Parser Adaptation to the Biomedical Domain without Re-Training

Jeff Mitchell
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
jeff.mitchell@ed.ac.uk

Mark Steedman
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
steedman@inf.ed.ac.uk

Abstract

We present a distributional approach to the problem of inducing parameters for unseen words in probabilistic parsers. Our KNN-based algorithm uses distributional similarity over an unlabelled corpus to match unseen words to the most similar seen words, and can induce parameters for those unseen words without retraining the parser. We apply this to domain adaptation for three different parsers that employ fine-grained syntactic categories, which allows us to focus on modifying the lexicon, while leaving the structure of the parser itself intact. We demonstrate uplifts for dependency recovery of 2%-6% on novel vocabulary in biomedical text.

1 Introduction

Parsing is an important component in many NLP applications. Shallower analyses may allow the discovery of local relations, but to handle the full complexity of speech and text requires knowledge of the hierarchical structures that parsers are designed to uncover. This is particularly true of long range dependencies such as that between *activities* and *decreased* in *the specific synthetic activities of electrophoretically purified myosin heavy chain decreased*. Such dependencies have proven to be useful features in many text mining and knowledge extraction applications, for example identifying biomarkers in the biomedical literature (Seoud and Mabrouk, 2013) or extracting family history from clinical text (Lewis et al., 2011).

Correctly identifying the dependencies within a string of words is generally based on finding the most probable structure over them, and this in turn requires knowing what sort of relations each word is likely to enter into. Unfortunately, gold standard training data, annotated with these syntactic relations, is generally in short supply. The vocabulary for which we have explicitly seen examples of the type of dependencies each word supports is therefore typically small and performance on real data is often degraded in handling out-of-vocabulary items.

Although the Penn Treebank has been a vital tool in the development and evaluation of parsing technology, providing a standard dataset for comparison of parsers, practical application of these techniques usually requires adaptation to new domains. Rimell and Clark (2009), for example, examine the adaptation of a WSJ-trained CCG parser to the biomedical domain. The divergence between these two domains, news and biology, is manifest in terms of both vocabulary and also stylistic differences in the prevalence of various syntactic structures. For example, biomedical writing eschews personal pronouns and tolerates long sequences of noun modifiers, whereas the style of news articles tends to reverse these preferences. Rimell and Clark's (2009) approach to adapting to these differences is based on retraining elements of the model using biomedical texts which have been hand-tagged with gold-standard tags. While this is undoubtedly effective, achieving an overall improvement of F-score of over 5%, it requires a considerable commitment of skilled resources to manually annotate a substantial corpus with the linguistically correct tags.

Here, we consider a distributional approach to domain adaptation using the information about syntactic structure that is implicit in raw text. We estimate parameters for unseen words using a KNN approach that matches them to the nearest seen words and averages over their parameters. We explore a number of different approaches to measuring distributional similarity and find that vectors based on counts of occurrence within ngram contexts give the best results. Bag-of-word approaches and neural embeddings, which have worked well for semantic tasks, do not appear to

Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis (Louhi), pages 79–89,
Lisbon, Portugal, 17 September 2015.

capture the information about syntactic similarity that this task requires.

Our use of ngram contexts is inspired by psycholinguistic research into the acquisition of syntactic categories. Cartwright and Brent (1997), for example, consider how children might use a word's distribution across a range of templates, such as $\langle the\ XXX\ is\ good \rangle$, to infer its syntactic properties. They show, in simulations, that such distributional information can be used to infer syntactic categories from child-directed speech. Mintz (2003) analyses distributions over a simpler type of template, which he calls a frequent frame, consisting of a pair of common lexical items flanking a word of interest, e.g. $\langle you\ XXX\ it \rangle$ or $\langle the\ XXX\ is \rangle$. In addition to showing how such distributional information can be used to induce categories, he also discusses the evidence that adults and children are sensitive to these frames. Redington et al. (1998) consider even simpler contexts, based simply on bigram colocations, e.g. $\langle the\ XXX \rangle$. Pinker (Pinker, 1987), on the other hand, has long contested the possibility of using such distributional information to acquire valid grammatical categories, and proposes instead that grammatical categories are bootstrapped using semantic knowledge.

