Multilingual Deep Lexical Acquisition for HPSGs via Supertagging

Phil Blunsom and Timothy Baldwin
Computer Science and Software Engineering
University of Melbourne, Victoria 3010 Australia
{pcbl,tim}@csse.unimelb.edu.au

Abstract

We propose a conditional random field-based method for supertagging, and apply it to the task of learning new lexical items for HPSG-based precision grammars of English and Japanese. Using a pseudo-likelihood approximation we are able to scale our model to hundreds of supertags and tens-of-thousands of training sentences. We show that it is possible to achieve start-of-the-art results for both languages using maximally language-independent lexical features. Further, we explore the performance of the models at the type- and token-level, demonstrating their superior performance when compared to a unigram-based baseline and a transformation-based learning approach.

1 Introduction

Over recent years, there has been a resurgence of interest in the use of precision grammars in NLP tasks, due to advances in parsing algorithm development, grammar development tools and raw computational power (Oepen et al., 2002b). **Precision grammars** are defined as implemented grammars of natural language which capture fine-grained linguistic distinctions, and are generative in the sense of distinguishing between grammatical and ungrammatical inputs (or at least have some in-built notion of linguistic "markedness"). Additional characteristics of precision grammars are that they are frequently bidirectional, and output a rich semantic abstraction for each spanning parse of the input string. Examples include DELPH-IN grammars such as the English Resource Grammar (Flickinger, 2002; Uszkoreit, 2002), the various PARGRAM grammars (Butt et al., 1999), and the Edinburgh CCG parser (Bos et al., 2004).

Due to their linguistic complexity, precision grammars are generally hand-constructed and thus restricted in size and coverage. Attempts to (semi-)automate the process of expanding the coverage of precision grammars have focused on either: (a) constructional coverage, e.g. in the form of error mining for constructional expansion (van Noord, 2004; Zhang and Kordoni, 2006), or relaxation of lexico-grammatical constraints to support partial and/or robust parsing (Riezler et al., 2002); or (b) lexical coverage, e.g. in bootstrapping from a pre-existing grammar and lexicon to learn new lexical items (Baldwin, 2005a). Our particular interest in this paper is in the latter of these two, that is the development of methods for automatically expanding the lexical coverage of an existing precision grammar, or more broadly **deep lexical acquisition** (DLA hereafter). In this, we follow Baldwin (2005a) in assuming a semi-mature precision grammar with a fixed inventory of lexical types, based on which we learn new lexical items. For the purposes of this paper, we focus specifically on supertagging as the mechanism for hypothesising new lexical items.

Supertagging can be defined as the process of applying a sequential tagger to the task of predicting the lexical type(s) associated with each word in an input string, relative to a given grammar. It was first introduced as a means of reducing parser ambiguity by Bangalore and Joshi (1999) in the context of the LTAG formalism, and has since been applied in a similar context within the CCG formalism (Clark and Curran, 2004). In both of these cases, supertagging provides the means to perform a beam search over the plausible lexical items for a given string context, and ideally reduces parsing complexity without sacrificing parser accuracy. An alternate application of supertagging is in DLA, in postulating novel lexical items with which to populate the lexicon of a given grammar to boost parser coverage. This can take place

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 164–171,
Sydney, July 2006. ©2006 Association for Computational Linguistics

either: (a) off-line for the purposes of rounding out the coverage of a static lexicon, in which case we are generally interested in globally maximising precision over a given corpus and hence predicting the single most plausible lexical type for each word token (**off-line DLA**: Baldwin (2005b)); or (b) on the fly for a given input string to temporarily expand lexical coverage and achieve a spanning parse, in which case we are interested in maximising recall by producing a (possibly weighted) list of lexical item hypotheses to run past the grammar (**on-line DLA**: Zhang and Kordoni (2005)). Our immediate interest in this paper is in the first of these tasks, although we would ideally like to develop an off-line method which is trivially portable to the second task of on-line DLA.

