A Neural Network Model for Low-Resource Universal Dependency Parsing

Long Duong,[12] **Trevor Cohn,**[1] **Steven Bird,**[1] and **Paul Cook**[3]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]Faculty of Computer Science, University of New Brunswick
lduong@student.unimelb.edu.au {t.cohn,sbird}@unimelb.edu.au paul.cook@unb.ca

Abstract

Accurate dependency parsing requires large treebanks, which are only available for a few languages. We propose a method that takes advantage of shared structure across languages to build a mature parser using less training data. We propose a model for learning a shared "universal" parser that operates over an inter-lingual continuous representation of language, along with language-specific mapping components. Compared with supervised learning, our methods give a consistent 8-10% improvement across several treebanks in low-resource simulations.

1 Introduction

Dependency parsing is an important task for Natural Language Processing (NLP) with application to text classification (Özgür and Güngör, 2010), relation extraction (Bunescu and Mooney, 2005), question answering (Cui et al., 2005), statistical machine translation (Xu et al., 2009), and sentiment analysis (Socher et al., 2013). A mature parser normally requires a large treebank for training, yet such resources are rarely available and are costly to build. Ideally, we would be able to construct a high quality parser with less training data, thereby enabling accurate parsing for low-resource languages.

In this paper we formalize the dependency parsing task for a low-resource language as a domain adaptation task, in which a *target* resource-poor language treebank is treated as *in-domain*, while a much larger treebank in a high-resource language forms the *out-of-domain* data. In this way, we can apply well-understood domain adaptation techniques to the dependency parsing task. However, a crucial requirement for domain adaptation is that the in-domain and out-of-domain data have compatible representations. In applying our approach to data from several languages, we must learn such a cross-lingual representation. Here we frame this representation learning as part of a neural network training. The underlying hypothesis for the joint learning is that there are some shared-structures across languages that we can exploit. This hypothesis is motivated by the excellent results of the cross-lingual application of unlexicalised parsing (McDonald et al., 2011), whereby a delexicalized parser constructed on one language is applied directly to another language.

Our approach works by jointly training a neural network dependency parser to model the syntax in both a source and target language. Many of the parameters of the source and target language parsers are shared, except for a small handful of language-specific parameters. In this way, the information can flow back and forth between languages, allowing for the learning of a compatible cross-lingual syntactic representation, while also allowing the parsers to mutually correct one another's errors. We include some language-specific components, in order to better model the lexicon of each language and allow learning of the syntactic idiosyncrasies of each language. Our experiments show that this outperforms a purely supervised setting, on both small and large data conditions, with a gain as high as 10% for small training sets. Our proposed joint training method also out-performs the conventional cascade approach where the parameters between source and target languages are related together through a regularization term (Duong et al., 2015).

Our model is flexible, allowing easy incorporation of peripheral information. For example, assuming the presence of a small bilingual dictionary is befitting of a low-resource setting, as this is prototypically one of the first artifacts generated by field linguists. We incorporate a bilingual dictionary as a set of soft constraints on the

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 339–348,
Lisbon, Portugal, 17-21 September 2015.

model, such that it learns similar representations for each word and its translation(s). For example, the representation of *house* in English should be close to *haus* in German. We empirically show that adding a bilingual dictionary improves parser performance, particularly when target data is limited.

The final contribution of the paper concerns the learned word embeddings. We demonstrate that these encode meaningful syntactic phenomena, both in terms of the observable clusters and through a verb classification task. The code for this paper is published as an open source project.[1]

2 Related Work

This work is motivated by the idea of delexicalized parsing, in which a parser is built without any lexical features and trained on a treebank for a resource-rich source language (Zeman et al., 2008). It is then applied directly to parse sentences in the target resource-poor languages. Delexicalized parsing relies on the fact that identical part-of-speech (POS) inventories are highly informative of dependency relations, and that there exists shared dependency structures across languages.