While the patterns and templates described above can be used to characterise a word's behaviour in terms of concrete occurrences in specific contexts, neural networks have recently become popular as a means to create more abstract representations. In this case, as the network adapts to the data, representations are learned that embed discrete inputs in a continuous space defined by its internal states. Researchers have been interested in the nature of such internal representations for some time (e.g., Small et al., 1995; Joanisse and Seidenberg, 1999). However, it has now become practical to induce such embeddings from large quantities of text and employ them in linguistic applications. For example, Tsuboi (2014) and Collobert et al. (2011) apply neural representations to POS tagging, and this suggests that at least some useful information about the syntax of unseen words might be gained from this source.

While POS tags can provide a coarse-grained description of words' syntactic behaviour, accurate parsing typically requires finer-grained detail. We can distinguish between two approaches, which may be combined, to specifying this additional level of detail. The first approach simply makes use of finer-grained syntactic categories, either instead of or in addition to POS tags (Steedman, 2000; Klein and Manning, 2003b; Petrov et al., 2006). These categories can then determine the missing information about the dependencies a word will take part in, such as whether a verb is intransitive or whether it takes prepositional arguments. The second approach instead increases the granularity of the production rules, by conditioning the probabilities on the heads of the phrases involved (Charniak, 2001; Collins, 2003). In this way, words are associated with probabilities for the structure of phrases that they head, determining, for example, the types of object that a verb phrase expands into.

Although the two approaches are compatible, a significant difference makes the former more conducive to our purposes. Enhancing the granularity of the syntactic categories results in a much richer lexicon containing more information about how words behave syntactically. In principle, this should lead to an enlargement of the lexicon having a greater impact on performance by itself. In the latter approach, of lexicalising the production rules, expanding the vocabulary of the parser may be much more complicated, requiring modifications throughout the model. In contrast our approach simply adds new entries to the lexicon without the need to retrain the parser. In fact, our approach does not even require full sentences and can be applied to an unlabelled corpus of ngram counts.

Our KNN approach and the three parsers we modify are described in Sections 2 and 3 respectively. We then use a biomedical dependency recovery task, specified in Section 4, to evaluate the performance of the modified parsers, as reported in Section 5.

2 Approach

Our approach is based on the assumption that words with similar syntactic properties should have similar distributional characteristics. We evaluate both neural embeddings and also raw context frequencies as the basis for measuring distributional similarity. These context vectors have components which correspond to occurrences within a corpus of raw biomedical text and we employ both SENNA (Collobert et al., 2011) and Skip-gram (Mikolov et al., 2013) em-

beddings. In all cases, we induce parameters for unseen words by averaging the parameters from the k nearest neighbours seen in the training data.

2.1 Context Vectors

Distributional similarity is here based on comparing vectors that are constructed from raw context counts. We considered two approaches to defining these contexts: ngrams and bags-of-word (BOW).

The ngram approach counts occurrences in 2gram, 3gram and 4gram contexts that are intended to emphasise syntactic - as opposed to semantic - characteristics, following the structure of templates and frames proposed by e.g. Cartwright and Brent (1997), Mintz (2003) and Redington et al. (1998). Thus our 2gram contexts have two forms that distinguish occurrence on the left from occurrence on the right: $\langle left\_token\ XXX \rangle$ and $\langle XXX\ right\_token \rangle$. The 3gram contexts are equivalent to Mintz's (2003) frequent frames: $\langle left\_token\ XXX\ right\_token \rangle$. And the 4gram contexts extend this frame to the right, mimicking the form of templates described by Brent (1991) and Cartwright and Brent (1997): $\langle left\_token\ XXX\ right\_token_1\ right\_token_2 \rangle$.

The BOW approach ignores the sequential information contained in the ngram contexts and relies instead on counts of individual words that occur anywhere in 5 word-windows each side of a target word.

In each case, we built distributional vectors using the most common of these contexts, with vector components based on a ratio of probabilities.

$$v_i = \frac{p(c_i|w_t)}{p(c_i)} = \frac{freq_{i,t} \cdot freq_{total}}{freq_i \cdot freq_t} \qquad (1)$$

where c_i is the ith context, w_t is the target word, $freq_{i,t}$ is the count of the number of times w_t occurs in context c_i, $freq_i$ is the overall count of the number of times context c_i occurs with all words, $freq_t$ is the overall count for w_t in all contexts and $freq_{total}$ is the total count for all words in all contexts. Target words with $freq_t < 10$ were discarded as containing too little useful information.

