General-Purpose Lexical Acquisition: Procedures, Questions and Results

Timothy Baldwin
Department of Computer Science and Software Engineering
University of Melbourne, VIC 3010, Australia
tim@csse.unimelb.edu.au

Abstract

We discuss a range of *in vitro* and *in vivo* approaches to deep lexical acquisition, and evaluate a representative sample of each in learning lexical items for a precision grammar. Evaluation focuses particularly on determining the effectiveness of each method at the token and type level, and over the four basic word classes of English. Each method is shown to have particular strengths and weaknesses but to have some part to play in the overall task of word learning.

1 Introduction

Over recent years, computational linguistics has benefitted considerably from advances in statistical modelling and machine learning, culminating in methods capable of deeper, more accurate automatic analysis, over a wider range of languages. Implicit in much of this work, however, has been the existence of **deep language resources** (DLR hereafter) of ever-increasing linguistic complexity, including lexical semantic resources (e.g. WordNet and FrameNet), precision grammars (e.g. the English Resource Grammar and the various ParGram grammars), and richly-annotated treebanks (e.g. PropBank and CCGbank).

Due to their linguistic complexity, DLRs are invariably constructed by hand and thus restricted in size and coverage. Our aim in this paper is to investigate a range of approaches for automatically expanding the coverage of an existing DLR, through the process of **deep lexical acquisition** (DLA hereafter). In this, we follow Baldwin (2005) in assuming a semi-mature DLR with a fixed inventory of lexical categories (whether they be semantic classes in a semantic ontology or lexical types in a precision grammar) and learning new lexical items by: (1) identifying which words in the original DLR a given unknown word is most similar to; and (2) analysing which lexical categories the similar words belong to, to derive the category membership for the unknown word.

We consider a range of DLA methods, which we categorise as being either *in vitro* or *in vivo* in their determination of lexical similarity. **In vitro** methods use a secondary lexical resource to model lexical similarity, whereas **in vivo** methods use some component of the target DLR for which we are attempting to learn new lexical items to model lexical similarity. In comparing these two approaches, we investigate their relative success in learning lexical items for different open word classes, namely nouns, verbs, adjectives and adverbs.

We demonstrate the proposed DLA methods relative to the English Resource Grammar (see Section 2.1), and in doing so assume the lexical types of the target DLR to be syntactico-semantic in nature. For example, we may predict that the word *dog* has a usage as an intransitive countable noun (`n_intr_le`,[1] cf. *The dog barked*), and also as a transitive verb (`v_np_trans_le`, cf. *It dogged my every step*).

The principal novel contribution of this paper is the evaluation of a supertagger-based *in vivo* approach to DLA, and its comparison with established *in vitro* approaches. The supertagger approach has

[1] All example lexical types given in this paper are taken directly from the English Resource Grammar – see Section 2.1.

In Proceedings of the Pacific Association for Computational Linguistics 2005, Tokyo, Japan.

an inherent advantage over the *in vitro* approaches in that it has direct access to token-level probabilities for each lexical type and lexical item, begging the question of whether there is any place for *in vitro* DLA if we are able to carry out *in vivo* DLA. In answering this question, we focus particularly on token- and type-based evaluation over a common task, and discover that while the supertagger method is superior to the *in vitro* method at the token level, the *in vitro* methods have a significant advantage over the supertagger at the type level. We also unearth some interesting idiosyncracies for particular word classes, and come to the conclusion that there is no one "best" way of going about DLA in the context of the target DLA task.

The remainder of this paper is structured as follows. Section 2 defines that target DLA task and reviews relevant resources. Section 3 outlines a range of *in vitro* DLA methods based on morphological, syntactic and ontological secondary LRs. Section 4 describes a range of *in vivo* techniques for performing DLA, and details a supertagging-based *in vivo* DLA method. Section 5 evaluates the different methods in the context of a DLA task over the English Resource Grammar.

2 Task Outline

This research compares a range of methods for DLA which can be run automatically given: (a) a pre-existing DLR which we wish to expand the coverage of; and (b) a set of secondary LRs/preprocessors for that language and/or some situated method which captures the behaviour of the DLR in action.