Building a dependency parser for a resource-poor language usually starts with the delexicalized parser and then uses other resources to refine the model. McDonald et al. (2011) and Ma and Xia (2014) exploited parallel data as the bridge to transfer constraints from the source resource-rich language to the target resource-poor languages. Täckström et al. (2012) also used parallel data to induce cross-lingual word clusters which added as features for their delexicalized parser. Durrett et al. (2012) constructed the set of language-independent features and used a bilingual dictionary as the bridge to transfer these features from source to target language. Täckström et al. (2013) additionally used high-level linguistic features extracted from the World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013).

For low-resource languages, no large parallel corpus is available. Some linguists are dependency-annotating small amounts of field data, e.g. for Karuk, a nearly-extinct language of Northwest California (Garrett et al., 2013). Accordingly, we adopt a different resource requirement: a small treebank in the target low-resource language.

Domain adaptation or joint-training is a different branch of research, and falls outside the scope of this paper. Nevertheless, we would like to contrast our work with Senna (Collobert et al., 2011), a neural network framework to perform a variety of NLP tasks such as part-of-speech (POS) tagging, named entity recognition (NER), chunking, and so forth. Both approaches exploit common linguistic properties of the data through joint learning. However, Collobert et al's goal is to find a single input representation that can work well for many tasks. Our goal is different: we allow the joint-training inputs to be different but constrain the parameter weights in the upper layer to be identical. Consequently, our method applies to the task where inputs are different, possibly from different languages or domains. Their method applies for different tasks in the same language/domain where the inputs are fairly similar.

2.1 Supervised Neural Network Parser

This section describes the monolingual neural network dependency parser structure of Chen and Manning (2014). This parser achieves excellent performance, and has a highly flexible formulation allowing auxilliary inputs. The model is based on a transition-based dependency parser (Nivre, 2006) formulated as a neural-network classifier to decide which transition to apply to each parsing state configuration.[2] That is, for each configuration, the selected list of words, POS tags and labels from the Stack, Queue and Arcs are extracted. Each word, POS and label is mapped into a low-dimension vector representation using an embedding matrix, which is then fed into a two-layer neural network classifier to predict the next parsing action. The set of parameters for the model is $E = \{E^{word}, E^{\text{pos}}, E^{\text{arc}}\}$ for the embedding layer, W_1 for the fully connected cubic hidden layer and W_2 for the softmax output layer. The model prediction function is

$$P(Y|X = \vec{x}, W_1, W_2, E) = \text{softmax}\Big(W_2 \times \text{cube}(W_1 \times \Phi\left[\vec{x}, E\right])\Big) \quad (1)$$

[1] http://github.com/longdt219/universal_dependency_parser

[2] Our approach is focused on a technique for transfer learning which can be more widely applied to other types of dependency parser (and models, generally) regardless of whether they are transition-based or graph-based.

where cube is a non-linear activation function, Φ is the embedding function that returns a vector representation of parsing state x using an embedding matrix E. We refer the reader to Chen and Manning (2014) for a more detailed description.

3 A Joint Interlingual Model

We assume a small treebank in a target resource-poor language, as well as a larger treebank in the source language. Our objective is to learn a model of both languages, subject to the constraint that both models are similar overall, while allowing for some limited language variability. Instead of just training two different parsers on source and then on target, we train them jointly, in order to learn an interlingual parser. This allows the method to take maximum advantage of the limited treebank data available, resulting in highly accurate predicted parses.

