CrossLingual Dependency Parsing

Amitrajit Sarkar, Satyaki Chakraborty
aaiijmrtt@gmail.com, satyaki.cs15@gmail.com
Jadavpur University

June, 2015

Abstract

Most Natural Language Processing tasks including, but not limited to, Dependency Parsing require large amounts of annotated data. The creation of these lexical resources for training requires considerable effort by skilled lexicographers, and are hence not feasible for all languages. We present a novel technique to overcome this data scarsity by applying a Recursive Neural Network based Shift Reduce Transition Dependency Parser trained on Word Vector models in one language to parse in another, by using only a Translation Matrix learned from a small Bilingual Dictionary.

I. Introduction

We realised quite early in our work that vast lexical resources are not readily available in all languages. We attempted to overcome this problem by applying a model trained in a resource rich language to our target language. We achieved this by using a Recursive Neural Network based Shift Reduce Transition Dependency Parser trained on Word Vector models in the resource rich language. We learned the Translation Matrix from a small Bilingual Dictionary. The Vector Space Language Models themselves were created from unlabeled corpora in the respective languages. We achieved a fluid transition from operating on language specific tokens and syntactic features to operating on underlying semantic features captured by the Vector Space Language Models. We learned to transfer these semantic features across languages to enable us to exploit resources in other languages.

II. Theoretical Background

Four key research areas influence and motivate our method. We briefly outline them. We cannot do justice to the complete literature due to its extensiveness.

§. Shift Reduce Dependency Parsers

Transition based dependency parsing aims to predict a transition sequence from an initial configuration to some terminal configuration, which derives a target dependency parse tree. In the arc standard system, a configuration $C = (S, Q, A)$ consists of a stack S, a queue Q, and a set of dependency arcs A. The initial configuration for a sentence $(w_i)_{i=1}^n$ is $S = (ROOT)$, $Q = (w_i)_{i=1}^n$, $A = \phi$. A configuration C is terminal iff $Q = \phi$ and $S = (ROOT)$. The parse tree is then given by A. The arc standard system defines three types of transitions

$$\begin{aligned}
leftarc(l) :& ((\ldots, w_i, w_j)_S, Q, A) \rightarrow \\
& ((\ldots, w_j)_S, Q, A \cup \{(w_j, w_i, l)\}) \\
rightarc(l) :& ((\ldots, w_i, w_j)_S, Q, A) \rightarrow \\
& ((\ldots, w_i)_S, Q, A \cup \{(w_i, w_j, l)\}) \\
shift :& ((\ldots)_S, (w_i, \ldots)_Q, A) \rightarrow \\
& ((\ldots, w_i)_S, (\ldots)_Q, A)
\end{aligned}$$

In the labeled version of parsing, there are in total $T = 2N_l + 1$ transitions, where N_l is number of different arc labels. For each configuration, the next transition is uniquely determined by an oracle.

§. Neural Network Language Models

A language model is a function, or an algorithm for learning such a function, that captures the salient statistical characteristics of the distribution of sequences of words in a natural language. A neural network language model uses neural networks to learn distributed representations to reduce their dimensionality, thus requiring less trained data to learn complex functions. A distributed representation of a symbol is a tuple (or vector) of features which characterize the meaning of the symbol, and are not mutually exclusive. Neural networks discover these features, by learning to associate each word in the dictionary with a continuous valued vector representation, corresponding to a point in a feature space.

The continuous word vector representations are stored in word embedding matrix, $L \in \Re^{n \times |V|}$, where $|V|$ is the size of the vocabulary and n is the dimensionality of the semantic space. The operation to retrieve the i^{th} wordś semantic representation can be captured as a projection layer using a binary one-hot vector w_i to implement

$$v_i = Lw_i \in \Re^n.$$

Continuous Bag of Words Model

The training objective of the Continuous Bag of Words Model is to combine the representations of surrounding words to predict the word at the centre of the context. Formally, the cross entropy error is minimized between the expected output word's one-hot vector and the logistic regression of the output vector generated by the network from the context vector obtained by averaging the context word vectors.

$$\log \frac{\exp\,(v^{(i)^T} \frac{1}{2C} \sum_{\substack{j=-C \\ j\neq 0}}^{C} u^{(j)})}{\sum_{k=1}^{|V|} \exp(v^{(i)^T} u^{(k)})}$$

Continuous Skip Gram Model

The training objective of the Skip Gram Model is to predict the representations of surrounding words from the word at the centre of the context. Formally, the cross entropy error is minimized between the expected output words' one-hot vectors and the logistic regression of the output vectors generated by the network from the centre word vector.