The DLA strategy we adopt in this research is based on that of Baldwin (2005), and uses some technique to arrive at a feature signature for each lexeme (in terms of intra- or inter-word context), and map this onto the system of lexical categories of choice via supervised learning. That is, we analyse the correlation between the feature signature of an unknown word and those of known words, and classify via a bootstrap process. This methodology can be applied to unannotated corpus data, for example, making it possible to tune a lexicon to a particular domain or register as exemplified in a particular repository of text. As it does not make any assumptions about the nature of the system of lexical types, we can apply it fully automatically to any DLR and feed the output directly into the lexicon without manual intervention or worry of misalignment. This is a distinct advantage when the inventory of lexical types is continually undergoing refinement, as is the case with the English Resource Grammar (see below).

2.1 English Resource Grammar

All experiments in this paper are targeted at the **English Resource Grammar** (ERG: Flickinger (2002), Copestake and Flickinger (2000)). The ERG is an implemented open-source broad-coverage precision Head-driven Phrase Structure Grammar (HPSG) developed for both parsing and generation. It contains roughly 10,500 lexical items, which, when combined with 59 lexical rules, compile out to around 20,500 distinct word forms.[2] Each lexical item consists of a unique identifier, a lexical type (one of roughly 600 leaf types organized into a type hierarchy with a total of around 4,000 types), an orthography, and a semantic relation. The grammar also contains 77 phrase structure rules which serve to combine words and phrases into larger constituents. Of the 10,500 lexical items, roughly 3,000 are multiword expressions.

To get a basic sense of the syntactico-semantic granularity of the ERG, the noun hierarchy, for example, is essentially a cross-classification of countability/determiner co-occurrence, noun valence and preposition selection properties. For example, lexical entries of `n_mass_count_ppof_le` type can be either countable or uncountable, and optionally select for a PP headed by *of* (example lexical items are *choice* and *administration*).

As our target lexical type inventory for DLA, we identified all open-class lexical types with at least 10 lexical entries, under the assumption that: (a) the ERG has near-complete coverage of closed-class lexical entries, and (b) the bulk of new lexical entries will correspond to higher-frequency lexical types. This resulted in the following breakdown:[3]

[2] All statistics and analysis relating to the ERG in this paper are based on the version of 11 June, 2004.

[3] Note that all results are over simplex lexemes only, and that we choose to ignore multiword expressions in this research.

Word class	*Lexical types*	*Lexical items*
Noun	28	3,032
Verb	39	1,334
Adjective	17	1,448
Adverb	26	721
Total	110	5,675

Note that it is relatively common for a lexeme to occur with more than one lexical type in the ERG: 22.6% of lexemes have more than one lexical item, and the average number of lexical types per lexeme is 1.12.

In evaluation, we assume we have prior knowledge of the basic word classes each lexeme belongs to (i.e. noun, verb, adjective and/or adverb), information which could be derived trivially from pre-existing shallow lexicons and/or the output of a tagger.

Recent development of the ERG has been tightly coupled with treebank annotation, and all major versions of the grammar are deployed over a common set of treebank data to help empirically trace the evolution of the grammar and retrain parse selection models (Oepen et al., 2002). We treat this as a held-out dataset for use in analysis of the *token* frequency of each lexical item, to complement analysis of *type*-level learning performance (see Section 5). We also use the treebank data directly in building a supertagger (see Section 4.2)

3 In Vitro Deep Lexical Acquisition

As the name suggests, *in vitro* DLA is based on analysis of lexemes in a context independent of the DLR we are looking to learn lexical items for. That is, we make use of a secondary LR or independent preprocessor to model lexical similarity, and use the target DLR only in classifying training instances.

In vitro DLA can be the only means available of performing DLA if we do not have access to annotated data for a given DLR. This would be the case if we were wanting to carry out DLR over a WordNet-style lexical ontology for which sense-annotated data did not exist, or over a precision grammar which did not have sufficient coverage to parse significant amounts of corpus data.

Below, we review past research on *in vitro* DLA, present the common classifier setup used in morphology- and syntax-based DLA, and review the models of morphology-, syntax- and ontology-based DLA utilised in this research.