Training a monolingual parser as described in section 2.1 requires optimizing the simple cross-entropy learning objective, $\mathcal{L} = -\sum_{i=1}^{|D|} \log P(Y = \vec{y}^{(i)} | X = \vec{x}^{(i)})$, where $P(Y|X)$ is given by equation 1 and $D = \{\vec{x}^{(i)}, \vec{y}^{(i)}\}_{i=1}^{n}$ is the training data. Joint training of a parser over the source and target languages can be achieved by simply adding two such cross-entropy objectives, i.e.,

$$\mathcal{L}_{\text{joint}} = -\sum_{i=1}^{|D_s|} \log P(Y_s = \vec{y}_s^{(i)} | X_s = \vec{x}_s^{(i)}) - \sum_{i=1}^{|D_t|} \log P(Y_t = \vec{y}_t^{(i)} | X_t = \vec{x}_t^{(i)}) \,, \quad (2)$$

where the training data, $D = D_s \cup D_t$, comprises data in both the source and target language. However training the model according to equation 2 will result in two independent parsers. To enforce similarity between the two parsers, we adopt parameter sharing: the neural network parameters, W_1 and W_2, are identical in both parsers. Thereby

$$P(Y_\alpha | X_\alpha = \vec{x}) = P(Y | X = \vec{x}, W_1, W_2, E_\alpha) \,,$$

where the subscript $\alpha \in \{s, t\}$ denotes the source or target language. We allow the embedding matrix E_α to differ in order to accommodate language-specific features, in terms of the representations of lexical types, E_s^{word}, part-of-speech, E_s^{pos} and dependency arc labels E_s^{arc}. This reflects the fact that different languages have different lexicon, parts-of-speech often exhibit different roles, and dependency edges serve different functions, e.g. in Korean a *static verb* can serve as an *adjective* (Kim, 2001). During training, the language-specific errors are back propagated through different branches according to the language, guiding learning towards an interlingual representation that informs parsing decisions in both languages. The set of parameters for the model is W_1, W_2, E_s, E_t where E_s, E_t are the embedding matrices for the source and target languages.

Generally speaking, we can understand the model as building the universal dependency parser that parses the universal language. Specifically, the model is the combination of two parts: the universal part (W_1, W_2) that is shared between the languages, and the conversion part (E_s, E_t) that maps a language-specific representation into the universal language. Naturally, we could stack several non-linear layers in the conversion components such that the model can better transform the input into the universal representation; we leave this exploration for future work. Currently, our cross-lingual word embeddings are meaningful for a pair of source and target languages. However, our model can easily be used for joint training over $k > 2$ languages. We also leave this avenue of enquiry for future work

One concern from equation 2 is that when the source language treebank D_s is much bigger than the target language treebank D_t, it is likely to dominate, and consequently, learning will mainly focus on optimizing the source language parser. We adjust for this disparity by balancing the two datasets, D_s and D_t, during training. When selecting mini-batches for online gradient updates, we select an equal number of classification instances from the source and target languages. Thus, for each step $|D_s| = |D_t|$, effectively reweighting the cross-entropy components in (2) to ensure parity between the languages.

The other concern is over-fitting, especially when we only have a small treebank in the target language. As suggested by Chen and Manning (2014), we apply drop-out, a form of regularization for both source and target language. That is, we randomly drop some of the activation units from both hidden layer and input layer. Following Srivastava et al. (2014), we randomly dropout 20% of the input layer and 50% of the hid-

den layer. Empirically, we observe a substantial improvement applying dropout to the model over MLE or l_2 regularization.

3.1 Incorporating a Dictionary

Our model is flexible, enabling us to freely add additional components. In this section, we assume the presence of a bilingual dictionary between the source and target language. We seek to incorporate this dictionary as a part of model learning, to encode the intuition that if two lexical items are translations of one another, the parser should treat them similarly.[3] Recall that the mapping layer is the combination of word, pos and arc embeddings, i.e., $E_\alpha = \{E_\alpha^{\text{word}}, E_\alpha^{\text{pos}}, E_\alpha^{\text{arc}}\}$. We can easily add bilingual dictionary constraints to the model in the form of regularization to minimize the l_2 distance between word representations, i.e., $\sum_{\langle i,j\rangle \in \mathcal{D}} \|E_s^{\text{word}(i)} - E_t^{\text{word}(j)}\|_F^2$, where $\mathcal{D}$ comprises translation pairs, word(i) and word(j).

