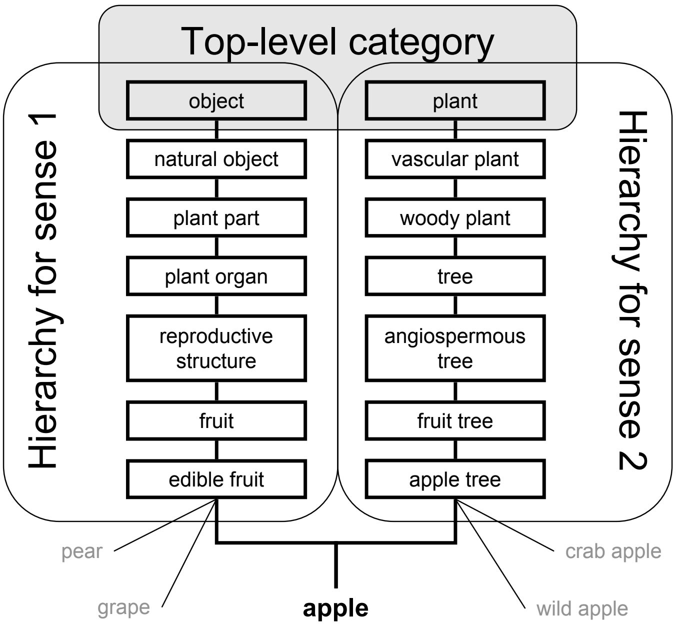

Figure 2: The WordNet taxonomy for both (WordNet) senses of the word *apple*

cide the levels of granularity that are most useful for disambiguation. For the purposes of generating features for the ME tagger we treat both types of hierarchy in the same fashion. One of these features is illustrated in Figure 5.3. Each predicate is effectively a question which asks whether the word (or word being used in a particular sense in the case of the WordNet hierarchy) is a descendent of the node to which the predicate applies. These predicates become more and more general as one moves up the hierarchy. For example in the hierarchy shown in Figure 5.2, looking at the nodes on the right hand branch, the lowest node represents the class of **apple trees** whereas the top node represents the class of all **plants**.

We expect these hierarchies to be particularly useful when tagging out of vocabulary words (OOV's). The identity of the word being tagged is by far the most important feature in our baseline model. When tagging an OOV this information is not available to the tagger. The automatic clustering has been trained on 100 times as much data as our tagger, and therefore will have information about words that tagger has not seen during training. To illustrate this point, suppose that we are tagging the OOV *pomegranate*. This word is in the WordNet database, and is in the same synset as the 'fruit' sense of the word *apple*. It is reasonable to assume that the model will have learned (from the many examples of all fruit words) that the predicate representing membership of this **fruit** synset should, if true, favor the selection of the correct tag for fruit words: NN1FOOD. The predicate will be true for the word *pomegranate* which will thereby benefit from the model's knowledge of how to tag the other words in its class. Even if this is not so at this level in the hierarchy, it is likely to be so at some level of granularity. Precisely which levels of detail are useful will be learned by the model during training.

5.2.1 Automatic Clustering of Text

We used the automatic agglomerative mutual-information-based clustering method of (Ushioda, 1996) to form hierarchical clusters from approximately 50 million words of tokenized, unannotated text drawn from similar domains as the treebank used to train the tagger. Figure 5.2 shows the position of the word *apple* within the hierarchy of clusters. This example highlights both the strengths and weaknesses of this approach. One strength is that the process of clustering proceeds in a purely objective fashion and associations between words that may not have been considered by a human annotator are present. Moreover, the clustering process considers all types that actually occur in the corpus, and not just those words that might appear in a dictionary (we will return to this later). A major problem with this approach is that

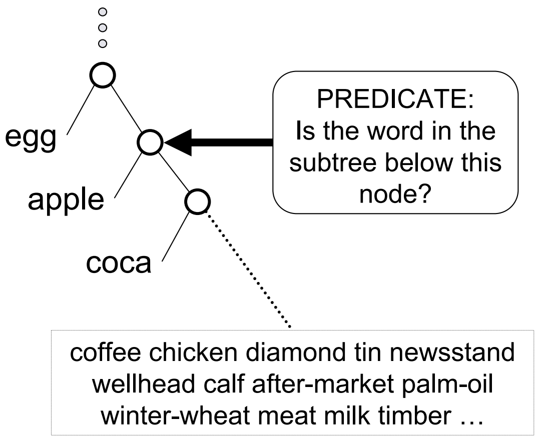
egg
apple
coca
PREDICATE:
Is the word in the subtree below this node?
coffee chicken diamond tin newsstand wellhead calf after-market palm-oil winter-wheat meat milk timber …

Figure 3: The dendrogram for the automatically acquired ontology, showing the word *apple*

the clusters tend to contain a lot of noise. Rare words can easily find themselves members of clusters to which they do not seem to belong, by virtue of the fact that there are too few examples of the word to allow the clustering to work well for these words. This problem can be mitigated somewhat by simply increasing the size of the text that is clustered. However the clustering process is computationally expensive. Another problem is that a word may only be a member of a single cluster; thus typically the cluster set assigned to a word will only be appropriate for that word when used in its most common sense.