3.1 Past research

The most widely-practised method of *in vitro* DLA extrapolates away from a DLR to corpus or web data, in analysing occurrences of words in template-based contexts which are predicted to correspond to particular lexical types. This most commonly takes the form of expert system-style DLA which is customised to (automatically) learning particular linguistic properties such as verb subcategorisation (e.g. Korhonen (2002) for English or Schulte im Walde (2003) for German, both of which employ an external parser to mine corpus data) or noun countability (e.g. Baldwin and Bond (2003a), which experiments with POS tagger, full text chunker and dependency parser to mine corpus data). Such an approach can also be used to learn not only the lexical type(s) for a predetermined lexeme, but also the lexemes themselves in the case of multiword expressions. For example, Baldwin (to appear) learns which verbs combine with intransitive prepositions to form verb particle constructions in English, and at the same time predicts the lexical type(s) of each such verb particle. One instance of a more general-purpose *in vitro* technique is the feature set proposed by Joanis and Stevenson (2003), which is shown to be applicable in a range of DLA tasks relating to English verbs.

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. This can occur as a one-step process, in mining lexical items directly from a DLR (e.g. a machine-readable dictionary (Sanfilippo and Poznański, 1992) or WordNet (Daudé et al., 2000)), or two-step process in reusing an existing system to learn lexical properties in one format and then mapping this onto the DLR of choice (e.g. Carroll and Fang (2004) for verb subcategorisation learning).

3.2 Classifier design

The general procedure we adopt for *in vitro* DLA (as applied to morphology- and syntax-based DLA) is taken from Baldwin (2005): we generate a feature signature for each word contained in a given secondary LR, take the subset of lexemes contained in

the original DLR as training data, and learn lexical items for the remainder of the lexemes through supervised learning. In all cases other than ontology-based DLA, we employ TiMBL 5.0 (Daelemans et al., 2003) as our learner, using the IB1 k-NN algorithm with k = 9 throughout.[4] We additionally employ the feature selection method of Baldwin and Bond (2003b), which generates a combined ranking of all features in descending order of "informativeness" and skims off the top-N features for use in classification; N was set to 100 in all experiments.

As observed above, a significant number of lexemes in the ERG occur in multiple lexical items. If we were to take all lexical type combinations observed for a single lexeme, the total number of lexical "multi"-types would be 451, of which 284 are singleton classes. Based on the sparseness of this data and also the findings of Baldwin and Bond (2003b) over a countability learning task, we choose to carry out DLA via a suite of 110 binary classifiers, one for each lexical type. One potential shortcoming of this architecture is that a given lexeme can be negatively classified by all unit binary classifiers and thus not assigned any lexical items. In this case, we fall back on the majority-class lexical type for each word class the word has been pre-identified as belonging to.

3.3 Morphology-based Deep Lexical Acquisition

Our first feature representation is based on a highly simplistic model of word morphology: it takes a simple word list and converts each lexeme into a character n-gram representation. Specifically, we generated all 1- to 6-grams for each lexeme, and applied a series of filters to: (1) filter out all n-grams which occurred less than 3 times in the lexicon data; and (2) filter out all n-grams which occur with the same frequency as larger n-grams they are proper substrings of. We then select the 3,900 character n-grams with highest saturation across the lexicon data (see Section 3.2).

In Baldwin (2005) we additionally experimented with derivational morphology, but found simple character n-grams to offer superior performance.

3.4 Syntax-based Deep Lexical Acquisition

Syntax-based DLA takes a raw text corpus and preprocesses it with a part of speech (POS) tagger. It then extracts a set of 39 feature types based on analysis of the token occurrences of a given lexeme, and filters over each feature type to produce a maximum of 50 feature instances of highest saturation (e.g. if the feature type is the word immediately proceeding the target word, the feature instances are the 50 words which proceed the most words in our lexicon). The feature signature associated with a given word will thus have a maximum of 3,900 items ($39 \times 50 \times 2$).

We learn the corpus feature values from the written component of the British National Corpus ($\sim$98M tokens: Burnard (2000)), which we tag with a Penn treebank-style tagger custom-built using fnTBL 1.0 (Ngai and Florian, 2001); we further lemmatise the output of the tagger using morph (Minnen et al., 2000). Note that the only corpus annotation we make use of is sentence tokenisation, and that the POS tagger is run automatically over the raw corpus data.