When the languages share the same POS tagset and arc set,[4] we can also add further constraints such as their language-specific embeddings be close together. This results a regularised training objective,

$$\mathcal{L}_{\text{dict}} = \mathcal{L}_{\text{joint}} - \lambda \Bigg(\sum_{\langle i,j\rangle \in \mathcal{D}} \|E_s^{\text{word}(i)} - E_t^{\text{word}(j)}\|_F^2 + \|E_s^{\text{pos}} - E_t^{\text{pos}}\|_F^2 + \|E_s^{\text{arc}} - E_t^{\text{arc}}\|_F^2 \Bigg), \quad (3)$$

where $\lambda \in [0, \infty]$ controls to what degree we bind these words or pos tags or arc labels together, with high λ tying the parameters and small λ allowing independent learning. We expect the best value of λ to fall somewhere between these extremes. Finally, we use a mini-batch size of 1000 instance pairs and adaptive learning rate trainer, `adagrad` (Duchi et al., 2011) to build our two separate models corresponding to equations 2 and 3.

4 Experiments

In this section, we compare our joint training approach with baseline methods of supervised learning in the target language, and cascaded learning of source and target parsers.

[3] However, this is not always the case. For example, modal or auxiliary verbs in English often have no translations in different languages or map to words with different syntactic functions.

[4] As was the case for our experiments.

4.1 Dataset

We experiment with the Universal Dependency Treebank (UDT) V1.0 (Nivre et al., 2015), simulating low resource settings.[5] This treebank has many desirable properties for our model: the dependency types (arc labels set) and coarse POS tagset are the same across languages. This removes the need for mapping the source and target language tagsets to a common tagset. Moreover, the dependency types are also common across languages allowing evaluation of the labelled attachment score (LAS). The treebank covers 10 languages,[6] with some languages very highly resourced—Czech, French and Spanish have 400k tokens—and only modest amounts of data for other languages—Hungarian and Irish have only around $25k$ tokens. Cross-lingual models assume English as the source language, for which we have a large treebank, and only a small treebank of 3k tokens exists in each target language, simulated by subsampling the corpus.

4.2 Baseline Cascade Model

We compare our approach to a baseline interlingual model based on the same parsing algorithm as presented in section 2.1, but with cascaded training (Duong et al., 2015). This works by first learning the source language parser, and then training the target language parser using a regularization term to minimise the distance between the parameters of the target parser and the source parser (which is fixed). In this way, some structural information from the source parser can be used in the target parser, however it is likely that the representation will be overly biased towards the source language and consequently may not prove as useful for modelling the target.

4.3 Monolingual Word Embeddings

While the E^{pos} and E^{arc} are randomly initialized, we initialize both the source and target language word embeddings $E_s^{\text{word}}, E_t^{\text{word}}$ of our neural network models with pre-trained embeddings. This is an advantage since we can incorporate the monolingual data which is often available, even for

[5] Evaluating on truly resource-poor languages would be preferable to simulation. However for ease of training and evaluation, which requires a small treebank in the target language, we simulate the low-resource setting using a small part of the UDT.

[6] Czech (cs), English (en), Finnish (fi), French (fr), German (de), Hungarian (hu), Irish (ga), Italian (it), Spanish (es), Swedish (sv).

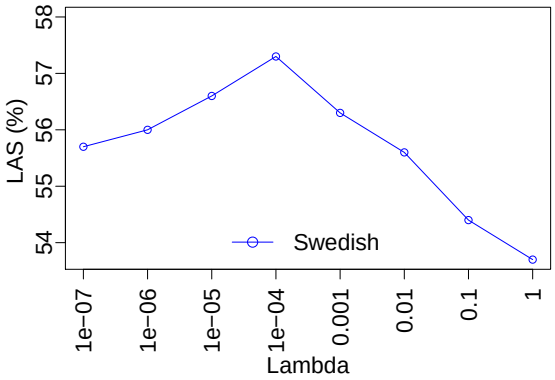

Figure 1: Sensitivity of regularization parameter λ against the LAS measured on the Swedish development set trained on 1000 (tokens).

resource-poor languages. We collect monolingual data for each language from the Machine Translation Workshop (WMT) data,[7] Europarl (Koehn, 2005) and EU Bookshop Corpus (Skadiņš et al., 2014). The size of monolingual data also varies significantly, with as much as 400 million tokens for English and German, and as few as 4 million tokens for Irish. We use the skip-gram model (Mikolov et al., 2013b) to induce 50-dimensional word embeddings.