In this research, we focus particularly on the Grammar Matrix-based DELPH-IN family of grammars (Bender et al., 2002), which includes grammars of English, Japanese, Norwegian, Modern Greek, Portuguese and Korean. The Grammar Matrix is a framework for streamlining and standardising HPSG-based multilingual grammar development. One property of Grammar Matrix-based grammars is that they are strongly lexicalist and adhere to a highly constrained lexicon-grammar interface via a unique (terminal) lexical type for each lexical item. As such, lexical item creation in any of the Grammar Matrix-based grammars, irrespective of language, consists predominantly of predicting the appropriate lexical type for each lexical item, relative to the lexical hierarchy for the corresponding grammar. In this same spirit of standardisation and multilinguality, the aim of this research is to develop maximally language-independent supertagging methods which can be applied to any Grammar Matrix-based grammar with the minimum of effort. Essentially, we hope to provide the grammar engineer with the means to semi-automatically populate the lexicon of a semi-mature grammar, hence accelerating the pace of lexicon development and producing a resource of sufficient coverage to be practically useful in NLP tasks.

The contributions of this paper are the development of a pseudo-likelihood conditional random field-based method of supertagging, which we then apply to the task of off-line DLA for grammars of both English and Japanese with only minor language-specific adaptation. We show the supertagger to outperform previously-proposed supertagger-based DLA methods.

The remainder of this paper is structured as follows. Section 2 outlines past work relative to this research, and Section 3 reviews the resources used in our supertagging experiments. Section 4 outlines the proposed supertagger model and reviews previous research on supertagger-based DLA. Section 5 then outlines the set-up and results of our evaluation.

2 Past Research

According to Baldwin (2005b), research on DLA falls into the two categories of *in vitro* methods, where we leverage a secondary language resource to generate an abstraction of the words we hope to learn lexical items for, and *in vivo* methods, where the target resource that we are hoping to perform DLA relative to is used directly to perform DLA. Supertagging is an instance of *in vivo* DLA, as it operates directly over data tagged with the lexical type system for the precision grammar of interest.

Research on supertagging which is relevant to this paper includes the work of Baldwin (2005b) in training a transformation-based learner over data tagged with ERG lexical types. We discuss this method in detail in Section 5.2 and replicate this method over our English data set for direct comparability with this previous research.

As mentioned above, other work on supertagging has tended to view it as a means of driving a beam search to prune the parser search space (Bangalore and Joshi, 1999; Clark and Curran, 2004). In supertagging, token-level annotations (gold-standard, automatically-generated or otherwise) for a given DLR are used to train a sequential tagger, akin to training a POS tagger over POS-tagged data taken from the Penn Treebank.

One related *in vivo* approach to DLA targeted specifically at precision grammars is that of Fouvry (2003). Fouvry uses the grammar to guide the process of learning lexical items for unknown words, by generating underspecified lexical items for all unknown words and parsing with them. Syntactico-semantic interaction between unknown words and pre-existing lexical items during parsing provides insight into the nature of each unknown word. By combining such fragments of information, it is possible to incrementally arrive at a consolidated lexical entry for that word. That is, the precision grammar itself drives the incremental learning process within a parsing context.

An alternate approach is to compile out a set of word templates for each lexical type (with the important qualification that they do not rely on preprocessing of any form), and check for corpus occurrences of an unknown word in such contexts. That is, the morphological, syntactic and/or semantic predictions implicit in each lexical type are made explicit in the form of templates which represent distinguishing lexical contexts of that lexical type. This approach has been shown to be particularly effective over web data, where the sheer size of the data precludes the possibility of linguistic preprocessing but at the same time ameliorates the effects of data sparseness inherent in any lexicalised DLA approach (Lapata and Keller, 2004).

Other work on DLA (e.g. Korhonen (2002), Joanis and Stevenson (2003), Baldwin (2005a)) has tended to take an *in vitro* DLA approach, in extrapolating away from a DLR to corpus or web data, and analysing occurrences of words through the conduit of an external resource (e.g. a secondary parser or POS tagger). *In vitro* DLA can also take the form of resource translation, in mapping one DLR onto another to arrive at the lexical information in the desired format.

3 Task and Resources

In this section, we outline the resources targeted in this research, namely the English Resource Grammar (ERG: Flickinger (2002), Copestake and Flickinger (2000)) and the JACY grammar of Japanese (Siegel and Bender, 2002). Note that our choice of the ERG and JACY as testbeds for experimentation in this paper is somewhat arbitrary, and that we could equally run experiments over any Grammar Matrix-based grammar for which there is treebank data.