The feature types used with the tagger are detailed in Table 1, where the position indices are relative to the target word (e.g. the word at position -2 is two words to the left of the target word, and the POS tag at position 0 is the POS of the target word). All features are relative to the POS tags and words in the immediate context of each token occurrence of the target word. "Bi-words" are word bigrams (e.g. bi-word $(1, 3)$ is the bigram made up of the words one and three positions to the right of the target word); "bi-tags" are, similarly, POS tag bigrams.

In Baldwin (2005) we tested alternate preprocessors, in the form of a full text chunker and dependency parser, and also smaller-sized corpora. We present only the POS tagger-based results in this paper as we found there to be very little difference in performance between the three systems, and use only the BNC as we found it to be (marginally) superior to the other corpora tested.

3.5 Ontology-based Deep Lexical Acquisition

The final *in vitro* DLA method we explore is based on the hypothesis that there is a strong correla-

[4] We also experimented with bsvm and SVMLight, and a maxent toolkit, but found TiMBL to be superior overall, we hypothesise due to the tight integration of continuous features in TiMBL.

Feature type	*Positions*	*Total*
POS tag	$(-4, -3, -2, -1, 0, 1, 2, 3, 4)$	9
Word	$(-4, -3, -2, -1, 1, 2, 3, 4)$	8
POS bi-tag	$(\,(-4,-1),(-4,0),(-3,-2),(-3,-1),(-3,0),(-2,-1),(-2,0),$ $(-1,0),(0,1),(0,2),(0,3),(0,4),(1,2),(1,3),(1,4),(2,3)\,)$	16
Bi-word	$((-3,-2),(-3,-1),(-2,-1),(1,2),(1,3),(2,3))$	6
		39

Table 1: Feature types used in syntax-based DLA

tion between the semantic and syntactic similarity of words, a claim which is best exemplified in the work of Levin (1993) on diathesis alternations. In our case, we model word similarity using the topology of the lexical ontology, and learn the syntactic behaviour of novel words relative to semantically-similar words for which we know the lexical types. We use WordNet 2.0 (Fellbaum, 1998) to determine word similarity, and for each sense of the target word in WordNet: (1) construct the set of "semantic neighbours" of that word sense, comprised of all synonyms, direct hyponyms and direct hypernyms; and (2) take a majority vote across the lexical types of the semantic neighbours which occur in the training data. Note that this diverges from the learning paradigm adopted for the morphology- and syntax-based DLA methods in that we use a simple voting strategy rather than relying on an external learner to carry out the classification. The full set of lexical entries for the target word is generated by taking the union of the majority votes across all senses of the word, such that a polysemous lexeme can potentially give rise to multiple lexical entries. This learning procedure is based on the method used by van der Beek and Baldwin (2004) to learn Dutch countability.

As for the suite of binary classifiers, we fall back on the majority class lexical type as the default in the instance that a given lexeme is not contained in WordNet 2.0 or no classification emerges from the set of semantic neighbours.

4 In Vivo Deep Lexical Acquisition

In vivo DLA directly leverages the target DLR to learn new lexical items. This is most commonly performed via an annotated corpus, e.g. a sense-annotated corpus in the case of a lexical ontology, or parsed data of some description (e.g. a treebank) in the case of a precision grammar. Indeed, all *in vivo* techniques discussed in this paper make use of corpus data. Note that we do not consider raw corpus data to be a secondary LR as long as any filtering/analysis of the data is performed based on techniques derived directly from the target DLR, independent of any external pre-processor. That is, DLA which aligns templates from raw corpus data directly with classes in the target DLR is considered to be *in vivo* DLA.

In the following sections, we review past research on *in vivo* DLA, before describing how a supertagger can be used to learn new lexical items for our target DLA task.

4.1 Past research

The *in vivo* approach to DLA is perhaps best exemplified by the research of Fouvry (2003), targeted at precision grammars. 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 pre-processing 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).