4.4 Bilingual Dictionary

For the extended model as described in section 3.1, we also need a bilingual dictionary. We extract dictionaries from PanLex (Kamholz et al., 2014) which currently covers around 1300 language varieties and about 12 million expressions. This dataset is growing and aims at covering all languages in the world and up to 350 million expressions. The translations in PanLex come from various sources such as glossaries, dictionaries, automatic inference from other languages, etc. Naturally, the bilingual dictionary size varies greatly among resource-poor and resource-rich languages.

4.5 Regularization Parameter Tuning

Joint training with a dictionary (see equation 3) includes a regularization sensitivity parameter λ. This parameter controls to what extent we should bind the source words and their target translation, common POS tags and arcs together. In this section we measure the sensitivity of our approach with respect to this parameter. In a real world scenario, getting development data to tune this parameter is difficult. Thus, we want a parameter that can work well cross-lingually. To simulate this, we only tune the parameter on one language and apply it directly to different languages. We trained on a small Swedish treebank with 1k tokens, testing several different values of λ. We evaluated on the Swedish development dataset. Figure 1 shows the labelled attachment score (LAS) for different λ. It's clearly visible that $\lambda = 0.0001$ gives the maximum LAS on the development set. Thus, we use this value for all the experiments involving a dictionary hereafter.

4.6 Results

For our initial experiments we assume that we have only a small target treebank with 3000 tokens (around 200 sentences). Ideally the much larger source language (English) treebank should be able to improve parser performance versus simple supervised learning on such a small collection. We apply the joint model (equation 2) and joint model with the dictionary constraints (equation 3) for each target language,

The results are reported in Table 1. The supervised neural network dependency parser performed worst, as expected, and the baseline cascade model consistently outperformed the supervised model on all languages by an average margin of 5.6% (absolute).[8] The joint model also consistently out-performed both baselines giving a further 1.9% average improvement over the cascade. This was despite the fact that the cascaded model had the benefit of tuning for the regularization parameters on a development corpus, while the joint model had no parameter tuning. Note that the improvement varies substantially across languages, and is largest for Czech but is only minor for Swedish. The joint model with the bilingual dictionary outperforms the joint model, however, the improvement is modest (0.7%). Nevertheless, this model gives substantial improvements compared with the cascaded and the supervised model (2.6% and 8.2%).

5 Analysis

5.1 Learning Curve

In section 4.6, we used a 3k token treebank in the target language. What if we have more or less

[7] http://www.statmt.org/wmt14/

[8] We use absolute percentage comparisons herein.

	cs	de	es	fi	fr	ga	hu	it	sv	μ
Supervised	43.1	47.3	60.3	46.4	56.2	59.4	48.4	65.4	52.6	53.2
Baseline Cascaded	49.6	59.2	66.4	49.5	63.2	59.5	50.5	69.9	61.4	58.8
Joint	55.2	61.2	69.1	51.4	65.3	60.6	51.2	71.2	61.4	60.7
Joint + Dict	55.7	61.8	70.5	51.5	67.2	61.1	51.0	71.3	62.5	61.4

Table 1: Labelled attachment score (LAS) for each model type trained on 3000 tokens for each target language (columns). All bar the supervised model also use a large English treebank.