The distance between two vectors, $\mathbf{u}$ and $\mathbf{v}$, was measured in terms of the city block metric:

$$dist(\mathbf{u}, \mathbf{v}) = \sum_i |u_i - v_i| \qquad (2)$$

This appeared to work more effectively on sparse vectors than the more usual cosine metric. We built these representations on a corpus of 1.2 billion words of titles and abstracts from the Medline database.

2.2 SENNA

Collobert et al. (2011) trained a neural net language model on a snapshot of the English Wikipedia ($\approx$ 631M words) and published the feature vectors[1] induced for each word in the first hidden layer of the network. They showed that these embeddings are useful in enhancing the performance of a number of tasks, including POS tagging and semantic role labelling. Using these representations as features, Bansal et al. (2014) obtained improvements in dependency recovery in the MST Parser (McDonald and Pereira, 2006).

Andreas and Klein (2014) also used these embeddings on a number of tasks, including an attempt to expand the vocabulary of the Berkeley Parser by matching unseen words to the nearest word already in the lexicon. However, instead of inducing parameters for the new vocabulary they simply replaced unseen words with their seen matches in the input. Unfortunately they did not find a reliable benefit from this approach.

Like the context vectors described above, the SENNA representations were derived from large quantities of raw text and reflect the distributional behaviour of words in that data. However, unlike our context vectors, which have components derived from explicit distributional contexts, the components of their neural embeddings are abstract dimensions whose values derive from the optimization of a particular mathematical model. In this case the form of this model was based on distinguishing between real 11-word phrases drawn from the unlabelled corpus and an incorrect phrase which had the central word replaced with a randomly chosen item. The model tries to maximise the difference between these two phrases in terms of scores which are a nonlinear function of the vectors representing the words they contain.

Training involved stochastic gradient ascent optimisation of an objective function based on a ranking criterion for the two phrase scores, and resulted in each word within a 100,000 word vocabulary being assigned a vector representation. The published embeddings are of dimension 50 and we measured the similarity of these vectors in terms of the cosine measure:

[1] http://ronan.collobert.com/senna/

$$dist(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|} \quad (3)$$

2.3 Skip-gram

Like the SENNA model, the Skip-gram model (Mikolov et al., 2013) is trained to differentiate between the correct central word of a phrase and a random replacement, which they refer to as negative sampling. Unlike SENNA, however, the Skip-gram model tries to make this prediction using only a single one of the surrounding words at a time and ignores the ordering of those words, i.e. taking a bag-of-words approach to context.

The published 300-dimensional vectors[2] were trained on 100B words of Google News text using stochastic gradient ascent, and cover a vocabulary of 3M words. We also retrained the same 300-dimensional model on our 1.2 billion word unlabelled biomedical corpus, giving a vocabulary of around 1M words. In both cases, we measured similarity using the cosine metric, Equation 3.

2.4 KNN Parameter Induction

Our approach to inducing parser parameters for unseen words is a form of k-nearest-neighbor induction.[3] Specifically, we constructed parameters for unseen words by finding the most similar words in the lexicon, using the distributional measures described above, and then averaging over their existing parameters in the parsing model. We did this for each parser, varying the dimensions of the context vectors, and the number of nearest neighbours to find the optimal model. To ensure that the parameters that we average over are well-estimated and reliable, we only consider words that appear more than a hundred times in the Penn Treebank when finding the nearest neighbours.

3 The Parsers

We extend the vocabulary of three parsers, all of which make use of fine-grained lexical categories. The first of these parsers induces sub-categories beneath the level of POS-tags during training while the other two require hand-annotation of the categories in the training data. In all cases, we modify the parser merely by inserting new items, along with their tag parameters, into the lexicon while leaving the rule probabilities in the rest of the parser unchanged. Sections 3.1, 3.2 and 3.3 outline these parsers, focusing particularly on the contents of the lexicon which our methods modify as decribed in Section 2.