One further approach to *in vivo* DLA which is immediately relevant to this research is **supertagging** (Bangalore and Joshi, 1999; Clark, 2002). 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. We consider supertagging to be a form of *in vivo* DLA as the supertagger is a crude surrogate for a parser for the target DLR, in modelling local dependencies between words and lexical types by way of word context. Also, unlike conventional POS taggers or chunk parsers, e.g., supertaggers operate over the native set of lexical types associated with a given DLR. They have the advantage over parsers that they are quick to train and run, and are tailored to handle unknown words in a robust manner.

4.2 Supertagger-based Deep Lexical Acquisition

The basic procedure of building a supertagger is identical to that for building a conventional POS tagger. Indeed, in this research we implemented our supertagger using an off-the-shelf trainable POS tagger, in the form of fnTBL 1.1 (Ngai and Florian, 2001). The supertagger was trained over the Redwoods treebank, that is roughly 11,000 sentence-tokenised dialogue turns from the Verbmobil corpus which have been parsed with the ERG and hand-disambiguated. In a slight divergence from the *in vitro* methods, we use 4-fold cross validation in training and testing our supertagger, based on the 4 preexisting partitions of the Redwoods data corresponding to distinct sections in the original Verbmobil corpus. Our motivation in this was that 10-fold cross-validation was going to lead to too few unknown words on each iteration, and there was no more principled way of partitioning the data.

In building our supertagger, we read the lexical type of each lexeme directly off the gold-standard parse for the sentence in question, producing a unique "supertag" for each word token. For each iteration of cross-validation, we then trained fnTBL 1.1 over the training data, with two minor modifications over the default POS tagging methodology: (1) the default lexical types for singular common and proper nouns were set to `n_intr_le` and `n_proper_le`, respectively; and (2) the threshold score for lexical and context transformation rules was reduced to 1,[5] due to the limited quantity of training data.

The concept of an "unknown word" becomes somewhat ill-defined in a supertagger context. First, it can refer to a word form that has occurred in the training data but not with the full range of lexical types (e.g. *dogs* may have occurred as `n_intr_le` but not `v_np_trans_le`); here the supertagger will tend to make the closed world assumption and be unable to dynamically identify this novel occurrence of the word. Second, it can refer to word forms which occur only in the test data but have been observed in alternate lexical forms in the training data (e.g. *dog* occurs in the training data but not *dogs*). Here, the supertagger will often be able to implicitly or explicitly make use of its knowledge of alternate word forms of the same lexeme in predicting the lexical items. Third, it can refer to a *lexeme* which occurs for the first time in the test data, possibly with multiple word forms. It is this third case of unknown lexemes we are most interested in in this research, as this is what constitutes an unknown word in the context of the ERG: in the first instance, the ERG applies the closed world assumption similarly to supertaggers, and in the second instance, the ERG has in-built morphological processing to predict novel word forms for a known lexeme.

We identify the unknown lexemes on each iteration of cross-validation via the Redwoods data. The chart output of each sentence contains the lexical type and lexical ID of each word token, and from the lexical ID it is possible to identify the base lexeme. We thus generate a list of all lexemes in the training and test data, and evaluate the performance of the supertagger over those lexemes which are not found

[5] For all transformation rules, the difference between "good" and "bad" transformations (transformations which produce correct and incorrect tag assignments, respectively) must be strictly greater than this threshold; the default setting is 2, but a setting of 1 was found to produce significantly better results.

in the training data. The average number of unknown lexemes on each iteration of cross-validation was 147.

5 Evaluation

In the case of the *in vitro* methods, we are able to simulate unknown lexemes via 10-fold stratified cross-validation over the 5,675 open-class lexical items of the ERG described in Section 2.1. In the case of the supertagger, on the other hand, we carry out 4-fold cross-validation across the four partitions of the Redwoods data, and allow the corpus to dictate the composition of unknown lexemes on each iteration.