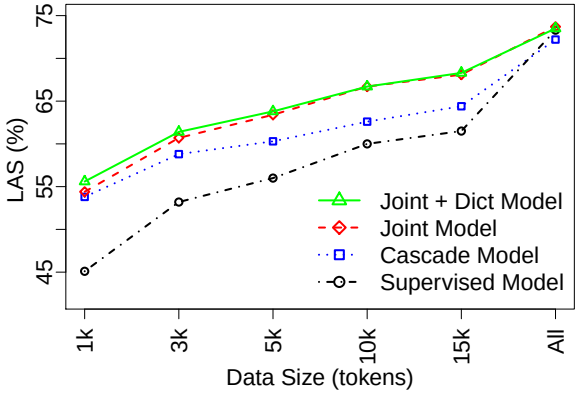

Figure 2: Learning curve for Joint model, Joint + Dict model, Baseline cascaded and Supervised model: the x-axis is the size of data (number of tokens); the y-axis is the average LAS measured on 9 languages (except English).

target language data? Figure 2 shows the learning curve with respect to various models on different data sizes averaged over all target languages. For small datasets of 1k training tokens, the cascaded model, joint model and joint + dict model performed similarly well, out-performing the supervised model by about 10% (absolute). With more training data, we see interesting changes to the relative performance of the different models. While the baseline cascade model still outperforms the supervised model, the improvement is diminishing, and by 15k, the difference is only 2.9%. On the other hand, compared with the supervised model, the joint and joint + dict models perform consistently well at all sizes, maintaining an 8% lead at 15k. This shows the superiority of joint training compared with single language training.

To understand this pattern of performance differences for the cascade versus the joint model, one needs to consider the cascade model formulation. In this approach, the target language parameters are tied (softly) with the source language parameters through regularization. This is a benefit for small datasets, providing a smoothing function to limit overtraining. However, when we have more training data, these constraints limit the capacity of the model to describe the target data. This is compounded by the problem that the source representation may not be appropriate for modelling the target language, and there is no way to correct for this. In contrast the joint model learns a mutually compatible representation automatically during joint training.

The performance results for the joint model with and without the dictionary are similar overall. Only on small datasets (1k, 3k), is the difference notable. From 5k tokens, the bilingual dictionary doesn't confer additional information, presumably as there is sufficient data for learning syntactic word representations. Moreover, translation entries exist between syntactically related word types as well as semantically related pairs, with the latter potentially limiting the beneficial effect of the dictionary.

When training on all the target language data, the supervised model does well, surpassing the cascade model. Surprisingly, the joint models outperform slightly, yielding a 0.4% improvement. This is an interesting observation suggesting that our method has potential for use not only for low resource problems, but also high resource settings.

5.2 Different Tagsets

In the above experiments, we used the universal POS tagset for all the languages in the corpus. However, for some languages,[9] the UDT also provides language specific POS tags. We use this data to test the relative performance of the model using a universal tagset cf. language specific tagsets. In this experiment, we applied the same joint model (see §3) but with a language specific tagset instead of UPOS for these languages. We expect the joint

[9] en, cs, fi, ga, it and sv.

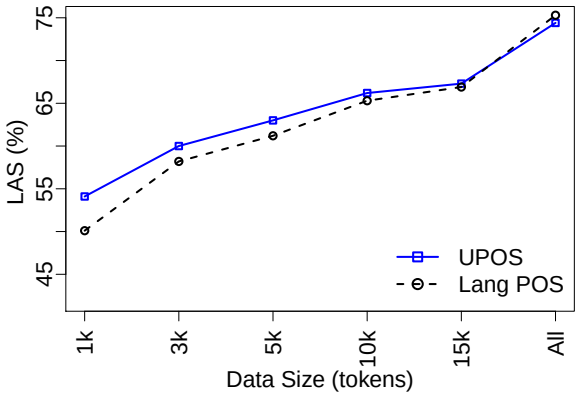

Figure 3: Learning curve for joint model using the UPOS tagset or language specific POS tagset: the x-axis is the size of data (number of tokens); the y-axis is the average LAS measured on 5 languages (except English).