Both the ERG and JACY are implemented open-source broad-coverage precision Head-driven Phrase Structure Grammars (HPSGs: Pollard and Sag (1994)). A lexical item in each of the grammars consists of a unique identifier, a lexical type (a leaf type of a type hierarchy), an orthography, and a semantic relation. For example, in the English grammar, the lexical item for the noun *dog* is simply:

```
dog_n1 := n_-_c_le &
 [ STEM < "dog" >,
   SYNSEM [ LKEYS.KEYREL.PRED "_dog_n_1_rel" ] ].
```

in which the lexical type of `n_-_c_le` encodes the fact that *dog* is a noun which does not subcategorise for any other constituents and which is countable, `"dog"` specifies the lexical stem, and `"_dog_n_1_rel"` introduces an ad hoc predicate name for the lexical item to use in constructing a semantic representation. In the context of the ERG and JACY, DLA equates to learning the range of lexical types a given lexeme occurs with, and generating a single lexical item for each.

Recent development of the ERG and JACY has been tightly coupled with treebank annotation, and all major versions of both grammars are deployed over a common set of dynamically-updateable treebank data to help empirically trace the evolution of the grammar and retrain parse selection models (Oepen et al., 2002a; Bond et al., 2004). This serves as a source of training and test data for building our supertaggers, as detailed in Table 1.

In translating our treebank data into a form that can be understood by a supertagger, multiword expressions (MWEs) pose a slight problem. Both the ERG and JACY include multiword lexical items, which can either be **strictly continuous** (e.g. *hot line*) or **optionally discontinuous** (e.g. transitive English verb particle constructions, such as *pick up* as in *Kim picked the book up*).

Strictly continuous lexical items are described by way of a single whitespace-delimited lexical stem (e.g. `STEM < "hot line" >`). When faced with instances of this lexical item, the supertagger must perform two roles: (1) predict that the words *hot* and *line* combine together to form a single lexeme, and (2) predict the lexical type associated with the lexeme. This is performed in a single step through the introduction of the `ditto` lexical type, which indicates that the current word combines (possibly recursively) with the left-adjacent word to form a single lexeme, and shares the same lexical type. This tagging convention is based on that used, e.g., in the CLAWS7 part-of-speech tagset.

Optionally discontinuous lexical items are less of a concern, as selection of each of the discontinuous "components" is done via lexical types. E.g. in the case of *pick up*, the lexical entry looks as follows:

```
pick_up_v1 := v_p-np_le &
 [ STEM < "pick" >,
   SYNSEM [ LKEYS [ --COMPKEY _up_p_sel_rel,
                    KEYREL.PRED "_pick_v_up_rel" ] ] ].
```

in which `"pick"` selects for the `_up_p_sel_rel` predicate, which in turn is associated with the stem `"up"` and lexical type `p_prtcl_le`. In terms of lexical tag mark-up, we can treat these as separate

	ERG	JACY
GRAMMAR		
Language	English	Japanese
Lexemes	16,498	41,559
Lexical items	26,297	47,997
Lexical types	915	484
Strictly continuous MWEs	2,581	422
Optionally discontinuous MWEs	699	0
Proportion of lexemes with more than one lexical item	0.29	0.14
Average lexical items per lexeme	1.59	1.16
TREEBANK		
Training sentences	20,000	40,000
Training words	215,015	393,668
Test sentences	1,013	1,095
Test words	10,781	10,669

Table 1. Make-up of the English Resource Grammar (ERG) and JACY grammars and treebanks

tags and leave the supertagger to model the mutual inter-dependence between these lexical types.

For detailed statistics of the composition of the two grammars, see Table 1.

For morphological processing (including tokenisation and lemmatisation), we use the pre-existing machinery provided with each of the grammars. In the case of the ERG, this consists of a finite state machine which feeds into lexical rules; in the case of JACY, segmentation and lemmatisation is based on a combination of ChaSen (Matsumoto et al., 2003) and lexical rules. That is, we are able to assume that the Japanese data has been pre-segmented in a form compatible with JACY, as we are able to replicate the automatic pre-processing that it uses.

4 Suppertagging

The DLA strategy we adopt in this research is based on supertagging, which is a simple instance of sequential tagging with a larger, more linguistically-diverse tag set than is conventionally the case, e.g., with part-of-speech tagging. Below, we describe the pseudo-likelihood CRF model we base our supertagger on and outline the feature space for the two grammars.

4.1 Pseudo-likelihood CRF-based Supertagging

CRFs are undirected graphical models which define a conditional distribution over a label sequence given an observation sequence. Here we use CRFs to model sequences of lexical types, where each input word in a sentence is assigned a single tag.