In each case, we calculate the **type precision** (the proportion of correctly hypothesised lexical entries) and **type recall** (the proportion of gold-standard lexical entries for which we get a correct hit) for unknown lexemes, which we roll together into the **type F-score** (the harmonic mean of the two) relative to the gold-standard ERG lexicon. We also measure the **token accuracy** for the lexicon derived from each method, relative to the Redwoods treebank: in the case of the *in vitro* methods, this is calculated over all token instances of unknown lexemes in the full treebank, whereas with the supertagger, it is calculated independently for each test partition. The token accuracy represents a weighted version of type precision, relative to the distribution of each lexical item in a representative text sample, and provides a crude approximation of the impact of each DLA method on parser coverage. That is, it gives more credit for a method having correctly hypothesised a commonly-occurring lexical item than a low-frequency lexical item, and no credit for having correctly identified a lexical item not occurring in the corpus.

The overall results are presented in Figure 1, which are then broken down into the four open word classes in Figures 2–5. The baseline method (*Base*) in each case is a simple majority-class classifier, which generates a unique lexical item for each lexeme pre-identified as belonging to a given word class of the following type:

Word class	Majority-class lexical type
Noun	`n_intr_le`
Verb	`v_np_trans_le`
Adjective	`adj_intrans_le`
Adverb	`adv_int_vp_le`

In each graph, we present the type F-score and token accuracy for each method, and mark the best-performing method in terms of each of these evaluation measures with a star ($\star$).

Looking first at the combined results over all lexical types (Figure 1), the most successful method in terms of type F-score is syntax-based DLA (type F-score = 0.630), and the most successful method in terms of token accuracy is the supertagger (token accuracy = 0.788). The disparity in type F-score and token accuracy for the *in vitro* methods and the supertagger is striking: all *in vitro* methods significantly outperform the supertagger in type F-score, at or above the baseline performance, while the token accuracy for the supertagger is over 20% better than the *in vitro* methods in absolute terms. The reason for the apparently anomalous performance of the supertagger is that it operates at high type precision but very low type recall, generally predicting the commonly-occurring lexical items for a given unknown lexeme with high reliability but failing to predict low-occurrence lexical items. The *in vitro* techniques obviously do not have direct access to token frequencies, and are hence disadvantaged in token-level evaluation.

Turning next to the results for the proposed methods over nouns, verbs, adjectives and adverbs (Figures 2–5, respectively), we observe some interesting effects, as outlined below for each individual method.

Morphology-based DLA hovers around baseline performance for all word classes except adjectives, where it produces the highest type F-score and second-highest token accuracy of all methods. That is, similarly-spelled adjectives tend to have similar syntax and semantics as evaluated at both the type and token levels, a somewhat surprising finding.

Syntax-based DLA leads to the highest type F-score for nouns, verbs and adverbs, and the highest token accuracy for adjectives.

Ontology-based DLA is below baseline in terms of type F-score for all word classes, but results in

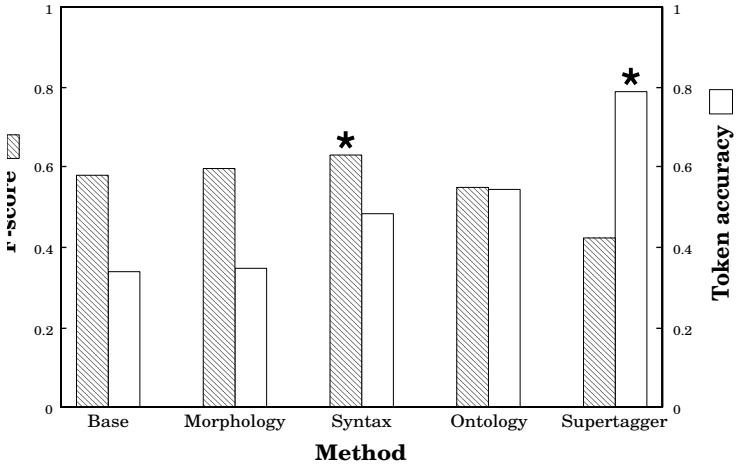

Figure 1: Results for the proposed deep lexical acquisition methods over ALL lexical types