Approximately 93% of running words in the test corpus, and 95% in the training corpus were covered by the words in the clusters (when restricted to verbs, nouns, adjectives and adverbs, these figures were 94.5% and 95.2% respectively). Approximately 81% of the words in the vocabulary from the test corpus were covered, and 71% of the training corpus vocabulary was covered.

5.2.2 WordNet Taxonomy

For this class of features, we used the hypernym taxonomy of WordNet (Fellbaum, 1998). Figure 5.2 shows the WordNet hypernym taxonomy for the two senses of the word *apple* that are in the database. The set of predicates query membership of all levels of the taxonomy for all WordNet senses of the word being tagged. An example of one such predicate is shown in the figure.

Only 63% of running words in both the training and the test corpus were covered by the words in the clusters. Although this figure appears low, it can be explained by the fact that WordNet only contains entries for words that have senses in certain parts of speech. Some very frequent classes of words, for example determiners, are not in WordNet. The coverage of only nouns, verbs, adjectives and adverbs in running text is 94.5% for both training and test sets. Moreover, approximately 84% of the words in the vocabulary from the test corpus were covered, and 79% on the training corpus. Thus, the effective coverage of WordNet on the important classes of words is similar to that of the automatic clustering method.

6 Experimental Results

The results of our experiments are shown in Table 1. The task of assigning semantic and syntactic tags is considerably more difficult than simply assigning syntactic tags due to the inherent ambiguity of the tagset. To gauge the level of human performance on this task, experiments were conducted to determine inter-annotator consistency; in addition, annotator accuracy was measured on 5,000 words of data. Both the agreement and accuracy were found to be approximately 97%, with all of the inconsistencies and tagging errors arising from the semantic component of the tags. 97% accuracy is therefore an approximate upper bound for the performance one would expect from an automatic tagger. As a point of reference for a lower bound, the overall accuracy of a tagger which uses only a single feature representing the identity of the word being tagged is approximately 73%.

The overall baseline accuracy was 82.58% with only 30.58% of OOV's being tagged correctly. Of the two lexical dependency-based approaches, the features derived from Collins' parser were the most effective, improving accuracy by 0.8% overall. To put the magnitude of this gain into perspective, dropping the features for the identity of the previous word from the baseline model, only degraded performance by 0.2%. The features from the link grammar parser were handicapped due to the fact that only 31% of the sentences were able to be parsed. When the model (Model 3 in Table 1) was evaluated on only the parsable portion on the test set, the accuracy obtained was roughly comparable to that using the dependencies from Collins' parses. To control for the differences between these parseable sentences and the full test set, Model 4 was tested on the same 31% of sentence that parsed. Its accuracy was within 0.2% of the accuracy on the whole test set in all cases. Neither of the lexical dependency-based approaches had a particularly strong effect on the performance on OOV's. This is in line with our intuition, since these features rely on the identity of the word being tagged, and the performance gain we see is due to the improvement in labeling accuracy of the context around the OOV.

In contrast to this, for the word-ontology-based feature sets, one would hope to see a marked improvement on OOV's, since these features were designed specifically to address this issue. We do see a strong response to these features in the accuracy of the models. The overall accuracy when using the automatically acquired ontology is only 0.1% higher than the accuracy using dependencies from Collins' parser. However the accuracy on OOV's jumps 3.5% to 35.08% compared to just 0.7% for Model 4. Performance for both clustering techniques was quite similar, with the WordNet taxonomical features being slightly more useful, especially for OOV's. One possible explanation for this is that overall, the coverage of both techniques is similar, but for rarer words, the MI clustering can be inconsistent due to lack of data (for an example, see Figure 5.2: the word *newsstand* is a member of a cluster of words that appear to be commodities), whereas the WordNet clustering remains consistent even for rare words. It seems reasonable to expect, however, that the automatic method would do better if trained on more data. Furthermore, all uses of words can be covered by automatic clustering, whereas for example, the common use of the word *apple* as a company name is beyond the scope of WordNet.

In Model 7 we combined the best lexical dependency feature set (Model 4) with the best clustering feature set (Model 6) to investigate the amount of information overlap existing between the feature sets. Models 4 and 6 improved the baseline performance by 0.8% and 1.3% respectively. In combination, accuracy was increased by 2.3%, 0.2% more than the sum of the component models' gains. This is very encouraging and indicates that these models provide independent information, with virtually all of the benefit from both models manifesting itself in the combined model.