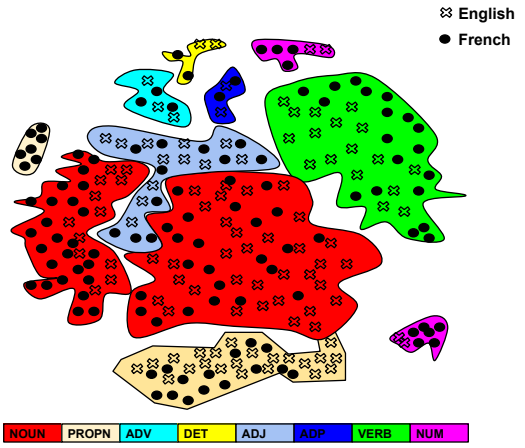

Figure 4: Universal Language visualization according to language and POS. (This should be viewed in colour.)

model to automatically learn to project the different tagsets into a common space, i.e., implicitly learn a tagset mapping between languages. Figure 3 shows the learning curve comparing the joint model with the two types of POS tagsets. For the small dataset, it is clear that the data is insufficient for the model to learn a good tagset mapping, especially for a morphologically rich language like Czech. However, with more data, the model is better able to learn the tagset mapping as part of joint training. Beyond 15k tokens, the joint model using the language specific POS tagset outperforms UPOS. Clearly there is some information lost in the UPOS tagset, although the UPOS mapping simultaneously provides implicit linguistic supervision. This explains why the UPOS might be useful in small data scenarios, but detrimental at scale. Using all the target data ("All") the language specific POS provides a 1% (absolute) gain over UPOS.

5.3 Universal Representation

As described in section 3, we can consider our joint model as the combination of two parts: a universal parser and a language-specific embedding E_s or E_t that converts the source and target language into the universal representation. We now seek to analyse qualitatively this universal representation through visualization. For this purpose we use a joint model of English and French, using all the available French treebank (more than 350k tokens) as well as a bilingual dictionary.[10] Figure 4 shows the t-SNE (Van Der Maaten, 2014) projection of the 50 dimensional word embeddings in both languages. We can see that English and French are mixed nicely together. The colouring denotes the POS tag, showing clearly that the words with similar POS tags are grouped together regardless of languages. This is partially understandable since word embeddings for dependency parsing need to convey the dependency context rather than surrounding words, as in most distributional embedding models. Words having similar dependency relation should be grouped together as they are treated similarly by the parser.

Some of the learned cross-lingual word-embeddings are shown in Table 2, which includes the five nearest neighbours to selected English words according to the monolingual word embedding (section 4.3) and our cross-lingual dependency word embeddings, trained using PanLex. The monolingual sets appear to be strongly characterised by distributional similarity. The cross-lingual embeddings display greater semantic similarity, while being more variable morphosyntactically. In many cases, the top five words of English and French are translations of each other, but with varying inflectional endings in the French forms. For example, "*buy*" vs "*vendez*" or "*invest*" vs "*in-*

[10] We also visualized the cross-lingual word embeddings without the dictionary, however the results were rather odd. Although we saw coherent POS clusters, the two languages were largely disjoint. We speculate that many components of the embeddings are use for only one language, and these outnumber the shared components, and thus more careful projection is needed for meaningful visualisation.

Words	Mono	Cross lingual embedding En	Cross lingual embedding Fr
sell	buy	buy	revendre
	eat	invest	vendez
	produce	integrate	acheter
	compete	guide	achètent
	burn	eat	investir
playing	serving	sailing	jouait
	acting	play	navigue
	paying	moving	jouent
	pursuing	faces	pièce
	running	ran	jouer
hard	difficult	crazy	dur
	harder	strange	dures
	easy	beautiful	hard
	magnificent	friendly	fou
	painful	difficult	folles
initially	originally	originally	réellement
	previously	previously	déjà
	officially	officially	récemment
	basically	actually	dernièrement
	already	already	surroît
university	teachers	school	universitaire
	student	education	université
	teacher	student	école
	student	medicine	scolaire
	training	participant	school
mobile	wireless	computers	mobile
	goods	Web	mobiles
	online	Internet	ordinateurs
	freight	computer	Web
	broadband	web	internet

Table 2: Examples of 5 nearest neighbours with the target English word using the original monolingual word embedding and our cross-lingual dependency based word embedding.

vestir". This is a direct consequence of incorporating the bilingual lexicon. Moreover, the top five closest words of both English and French mostly have the same part of speech. This is consistent with the finding in Figure 4.