3.1 The Berkeley Parser

While an unlexicalized parser that uses syntactic categories based solely on the symbols found in the Penn Treebank will generally perform poorly, a number of results show that refining these categories can substantially improve performance. Klein and Manning (2003b), for example, show that the performance of an unlexicalised model can be substantially improved by splitting the existing symbols down into finer categories. Their subcategorizations were developed by hand based on linguistic intuitions and a careful error analysis. The Berkeley Parser[4] (Petrov et al., 2006), in contrast, is based on a method for automatically finding useful subcategorizations during training by splitting and merging the original nodes.

The model is an unlexicalized generative PCFG, but the granularity of the terminal and non-terminal categories found in training give it a much greater sensitivity to the syntactic behaviour of words and phrases than is possible using standard POS tags. The lexicon specifies each word's association to the terminal categories, and the rest of the parser is entirely unlexicalized. Parsing is complicated by the large number of syntactic categories which threaten to make standard techniques infeasible, due to the size of the search space and also even just the amount of memory required to hold the chart. However, the hierarchical structure resulting from the split-merge process enables a form of coarse to fine pruning that makes the problem tractable (Petrov and Klein, 2007). Training is based on the EM algorithm along with 6 cycles of splitting each symbol into two and remerging the 50% of sub-symbols carrying the least information. Output from the Berkeley Parser consists of trees labelled with the original Penn Treebank symbols, and we use the EnglishGrammat-

[2] https://code.google.com/p/word2vec/

[3] We also evaluated Support Vector Regression as a means of inducing parameters, but we found it to be less effective. Although the characteristics of SVMs do in general make them powerful modelling tools, this particular task required us to use one SVM model for every parameter type to be induced (e.g. $\approx$ 400 CCG categories). In fact, the requirement to optimise the C and gamma hyper-parameters resulted in evaluation of about 100 models per parameter (e.g. $\approx$ 40,000 models). In contrast, the KNN approach induces all the parameters in one single model, producing a much more constrained problem, which probably contributes to its superior generalisation in this case.

[4] https://code.google.com/p/berkeleyparser/

icalStructure class from the Stanford Parser[5] to convert the trees to Stanford-style dependencies. Out-of-vocabulary items are handled by a process that uses orthography and sentence position to estimate probabilities for unseen words.

Expanding the lexicon of this model using our KNN method is complicated by the fact that it is generative, so that inserting new vocabulary with non-zero probabilities requires adjusting the probabilities of everything else in the lexicon to maintain normalization. Since the parser uses a cutoff of a word count of 100 or lower to determine whether word given tag probabilities are smoothed, we assigned all new vocabulary a count of 101, and partitioned this count according to the induced tag and sub-tag probabilities. In fact, our attempts to use KNN to induce probabilities over the sub-categories below the level of POS tags were fruitless, producing worse results than the original model in all experiments. Thus, we resorted to using the KNN approach to induce POS level probabilities and then basing the lower level probabilities on a 50-50 interpolation of a general profile for each POS tag and the probabilities assigned by the OOV process.

3.2 C&C

Whereas the Berkeley Parser automatically induces a set of fine-grained categories during training in an attempt to maximize parsing performance, the categories of CCG (Steedman, 2000) have been linguistically designed to represent the dependencies that words will support. In particular, they have a close correspondence to the functional types of lambda calculus representations. So, for example, an intransitive verb has the CCG category $S\backslash NP$, which can be interpreted as identifying this as a syntactic structure that takes a noun phrase to its left (represented by $\backslash NP$) to produce a sentence (represented by S). In other words, it is a function from entities of type NP to type S. In comparison, a transitive verb has the type $(S\backslash NP)/NP$, which describes structure that takes a noun phrase to its right ($/NP$) to produce a structure equivalent to an intranstive verb ($S\backslash NP$), which is itself a category looking for an NP to its left to produce a sentence. Thus, the transitive verb category is a function from two NPs - one to the right and one to the left - to an entity of type S.

[5]http://nlp.stanford.edu/software/lex-parser.shtml

The C&C parser[6] (Curran et al., 2007) is a discriminative parser, which has been trained on CCGbank (Hockenmaier and Steedman, 2007), a translation of the Penn Treebank into the CCG formalism. Roughly, the parser can be split into three modules: a POS-tagger, a super-tagger and the parser itself. The POS-tagger assigns fixed POS tags to the text to be parsed, based on a window of five words centred on the word to be tagged. The super-tagger takes these POS tags and words as input and, using the same five token window, passes CCG tags to the parser. The parser in turn tries to build a derivation from the CCG tags it has been given, but can request a re-analysis from the super-tagger if this fails.