The joint probability density of a sequence labelling, $\mathbf{a}$ (a vector of lexical types), given the input sentence, $\mathbf{s}$, is given by:

$$p_\Lambda(\mathbf{a}|\mathbf{s}) = \frac{\exp \sum_t \sum_k \lambda_k h_k(t, a_{t-1}, a_t, \mathbf{s})}{Z_\Lambda(\mathbf{s})} \quad (1)$$

where we make a first order Markov assumption over the label sequence. Here t ranges over the word indices of the input sentence ($\mathbf{s}$), k ranges over the model's features, and $\Lambda = \{\lambda_k\}$ are the model parameters (weights for their corresponding features). The feature functions h_k are predefined real-valued functions over the input sentence coupled with the lexical type labels over adjacent "times" (= sentence locations) t. These feature functions are unconstrained, and may represent overlapping and non-independent features of the data. The distribution is globally normalised by the partition function, $Z_\Lambda(\mathbf{s})$, which sums out the numerator in (1) for every possible labelling:

$$Z_\Lambda(\mathbf{s}) = \sum_{\mathbf{a}} \exp \sum_t \sum_k \lambda_k h_k(t, a_{t-1}, a_t, \mathbf{s})$$

We use a linear chain CRF, which is encoded in the feature functions of (1).

The parameters of the CRF are usually estimated from a fully observed training sample, by maximising the likelihood of these data. I.e. $\Lambda^{ML} = \arg\max_\Lambda p_\Lambda(\mathcal{D})$, where $\mathcal{D} = \{(\mathbf{a}, \mathbf{s})\}$ is the complete set of training data.

However, as calculating $Z_\Lambda(\mathbf{s})$ has complexity quadratic in the number of labels, we need to approximate $p_\Lambda(\mathbf{a}|\mathbf{s})$ in order to scale our model to hundreds of lexical types and tens-of-thousands of training sentences. Here we use the pseudo-likelihood approximation p_Λ^{PL} (Li, 1994) in which the marginals for a node at time t are calculated with its neighbour nodes' labels fixed to those ob-

FEATURE	DESCRIPTION
WORD CONTEXT FEATURES	
$\mathit{lexeme}(\mathbf{s}_t) = x$ & $\mathbf{a}_t = l$	lexeme + label
$\mathbf{s}_t = w$ & $\mathbf{a}_t = l$	word unigram + label
$\mathbf{s}_{t-1} = w$ & $\mathbf{a}_t = l$	previous word unigram + label
$\mathbf{s}_{t+1} = w$ & $\mathbf{a}_t = l$	next word unigram + label
$\mathbf{s}_t = w$ & $\mathbf{s}_{t-1} = y$ & $\mathbf{a}_t = l$	previous word bigram + label
$\mathbf{s}_t = w$ & $\mathbf{s}_{t+1} = y$ & $\mathbf{a}_t = l$	next word bigram + label
$\mathbf{a}_{t-1} = l$ & $\mathbf{a}_t = m$	clique label pair
LEXICAL FEATURES	
$\mathit{prefix}_n(\mathbf{s}_t)$ & $\mathbf{a}_t = l$	n-gram prefix + label
$\mathit{suffix}_n(\mathbf{s}_t) = x$ & $\mathbf{a}_t = l$	n-gram suffix + label
$\mathit{contains}(\mathbf{s}_t, C_i)$ & $\mathbf{a}_t = l$	word contains element of character set C_i + label

Table 2. Extracted feature types for the CRF model

served in the training data:

$$U_\Lambda^{PL}(i,\mathbf{s},t) = \sum_k \lambda_k(h_k(t,\hat{a}_{t-1},i,\mathbf{s}) + h_k(t,i,\hat{a}_{t+1},\mathbf{s})) \qquad (2)$$

$$p_\Lambda^{PL}(\mathbf{a}|\mathbf{s}) = \prod_t \frac{\exp(U_\Lambda^{PL}(\mathbf{a}_t,\mathbf{s},t))}{\sum_l (U_\Lambda^{PL}(l,\mathbf{s},t))} \qquad (3)$$

where $\hat{\mathbf{a}}_t$ is the lexical type label observed in the training data and l ranges over the label set. This approximation removes the need to calculate the partition function, thus reducing the complexity to be linear in the number of labels and training instances.