7 Conclusion

We have described a method for simultaneously labeling the syntax and semantics of words in running text. We develop this method starting from a state-of-the-art maximum entropy POS tagger which itself outperforms previous attempts to tag this data (Black et al., 1996b). We augment this tagging model with two distinct types of knowledge: the identity of dependent words in the sentence, and word class membership information of the word being tagged. We define the features in such a manner that the useful lexical dependencies are selected by the model, as is the granularity of the word classes used. Our experimental results show that large gains in performance are obtained using each of the techniques. The dependent words boosted overall performance, especially when tagging verbs. The hierarchical ontology-based approaches also increased overall performance, but with particular emphasis on OOV's, the intended target for this feature set. Moreover, when features from both knowledge sources were applied in combination, the gains were cumulative, indicating little overlap.

Visual inspection the output of the tagger on held-out data suggests there are many remaining errors arising from special cases that might be better handled by models separate from the main tagging model. In particular, numerical expressions and named entities cause OOV errors that the techniques presented in this paper are unable to handle. In future work we would like to address these issues, and also evaluate our system when used as a component of a WSD system, and when integrated within a machine translation system.

#	Model	Accuracy (± c.i.)	OOV's	Nouns	Verbs	Adj/Adv
1	Baseline	82.58± 0.32	30.58	68.47	74.32	70.99
2	+ Dependencies (link grammar)	82.74± 0.32	30.92	68.18	74.96	73.02
3	As above (only parsed sentences)	83.59± 0.53	30.92	69.16	77.21	73.52
4	+ Dependencies (Collins' parser)	83.37± 0.31	31.24	69.36	75.78	72.62
5	+ Automatically acquired ontology	83.71± 0.31	35.08	71.89	75.83	75.34
6	+ WordNet ontology	83.90± 0.31	36.18	72.28	76.29	74.47
7	+ Model 4 + Model 6	84.90± 0.31	37.02	72.80	78.36	76.16

Table 1: Tagging accuracy (%), '+' being shorthand for "Baseline +", 'c.i.' denotes the confidence interval of the mean at a 95% significance level, calculated using bootstrap resampling.

References

E. Black and A. Finch. 2001. Developing and proving effective broad-coverage semantic-and-syntactic tagsets for natural language: The atr approach. In *Proceedings of ICCPOL-2001*.

E. Black, S. Eubank, H. Kashioka, R. Garside, G. Leech, and D. Magerman. 1996a. Beyond skeleton parsing: producing a comprehensive large–scale general–english treebank with full grammatical analysis. In *Proceedings of the 16th Annual Conference on Computational Linguistics*, pages 107–112, Copenhagen.

E. Black, S. Eubank, H. Kashioka, and J. Saia. 1996b. Reinventing part-of-speech tagging. *Journal of Natural Language Processing (Japan)*, 5:1.

Ezra Black, Andrew Finch, and Hideki Kashioka. 1998. Trigger-pair predictors in parsing and tagging. In *Proceedings, 36th Annual Meeting of the Association for Computational Linguistics, 17th Annual Conference on Computational Linguistics*, Montreal, Canada.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, San Francisco. Morgan Kaufmann Publishers.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Dennis Grinberg, John Lafferty, and Daniel Sleator. 1995. A robust parsing algorithm for LINK grammars. Technical Report CMU-CS-TR-95-125, CMU, Pittsburgh, PA.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6:225–242.

A. K. Lamjiri, O. El Demerdash, and L.Kosseim. 2004. Simple features for statistical word sense disambiguation. In *Proc. ACL 2004 – Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, Barcelona, Spain, July. ACL-2004.

C. Li and H. Li. 2002. Word translation disambiguation using bilingual bootstrapping.

Xiaobin Li, Stan Szpakowicz, and Stan Matwin. 1995. A wordnet-based algorithm for word sense disambiguation. In *IJCAI*, pages 1368–1374.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

B. Merialdo. 1994. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

Rada Mihalcea and Dan I. Moldovan. 1998. Word sense disambiguation based on semantic density. In Sanda Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 16–22. Association for Computational Linguistics, Somerset, New Jersey.

I. Nancy and V. Jean. 1998. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1:1–40.

G. Ramakrishnan and B. Prithviraj. 2004. Soft word sense disambiguation. In *International Conference on Global Wordnet (GWC 04)*, Brno, Czeck Republic.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*.

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10:187–228.

A. Suarez. 2002. A maximum entropy-based word sense disambiguation system. In *Proc. International Conference on Computational Linguistics*.

A. Ushioda. 1996. Hierarchical clustering of words. In *In Proceedings of COLING 96*, pages 1159–1162.

D. Yarowsky. 1993. One sense per collocation. In *In the Proceedings of ARPA Human Language Technology Workshop*.