Each module uses a log-linear model to predict which structures, ω, are most likely given the input, S:

$$p(\omega|S) = \frac{e^{\sum_i \lambda_i f_i(\omega)}}{Z_S} \quad (4)$$

where the f_i are a set of features, the λ_i are feature weights and Z_S is a normalising constant.

Here we only consider modifying the POS-tagger and super-taggers, and then only to introduce weights connecting a new lexical item with its corresponding tag. Both taggers make use of many additional features, for example features relating to the dependency of a tag on the two words to either side. However, these additional feature weights do not seem to be effectively estimated by the approach we consider here. Instead, we focus on estimating the feature weights that correspond to the likelihood of a given word taking a particular tag.

3.3 EasyCCG

EasyCCG[7] (Lewis and Steedman, 2014) is another CCG-based parser that also relies on a log-linear model, as described by Equation 4, but only within what is essentially its super-tagger. POS-tagging is avoided as it represents a bottle-neck within the C&C parser, with wrongly assigned POS tags being difficult to recover from. Similarly, the probabilistic model of parse trees is discarded, and instead an A* parser (Klein and Manning, 2003a) is used to search for the valid CCG derivation that maximises the probabilities of the categories assigned to words in the input. The effectiveness of

[6]http://svn.ask.it.usyd.edu.au/trac/candc

[7]http://homepages.inf.ed.ac.uk/s1049478/easyccg.html

this approach depends both on the constraints imposed on derivations by the CCG formalism and also on the performance of the super-tagger, with the latter aspect being reliant on the features chosen for this model.

Whereas the features used by the C&C parser are structures that are explicitly present in the training data, such as a particular sequence of tags or a CCG rule that involves particular head and dependent words, EasyCCG uses low-dimensional word vectors as features, alongside more traditional features such as capitalisation and 2-character suffixes. The CCG category of an input token is then predicted by a log-linear classifier using the features in a 7-word window surrounding it. The word vectors are initialised using the 50-dimensional embeddings induced by Turian (2010) on 37 million words of newswire text, and are further optimised during training on CCGbank. The use of these word vectors allows EasyCCG to generalise well to out-of-domain data, both because embeddings are available for a wider vocabulary than is found in CCGbank and also because the low dimensionality of the vectors counters some of the problems of sparsity.

4 Evaluation

We measure the performance of our parsers in terms of the ability to recover dependencies from biomedical text. Dependency recovery is not only a useful component in processing both clinical text (Lewis et al., 2011; Sohn et al., 2012) and biomedical literature (Seoud and Mabrouk, 2013; Cohen and Elhadad, 2012; Miyao et al., 2008; Poon and Vanderwende, 2010; Qian and Zhou, 2012), it also provides an evaluation metric that is independent of the particular syntactic formalism employed in the parser.

BioInfer (Pyysalo et al., 2007b) is a corpus of about 35,000 words from PUBMED abstracts, annotated with grammatical relations using a slight modification of the Stanford dependencies scheme (de Marneffe et al., 2006). Our models were tuned on a development set of 600 sentences and then evaluated on the remaining 500 sentence test set, using the same split as Pyysalo et al. (2007a) and Rimmel and Clark (2009). The vocabulary in these sentences diverges considerably from that found in the WSJ, with about 27% of the tokens being unseen. Of the $\approx 3,000$ unseen word types found in BioInfer, 92% occur in the unlabelled Medline corpus that we use to induce distributional representations, and over 80% are assigned parameters by the KNN method. In contrast, only about 700 of those unseen words are present in the SENNA vocabulary, all of which are assigned parameters.