Because maximum likelihood estimators for log-linear models have a tendency to overfit the training sample (Chen and Rosenfeld, 1999), we define a prior distribution over the model parameters and derive a maximum *a posteriori* (MAP) estimate, $\Lambda^{MAP-PL} = \arg\max_\Lambda p_\Lambda^{PL}(\mathcal{D})p(\Lambda)$. We use a zero-mean Gaussian prior, with the probability density function $p_0(\lambda_k) \propto \exp\left(-\frac{\lambda_k^2}{2\sigma_k^2}\right)$. This yields a log-pseudo-likelihood objective function of:

$$\mathcal{L}^{PL} = \sum_{(\mathbf{a},\mathbf{s})\in\mathcal{D}} \log p_\Lambda^{PL}(\mathbf{a}|\mathbf{s}) + \sum_k \log p_0(\lambda_k) \qquad (4)$$

In order to train the model, we maximize (4). While the log-pseudo-likelihood cannot be maximised for the parameters, Λ, in closed form, it is a convex function, and thus we resort to numerical optimisation to find the globally optimal parameters. We use L-BFGS, an iterative quasi-Newton optimisation method, which performs well for training log-linear models (Malouf, 2002; Sha and Pereira, 2003). Each L-BFGS iteration requires the objective value and its gradient with respect to the model parameters.

As we cannot observe label values for the test data we must use $p_\Lambda(\mathbf{a}|\mathbf{s})$ when decoding. The Viterbi algorithm is used to find the maximum posterior probability alignment for test sentences, $\mathbf{a}^* = \arg\max_\mathbf{a} p_\Lambda(\mathbf{a}|\mathbf{s})$.

4.2 CRF features

One of the strengths of the CRF model is that it supports the use of a large number of non-independent and overlapping features of the input sentence. Table 2 lists the word context and lexical features used by the CRF model (shared across both grammars).

Word context features were extracted from the words and lexemes of the sentence to be labelled combined with a proposed label. A clique label pair feature was also used to model sequences of lexical types.

For the lexical features, we generate a feature for the unigram, bigram and trigram prefixes and suffixes of each word (e.g. for *bottles*, we would generate the prefixes *b*, *bo* and *bot*, and the suffixes *s*, *es* and *les*); for words in the test data, we generate a feature only if that feature-value is attested in the training data. We additionally test each word for the existence of one or more elements of a range of character sets C_i. In the case of English, we focus on five character sets: upper case letters, lower case letters, numbers, punctuation and hyphens. For the Japanese data, we employ six character sets: Roman letters, hiragana, katakana, kanji, (Arabic) numerals and punctuation. For example, カビ臭い "mouldy" would be flagged as containing katakana character(s), kanji character(s) and hiragana character(s) only. Note that the only language-dependent component of

	ERG					JACY				
	Acc	Acc$_U$	Prec	Rec	F-score	Acc	Acc$_U$	Prec	Rec	F-score
Baseline	0.802	0.053	0.184	0.019	0.034	0.866	0.592	0.680	0.323	0.438
fnTBL	0.915	0.236	0.370	0.038	0.068	—	—	—	—	—
CRF$_{-\text{LEX}}$	0.911	0.427	0.339	0.053	0.092	0.920	0.816	0.548	0.414	0.471
CRF$_{+\text{LEX}}$	**0.917**	**0.489**	**0.509**	**0.059**	**0.105**	**0.932**	**0.827**	**0.696**	**0.424**	**0.527**

Table 3. Results of supertagging for the ERG and JACY (best result in each column in **bold**)

the lexical features is the character sets, which requires little or no specialist knowledge of the language. Note also that for languages with infixing, such as Tagalog, we may want to include n-gram infixes in addition to n-gram prefixes and suffixes. Here again, however, the decision about what range of affixes is appropriate for a given language requires only superficial knowledge of its morphology.

5 Evaluation

Evaluation is based on the treebank data associated with each grammar, and a random training–test split of 20,000 training sentences and 1,013 test sentences in the case of the ERG, and 40,000 training sentences and 1,095 test sentences in the case of the JACY. This split is fixed for all models tested.

Given that the goal of this research is to acquire novel lexical items, our primary focus is on the performance of the different models at predicting the lexical type of any lexical items which occur only in the test data (which may be either novel lexemes or previously-seen lexemes occurring with a novel lexical type). As such, we identify all unknown lexical items in the test data and evaluate according to: **token accuracy** (the proportion of unknown lexical items which are correctly tagged: Acc$_U$); **type precision** (the proportion of correctly hypothesised unknown lexical entries: Prec); **type recall** (the proportion of gold-standard unknown lexical entries for which we get a correct prediction: Rec); and **type F-score** (the harmonic mean of type precision and type recall: F-score). We also measure the **overall token accuracy** (Acc) across all words in the test data, irrespective of whether they represent known or unknown lexical items.