Levin (1993) has shown that there is a strong connection between a verb's meaning and its syntactic behaviour. We compare the English side of our cross-lingual dependency based word embeddings with various other pre-trained monolingual English word embeddings and our monolingual embedding (section 4.3) on Verb-143 dataset (Baker et al., 2014). This dataset contains 143 pairs of verbs that are manually given score from 1 to 10 according to the meaning similarity. Table 3 shows the Pearson correlation

	Correlation
Senna (Collobert et al., 2011)	0.36
Skip-gram (Mikolov et al., 2013a)	0.27
RNN (Mikolov et al., 2011)	0.31
Our monolingual embedding	0.39
Our crosslingual embedding	0.44

Table 3: Compare the English side of our cross-lingual embeddings with various other embeddings evaluated on Verb-143 dataset (Baker et al., 2014). We directly use the pre-trained models from corresponding papers.

with human judgment for our embeddings and other pre-trained embeddings. As expected, our cross-lingual embeddings out-perform others embeddings on this dataset. This is partly because the syntactic behaviour is well encoded in our word embeddings through dependency relation.

Our embeddings encode not just cross-lingual correspondences, but also capture dependency relations which we expect might be beneficial for other NLP tasks based on dependency parsing, e.g., cross-lingual semantic role labelling where long-distance relationship can be captured by word embedding.

6 Conclusion

In this paper, we present a training method for building a dependency parser for a resource-poor language using a larger treebank in a high-resource language. Our approach takes advantage of the shared structure among languages to learn a universal parser and language-specific mappings to the lexicon, parts of speech and dependency arcs. Compared with supervised learning, our joint model gives a consistent 8-10% improvement over several different datasets in simulation low-resource scenarios. Interestingly, some small but consistent gains are still realised by joint cross-lingual training even on large complete treebanks. This suggests that our approach has utility not just in low resource settings. Our joint model is flexible, allowing the incorporation of a bilingual dictionary, which results in small improvements particularly for tiny training scenarios.

As the side-effect of training our joint model, we obtain cross-lingual word embeddings specialized for dependency parsing. We expect these embeddings to be beneficial to other syntatic and se-

mantic tasks. In future work, we plan to extend joint training to several languages, and further explore the idea of learning and exploiting cross-lingual embeddings.

Acknowledgments

This work was supported by the University of Melbourne and National ICT Australia (NICTA). Trevor Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 278–289.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA. ACM.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China, July. Association for Computational Linguistics.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Garrett, Clare Sandy, Erik Maier, Line Mikkelsen, and Patrick Davidson. 2013. Developing the Karuk Treebank. Fieldwork Forum, Department of Linguistics, UC Berkeley.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–50, Reykjavik, Iceland. European Language Resources Association (ELRA).

Min-joo Kim. 2001. Does korean have adjectives. In *MIT Working Papers 43. Proceedings of HUMIT 2001*, pages 71–89. MIT Working Papers.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand.

B. Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348. Association for Computational Linguistics.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 62–72.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Honza Cernocky. 2011. Rnnlm – recurrent neural network language modeling toolkit. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, December.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0.

Joakim Nivre. 2006. *Inductive Dependency Parsing (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Levent Özgür and Tunga Güngör. 2010. Text classification with the support of pruned dependency patterns. *Pattern Recogn. Lett.*, 31(12):1598–1607, September.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia, June. Association for Computational Linguistics.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, January.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado, June. Association for Computational Linguistics.

Daniel Zeman, Univerzita Karlova, and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *In IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.