Parser	Type	D	k	F-Score
Berkeley	original	-	-	70.67
	2gram	**200**	**10**	**71.37**
	3gram	50	10	70.55
	4gram	2000	5	69.76
	BOW	50	5	70.12
	SG-bio	300	5	68.44
	SENNA	50	10	70.49
	SG-news	300	16	70.41
C&C	original	-	-	76.39
	2gram	200	4	77.52
	3gram	**500**	**3**	**77.82**
	4gram	2000	3	77.61
	BOW	50	5	75.95
	SG-bio	300	3	76.26
	SENNA	50	10	77.02
	SG-news	300	1	76.64
EasyCCG	original	-	-	78.23
	2gram	**100**	**7**	**79.16**
	3gram	1000	7	78.78
	4gram	10000	3	79.02
	BOW	10	20	78.11
	SG-bio	300	18	76.80
	SENNA	50	20	78.65
	SG-news	300	10	78.01

Table 1: F-scores for recovery of dependencies on the BioInfer development set for the best performing D and k for each type of KNN model.

5 Results

Table 1 compares the performance of the Berkeley, C&C and EasyCCG parsers on the BioInfer development set, after KNN adaptation using various forms of distributional similarity. The results for each parser are grouped together with the first line in each of these groups giving the baseline F-score achieved on the BioInfer development set before expanding the vocabulary. Each subsequent line then corresponds to the best model found for each type of representation, with columns containing D, the number of dimensions in the distributional vectors, k, the number of nearest neighbours, and lastly the F-Score.

The types of distributional representation used in the KNN algorithm are subdivided into those constructed on our Medline titles and abstracts and those trained by their authors on other data sources before being made publicly available. The former group consist of the ngram contexts (2gram, 3gram and 4gram), the bag-of-words contexts (BOW) and the retrained Skip-gram model (SG-bio). The downloaded Skip-gram (SG-news) and SENNA (SENNA) vectors make up the latter group.

Looking first at the differences between these approaches to constructing distributional representations, it is reasonably clear that within each parser the worst performing models tend to be those based on bag-of-words contexts (BOW, SG-news and SG-bio). Of the neural embedding models, SENNA gets the best performance, which we attribute to its preservation of sequential order in handling context. Surprisingly, the Skip-gram model retrained on biomedical data (SG-bio) fared worse than the original (SG-news), due probably in large part to the fact that the original training data was almost 100 times larger than our 1.2B word corpus. The ngram contexts achieved the best F-Scores fairly consistently for all parsers, vindicating our appeal to the psycholinguistic research of Cartwright and Brent (1997), Mintz (2003) and Redington et al. (1998).

Turning now to each parser individually, the baseline performance of the Berkeley Parser proved difficult to exceed, with only the 2gram distributional contexts giving any improvement. The best model used the 200 most frequent bigrams as contexts and averaged over 10 nearest neighbours to achieve an uplift of only 0.7% in F-Score. All other types of model resulted in the Berkeley Parser's performance degrading. For the C&C parser, in contrast, most types of representation, except SG-bio and BOW, achieved an uplift. The best model used the 500 most frequent 3gram contexts, and 3 nearest neighbours to infer parameters for unseen words, improving the F-Score by 1.43%. In comparison, the EasyCCG models achieve higher F-Scores but show smaller uplifts. Here, the best model is based on 2grams, using only 100 such contexts, but requiring 7 nearest neighbours to raise the F-Score by 0.93%.

The results of applying these best performing models to the BioInfer test set are given in Table 2. We evaluate performance on both the set of all dependencies and also the subset of dependencies involving unseen words only. All parsers show an uplift on both measures, with C&C achieving the greatest gains: 2.13% over the whole test set and 6.44% on unseen words. The other parsers obtain smaller uplifts of around 3% on the unseen words but these OOV improvements are nonetheless significant at $p < 0.01$ on a bootstrap test (Efron and Tibshirani, 1993) for all parsers. The improvements over the whole test set are diluted by comparison, although still positive.

Parser	Model	F-score	
		All	Unseen
Berkeley	original	69.85	52.78
	enhanced	70.17	55.98
C&C	original	75.56	63.84
	enhanced	77.69	70.28
EasyCCG	original	77.19	71.44
	enhanced	78.31	74.15

Table 2: F-scores for recovery of dependencies for the original models and the best performing KNN enhanced models on the BioInfer test set.