5.1 Baseline: Unigram Supertagger

As a baseline model, we use a simple unigram supertagger trained based on maximum likelihood estimation over the relevant training data, i.e. the tag t_w for each token instance of a given word w is predicted by:

$$t_w = \arg\max_t p(t|w)$$

In the instance that w was not observed in the training data, we back off to the majority lexical type in the training data.

5.2 Benchmark: fnTBL

In order to benchmark our results with the CRF models, we reimplemented the supertagger model proposed by Baldwin (2005b) which simply takes fnTBL 1.1 (Ngai and Florian, 2001) off the shelf and trains it over our particular training set. fnTBL is a transformation-based learner that is distributed with pre-optimised POS tagging modules for English and other European languages that can be redeployed over the task of supertagging. Following Baldwin (2005b), the only modifications we make to the default English POS tagging methodology are: (1) to set the default lexical types for singular common and proper nouns to `n_-_c_le` and `n_-_pn_le`, respectively; and (2) reduce the threshold score for lexical and context transformation rules to 1. It is important to realise that, unlike our proposed method, the English POS tagger implementation in fnTBL has been fine-tuned to the English POS task, and includes a rich set of lexical templates specific to English.

Note that were only able to run fnTBL over the English data, as encoding issues with the Japanese proved insurmountable. We are thus only able to compare results over the English, although this is expected to be representative of the relative performance of the methods.

5.3 Results

The results for the baseline, benchmark fnTBL method for English and our proposed CRF-based supertagger are presented in Table 3, for each of the ERG and JACY. In order to gauge the impact of the lexical features on the performance of our CRF-based supertagger, we ran the supertagger first without lexical features (CRF$_{-\text{LEX}}$) and then with the lexical features (CRF$_{+\text{LEX}}$).

The first finding of note is that the proposed model surpasses both the baseline and FNTBL in all cases. If we look to token accuracy for unknown lexical types, the CRF is far and away the superior method, a result which is somewhat diminished but still marked for type-level precision, recall and F-score. Recall that for the purposes of this paper, our primary interest is in how successfully we are able to learn new lexical items, and in this sense the CRF appears to have a clear edge over the other models. It is also important to recall that our results over both English and Japanese have been achieved with only the bare minimum of lexical feature engineering, whereas those of FNTBL are highly optimised.

Comparing the results for the CRF with and without lexical features ($\text{CRF}_{\pm\text{LEX}}$), the lexical features appear to have a strong bearing on type precision in particular, for both the ERG and JACY.

Looking to the raw numbers, the type-level performance for all methods is far from flattering. However, it is entirely predictable that the overall token accuracy should be considerably higher than the token accuracy for unknown lexical items. A breakdown of type precision and recall for unknown words across the major word classes for English suggests that the $\text{CRF}_{+\text{LEX}}$supertagger is most adept at learning nominal and adjectival lexical items (with an F-score of 0.671 and 0.628, respectively), and has the greatest difficulties with verbs and adverbs (with an F-score of 0.333 and 0.395, respectively). In the case of Japanese, conjugating adjectives and verbs present the least difficulty (with an F-score of 0.933 and 0.886, respectively), and non-conjugating adjectives and adverbs are considerably harder (with an F-score of 0.396 and 0.474, respectively).

It is encouraging to note that type precision is higher than type recall in all cases (a phenomenon that is especially noticeable for the ERG), as this means that while we are not producing the full inventory of lexical items for a given lexeme, over half of the lexical items that we produce are genuine (with $\text{CRF}_{+\text{LEX}}$). This suggests that it should be possible to present the grammar developer with a relatively low-noise set of automatically learned lexical items for them to manually curate and feed into the lexicon proper.

One final point of interest is the ability of the CRF to identify multiword expressions (MWEs). There were no unknown multiword expressions in either the English or Japanese data, such that we can only evaluate the performance of the supertagger at identifying known MWEs. In the case of English, $\text{CRF}_{+\text{LEX}}$identified strictly continuous MWEs with an accuracy of 0.758, and optionally discontinuous MWEs (i.e. verb particle constructions) with an accuracy of 0.625. For Japanese, the accuracy is considerably lower, at 0.536 for continuous MWEs (recalling that there were no optionally discontinuous MWEs in JACY).