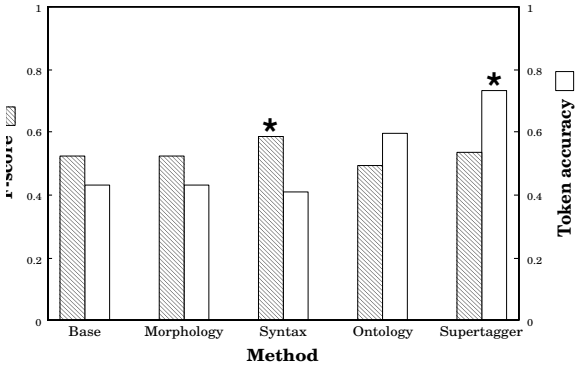

Figure 2: Results for the proposed deep lexical acquisition methods over NOUN lexical types

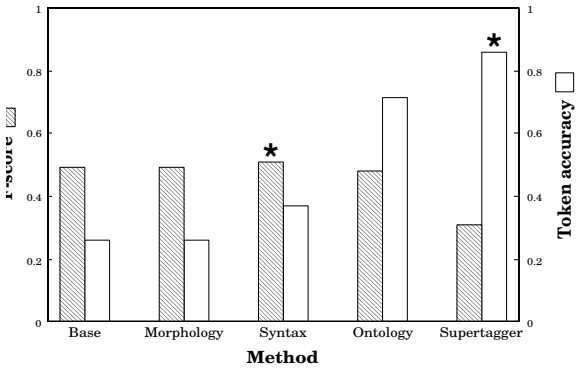

Figure 3: Results for the proposed deep lexical acquisition methods over VERB lexical types

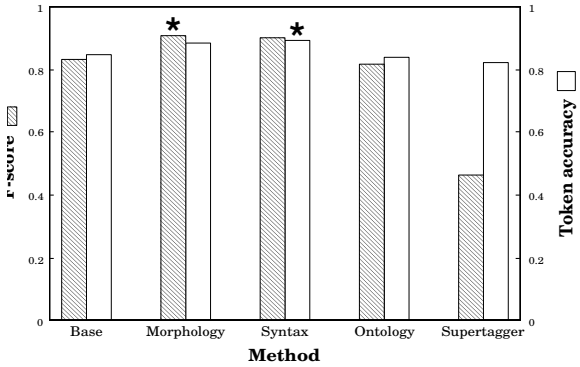

Figure 4: Results for the proposed deep lexical acquisition methods over ADJECTIVE lexical types

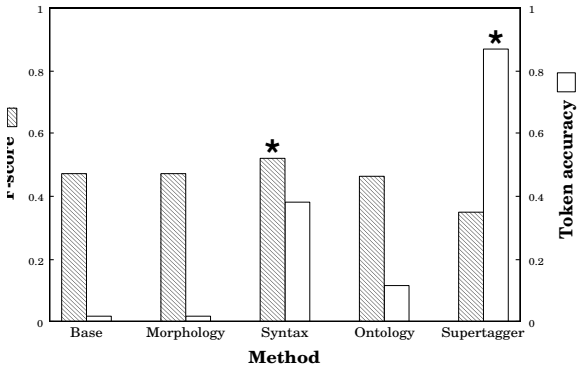

Figure 5: Results for the proposed deep lexical acquisition methods over ADVERB lexical types

Note: Base = baseline, Morphology = morphology-based DLA with character n-grams, Syntax = syntax-based DLA with POS tagging, Ontology = ontology-based DLA based on WordNet, and Supertagger = the supertagger trained over the Redwoods data

the highest token accuracy of all *in vitro* methods for nouns and verbs. These results require some qualification: ontology-based DLA tends to be liberal in its generation of lexical items, giving rise to over 20% more lexical items than the other methods (7,307 vs. 5-6000 for the other methods) and proportionately low type precision; this effect is particularly noticeable for word classes with high mean polysemy, such as verbs (average senses/word = 5.0) and nouns (average senses/word = 3.3). This correlates with an inherent advantage in terms of token accuracy, which we have no way of balancing up in our token-based evaluation, as the treebank data offers no insight into the true worth of false negative lexical items (i.e. we have no way of distinguishing between unobserved lexical items which are plain wrong from those which are intuitively correct and could be expected to occur in alternate sets of treebank data). We leave investigation of the impact of these extra lexical items on the overall parser performance (in terms of chart complexity and parse selection) as an item for future research.