6 Discussion

We have demonstrated a KNN algorithm to estimate parameters for new lexical items that produces improvements in F-score of up to 6% in the recovery of dependencies in biomedical text. These improvements were obtained without having to retrain the parsers, based simply on distributional representations constructed on unlabelled corpora. In fact, since the context vectors comprehensively outperformed the neural embeddings, our approach achieved these gains without having to induce a clustering or other model over the unlabelled corpora and required only counts for ngrams containing the seen and unseen words. In principle, this method could be applied on the fly, as and when the parser encounters new vocabulary. The success of this ngram based approach is also consistent with psycholinguistic research into syntactic acquisition (Cartwright and Brent, 1997; Mintz, 2003; Redington et al., 1998)

We were able to assign parameters to over 80% of the unseen word types. This introduction of parameters for new word types into the lexicon was the only modification made to the parsers, with the remainder of the models being left unchanged. When combined with methods that could adapt the

existing model parameters to the statistics of the new domain, such as self-training (e.g., Deoskar et al., 2014), we expect further improvements to be achievable.

Nonetheless, there were substantial variations in the strength of the improvement attained, with the weak performance of the Berkeley Parser being a notable disappointment. Several differences could be invoked to explain this shortfall. Firstly, the Berkeley Parser has a strong OOV process, and it may just be difficult to beat the estimates it produces, without seeing gold standard data. Secondly, it is a generative rather than a discriminative model, and this complicates the process of modifying the lexicon with questions of how much probability mass to give to unseen words and how to renormalise the lexicon afterwards. Thirdly, rather than representing a single coherent type of linguistic information, the categories induced by the splitting and merging process are just simply the results of whatever splits happened to give the most improvement during training. An example of a subcategory within DT might differentiate definiteness from indefiniteness, while a subcategory in NNP might separate personal names from place names. The inhomogeneity in the type of information encoded in these subcategories probably contributed to our being unable to find distributional information which could be used to induce useful probabilities for them. Consequently, our KNN parameter induction worked only at the level of POS tags for this parser and was therefore less predictive. Andreas and Klein (2014) also struggled to obtain performance improvements for the Berkeley Parser using a distributional matching method. Their problems were also compounded by using SENNA vectors, which we found to give weaker benefits than the ngram context approach.

Our method has certain aspects in common with other approaches to domain adaptation. For example, Koo et al. (2008) train a dependency parser on features deriving from distributional clusters, with two words having similar cluster features if they have similar bigram distributions. Thus, these clusters engender a form of distributional similarity comparable to that used in our KNN algorithm.

KNN algorithms are also commonly used in Graph-Based Semi-Supervised Learning approaches (Das and Petrov, 2011; Altun et al., 2006; Subramanya et al., 2010), with the k-nearest-neighbour sets determining the edges that structure the graph. POS tags are then propagated through the graph from labelled to unlabelled data. Although similarity in these cases is commonly being assessed between token sequences, as opposed to word types, the features used are similar to the ngram templates used here and the bigram distributions used by Koo et al. (2008).

A major difference in our approach is that it does not require retraining the parser or constructing a full model on the unlabelled data. We simply copy parameters from words in the existing lexicon to unseen words, based on a distributional measure of similarity. Moreover, we don't need to see the entire unlabelled corpus. Instead, we can estimate parameters for an unseen word based simply on a set of ngrams centered on it, along with the corresponding ngrams for the existing lexicon.

A reasonable direction for future work would be to develop the way we select the contexts on which our distributional representations are based. In particular, it would make sense to exploit the approach of Brent (1991) and Manning (1993) in which these contexts have an *a priori* linguistic association with particular syntactic frames, as opposed to a merely empirical association deriving from a k-nearest-neighbour model.

Acknowledgements

We would like to thank our colleagues and reviewers for criticism, advice and discussion. This work was supported by ERC Advanced Fellowship 249520 GRAMPLUS and EU Cognitive Systems project FP7-ICT-270273 Xperience.

References

Yasemin Altun, David McAllester, and Mikhail Belkin. 2006. Maximum margin semi-supervised learning for structured variables. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 33–40. MIT Press, Cambridge, MA.

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 822–827. The Association for Computational Linguistics.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for

dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 809–815. The Association for Computational Linguistics.