6 Conclusion

In this paper we have explored a method for learning new lexical items for HPSG-based precision grammars through supertagging. Our pseudo-likelihood conditional random field-based approach provides a principled way of learning a supertagger from tens-of-thousands of training sentences and with hundreds of possible tags.

We achieve start-of-the-art results for both English and Japanese data sets with a largely language-independent feature set. Our model also achieves performance at the type- and token-level, over different word classes and at multiword expression identification, superior to a probabilistic baseline and a transformation based learning approach.

Acknowledgements

We would like to thank Dan Flickinger and Francis Bond for support and expert assistance with the ERG and JACY, respectively, and the three anonymous reviewers for their valuable input on this research. The research in this paper has been supported by the Australian Research Council through Discovery Project grant number DP0663879, and also NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation

References

Timothy Baldwin. 2005a. Bootstrapping deep lexical resources: Resources for courses. In *Proc. of the ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition*, pages 67–76, Ann Arbor, USA.

Timothy Baldwin. 2005b. General-purpose lexical acquisition: Procedures, questions and results. In *Proc. of the 6th Meeting of the Pacific Association for Computational Linguistics (PACLING 2005)*, pages 23–32, Tokyo, Japan. (Invited Paper).

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–65.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar Matrix. An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammar. In *Proc. of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Francis Bond, Sanae Fujita, Chikara Hashimoto, Kaname Kasahara, Shigeko Nariyama, Eric Nichols, Akira Ohtani, Takaaki Tanaka, and Shigeaki Amano. 2004. The Hinoki treebank: A treebank for text understanding. In *Proc. of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 554–9, Hainan Island, China.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proc. of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 1240–7, Geneva, Switzerland.

Miriam Butt, Tracy Holloway King, Maria-Eugenia Nino, and Frederique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications, Stanford, USA.

Stanley F. Chen and Ronald Rosenfeld. 1999. A survey of smoothing techniques for maximum entropy models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proc. of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 282–8, Geneva, Switzerland.

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proc. of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece.

Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Oepen et al. (Oepen et al., 2002b).

Frederik Fouvry. 2003. *Robust Processing for Constraint-based Grammar Formalisms*. Ph.D. thesis, University of Essex.

Eric Joanis and Suzanne Stevenson. 2003. A general feature space for automatic verb classification. pages 163–70, Budapest, Hungary.

Anna Korhonen. 2002. *Subcategorization Acquisition*. Ph.D. thesis, University of Cambridge.

Mirella Lapata and Frank Keller. 2004. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks. pages 121–8, Boston, USA.

Stan Z. Li. 1994. Markov random field models in computer vision. In *ECCV (2)*, pages 361–370.

Rob Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proc. of the 6th Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, Taipei, Taiwan.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. 2003. *Japanese Morphological Analysis System ChaSen Version 2.3.3 Manual*. Technical report, NAIST.

Grace Ngai and Radu Florian. 2001. Transformation-based learning in the fast lane. In *Proc. of the 2nd Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL2001)*, pages 40–7, Pittsburgh, USA.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. 2002a. LinGO Redwoods: A rich and dynamic treebank for HPSG. In *Proc. of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.

Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors. 2002b. *Collaborative Language Engineering*. CSLI Publications, Stanford, USA.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, USA.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proc. of the 40th Annual Meeting of the ACL and 3rd Annual Meeting of the NAACL (ACL-02)*, Philadelphia, USA.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. of the 3rd International Conference on Human Language Technology Research and 4th Annual Meeting of the NAACL (HLT-NAACL 2003)*, pages 213–20, Edmonton, Canada.

Melanie Siegel and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proc. of the 3rd Workshop on Asian Language Resources and International Standardization*, Taipei, Taiwan.

Hans Uszkoreit. 2002. New chances for deep linguistic processing. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Gertjan van Noord. 2004. Error mining for wide-coverage grammar engineering. In *Proc. of the 42nd Annual Meeting of the ACL*, Barcelona, Spain.

Yi Zhang and Valia Kordoni. 2005. A statistical approach towards unknown word type prediction for deep grammars. In *Proc. of the Australasian Language Technology Workshop 2005*, pages 24–31, Sydney, Australia.

Yi Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.