Another noteworthy feature of Figures 2–5 is the huge variation in absolute performance across the word classes: adjectives are very predictable, with a majority class-based baseline type F-score of 0.832 and token accuracy of 0.847; adverbs, on the other hand, are similar to verbs and nouns in terms of their baseline type F-score (at 0.471), but the adverbs that occur commonly in corpus data appear to belong to less-populated lexical types (as seen in the baseline token accuracy of a miniscule 0.017). Nouns appear the hardest to learn in terms of the relative increment in token accuracy over the baseline. Verbs are extremely difficult to get right at the type level, but the supertagger is highly adept at getting the commonly-occurring lexical items right.

To summarise these findings, the supertagger is far and away the superior method at the token level, whereas syntax-based DLA is the most solid performer at the type level.

6 Conclusion

We have categorised deep lexical acquisition research according to the *in vitro*/*in vivo* dichotomy, presented a representative selection of methods for each category, and discussed the relative advantages and disadvantages of each. We described three basic paradigms for *in vitro* deep lexical acquisition, based on morphological, syntactic and ontological language resources, and presented supertagging as an instantiation of *in vivo* deep lexical acquisition. All methods were road-tested over a DLA task for a precision grammar of English. We discovered surprising variation in the results for the different DLA methods, with the *in vitro* methods generally performing well at the type level and the *in vivo* method excelling at the token level. Each component learning method was found to have strengths and weaknesses over different word classes, with the best overall methods being syntax-based DLA and supertagging.

The results presented in this paper are based on one particular language (English) and a very specific style of DLR (a precision grammar, namely the English Resource Grammar), so some caution must be exercised in extrapolating the results too liberally over new languages/DLA tasks. In future research, we are interested in carrying out experiments over other languages and alternate DLRs, to determine how well these results generalise and formulate alternate strategies for DLA.

Acknowledgements

This research was supported in part by NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation.

References

Timothy Baldwin and Francis Bond. 2003a. Learning the countability of English nouns from corpus data. In *Proc. of the 41st Annual Meeting of the ACL*, pages 463–70, Sapporo, Japan.

Timothy Baldwin and Francis Bond. 2003b. A plethora of methods for learning English countability. In *Proc. of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 73–80, Sapporo, Japan.

Timothy Baldwin. 2005. 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. to appear. The deep lexical acquisition of English verb-particle constructions. *Computer*

Speech and Language, Special Issue on Multiword Expressions. (Accepted for publication 12/8/2004; 21 pages).

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

Lou Burnard. 2000. *User Reference Guide for the British National Corpus*. Technical report, Oxford University Computing Services.

John Carroll and Alex Fang. 2004. The automatic acquisition of verb subcategorisations and their impact on the performance of an HPSG parser. In *Proc. of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 107–14, Sanya City, China.

Stephen Clark. 2002. Supertagging for combinatory categorial grammar. In *Proc. of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pages 19–24, Venice, Italy.

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.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2003. *TiMBL: Tilburg Memory Based Learner, version 5.0, Reference Guide*. ILK Technical Report 03-10.

Jordi Daudé, Lluis Padró, and German Rigau. 2000. Mapping WordNets using structural information. In *Proc. of the 38th Annual Meeting of the ACL*, Hong Kong, China.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*. CSLI Publications, Stanford, USA.

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. In *Proc. of the 10th Conference of the EACL (EACL 2003)*, 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.

Beth Levin. 1993. *English Verb Classes and Alterations*. University of Chicago Press, Chicago, USA.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of the first International Natural Language Genration Conference*, Mitzpe Ramon, Israel.

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. 2002. LinGO Redwoods: A rich and dynamic treebank for HPSG. In *Proc. of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.

Antonio Sanfilippo and Victor Poznański. 1992. The acquisition of lexical knowledge from combined machine-readable dictionary sources. In *Proc. of the 3rd Conference on Applied Natural Language Processing (ANLP)*, pages 80–7, Trento, Italy.

Sabine Schulte im Walde. 2003. *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart. Published as AIMS Report 9(2).

Leonoor van der Beek and Timothy Baldwin. 2004. Crosslingual countability classification with EuroWordNet. In *Papers from the 14th Meeting of Computational Linguistics in the Netherlands*, pages 141–55, Antwerp, Belgium. Antwerp Papers in Linguistics.