Michael R. Brent. 1991. Automatic acquisition of subcategorization frames from untagged text. In Douglas E. Appelt, editor, *29th Annual Meeting of the Association for Computational Linguistics, 18-21 June 1991, University of California, Berkeley, California, USA, Proceedings*, pages 209–214. Morgan Kaufmann.

Timothy A. Cartwright and Michael R. Brent. 1997. Syntactic categorization in early language acquisition: Formalizing the role of distributional analysis. *Cognition*, 63:121–170.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 124–131. Association for Computational Linguistics.

Raphael Cohen and Michael Elhadad. 2012. Syntactic dependency parsers for biomedical-NLP. In *AMIA 2012, American Medical Informatics Association Annual Symposium, Chicago, Illinois, USA, November 3-7, 2012*, pages 121–128.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637, December.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic, June. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 600–609. The Association for Computer Linguistics.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, pages 449–454, Genoa, Italy, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L06-1260.

Tejaswini Deoskar, Christos Christodoulopoulos, Alexandra Birch, and Mark Steedman. 2014. Generalizing a strongly lexicalized parser using unlabeled data. In *In Proceedings of Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden.

B. Efron and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Comput. Linguist.*, 33(3):355–396, September.

Marc F. Joanisse and Mark S. Seidenberg. 1999. Impairments in verb morphology after brain injury: A connectionist model. *Proceedings of the National Academy of Sciences*, 96:7592.

Dan Klein and Christopher D. Manning. 2003a. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 40–47. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003b. Accurate unlexicalized parsing. In Erhard W. Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan*, pages 423–430. The Association for Computer Linguistics.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 595–603. The Association for Computer Linguistics.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar, October. Association for Computational Linguistics.

Neal Lewis, Daniel Gruhl, and Hui Yang. 2011. Dependency parsing for extracting family history. In *2011 IEEE International Conference on Healthcare Informatics, Imaging and Systems Biology, HISB 2011, San Jose, CA, USA, July 26-29, 2011*, pages 237–242. IEEE.

Christopher D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora.

In Lenhart K. Schubert, editor, *31st Annual Meeting of the Association for Computational Linguistics, 22-26 June 1993, Ohio State University, Columbus, Ohio, USA, Proceedings*, pages 235–242. The Association for Computer Linguistics.

Ryan T. McDonald and Fernando C. N. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Toben H. Mintz. 2003. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition.*, 90(1):91–117.

Yusuke Miyao, Kenji Sagae, Rune Sætre, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Evaluating contributions of natural language parsers to protein-protein interaction extraction. *Bioinformatics*, 25(3):394–400, December.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 404–411. The Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computational Linguistics.

Steven Pinker. 1987. The bootstrapping problem in language acquisition. In B. MacWhinney, editor, *Mechanisms of language aquisition*, pages 399–441. Lawrence Erlbaum Assoc, Hillsdale, NJ.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 813–821, Los Angeles, CA, USA. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Katri Haverinen, Juho Heimonen, Tapio Salakoski, and Veronika Laippala. 2007a. On the unification of syntactic annotations under the Stanford dependency scheme: a case study on BioInfer and GENIA. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 25–32. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. 2007b. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50–73.

Longhua Qian and Guodong Zhou. 2012. Tree kernel-based protein-protein interaction extraction from biomedical literature. *Journal of Biomedical Informatics*, 45(3):535–543.

Martin Redington, Nick Chater, and Steven Finch. 1998. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, 22(4):425–469.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, 42(5):852–865.

Rania Abul A. Seoud and Mai S. Mabrouk. 2013. Tmthcc: A tool for text mining the biomedical literature for hepatocellular carcinoma (hcc) biomarkers identification. *Computer methods and programs in biomedicine*, 112(3):640–648, August.

Steven L. Small, John Hart, Tran Nguyen, and Barry Gordon. 1995. Distributed representations of semantic knowledge in the brain. *Brain*, 118.

Sunghwan Sohn, Stephen Wu, and Christopher G Chute. 2012. Dependency parser-based negation detection in clinical narrative. *AMIA Summits on Translational Science Proceedings*, 2012:1–8.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 167–176. Association for Computational Linguistics.

Yuta Tsuboi. 2014. Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 938–950, Doha, Qatar, October. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the*

48th Annual Meeting of the Association for Computational Linguistics, pages 384–394. Association for Computational Linguistics.