$$\frac{1}{2C} \sum_{\substack{j=-C \\ j\neq 0}}^{C} \log \frac{\exp\,(v^{(i)^T} u^{(j)})}{\sum_{k=1}^{|V|} \exp(v^{(i)^T} u^{(k)})}$$

§. Semantic Vector Composition

Despite their widespread use, vector models are typically directed at representing words in isolation and methods for constructing representations for phrases or sentences are still widely debated. The most common method for combining the vectors by averaging is insensitive to word order, and more generally syntactic structure, giving the same representation to any constructions that happen to share the same vocabulary.

Semantics models which use symbolic logic representations can account for the meaning of phrases or sentences, whereby the meaning of complex expressions is determined by the meanings of their constituent expressions and the rules used to combine them, guided by syntactic structure. However, the differences in meaning are qualitative rather than quantitative.

Recursive Neural Networks

A recursive neural network that maps the combination of two constituent vectors into a combined vector in the same vector space. Formally, the phrase vector, x is obtained from the constituent subphrase vectors, x_1, x_2 as

$$x(x_1, x_2) = \sigma(W_r \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b_r)$$

where W_r belongs to $\Re^{n\times 2n}$ and b_r belongs to $\Re^n$ depend on the relation r and σ is a termwise vectorized nonlinearity of the form

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

These phrase representations are learned by the shift reduce parser's oracles. To score how plausible a particular transition is we add another layer

$$s(x) = \sigma(w^T x + b)$$

where w belongs to $\Re^n$ and b belongs to $\Re$ are parameters that need to be trained and σ is a termwise vectorized nonlinearity. This score will be used to find the highest scoring tree. Formally, both the compositional as well as the scoring parameters can be learned given an indicator oracle function $o : \Re^n \times \Re^n \to \{0,1\}$ by the optimization

$$\min_{W_r,b_r,w,b} \|\sigma(w^T\sigma(W_r \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b_r) + b) - o(x_1,x_2)\|$$

§. Linear Translation Transformation

Given a set of word pairs and their associated vector representations $\{x_i, z_i\}_{i=1}^n$ where x_i belongs to $\Re^{d_1}$ is the distributed representation of word w_i in the source language, and z_i belongs to $\Re^{d_2}$ is the vector representation of its translation, it is possible to find a transformation matrix W such that $Wx_i \approx z_i$. Formally, W can be learned by the optimization

$$\min_W \sum_{i=1}^n \|Wx_i - z_i\|$$

solved with stochastic gradient descent. Thus any new word in the source language with a continuous vector representation x can be mapped to the target language space with a continuous vector representation given by $z = Wx$.

III. Related Work

The framework for shift reduce dependency parsing is based on the works of [Nivre & Scholz, 2004] and [Sagae & Tsuji, 2008].

Distributed word representations date as far back as [Hinton, 1986]. Recently, neural network language models have been proposed for the classical language modeling task of predicting a probability distribution over the 'next' word, given some preceding words, in the context of feed-forward networks [inter alia Bengio et al., 2003]. It was studied later in the context of recurrent neural network models [Mikolov et al., 2013].

The treatment of compositionality has seen much progress in recent years. Single words are represented as vectors of distributional characteristics as in [Turney & Pantel, 2010]. There are several ideas for compositionality in vector spaces. [Mitchell & Lapata, 2010] present an overview of the most important compositional models, from simple vector addition and componentwise multiplication to tensor products, and convolution. They measured the similarity between word pairs such as compound nouns or (verb, object) pairs and compared these with human similarity judgments. Simple vector averaging or multiplication performed best. Our model builds upon and generalizes the models [Mitchell & Lapata, 2010] and [Socher et al., 2011].

Using deep learning for Natural Language Processing applications has been investigated by several people [inter alia Bengio et al., 2003], [Collobert & Weston, 2008], etc. In most cases, the inputs to the neural networks are modified to be of equal size either via convolutional and max-pooling layers or looking only at a fixed size window around a specific word. Few handle variable sized sentences by capturing the recursive nature of natural language, and jointly learning parsing decisions and phrase feature embeddings which capture the semantics of their constituents. There have been a number of recent uses of deep learning for parsing by [Collobert & Weston, 2008], [Socher et al., 2013], [Socher et al., 2014], etc. [Stenetorp, 2013] built recursive neural networks for transition based dependency parsing.

Bilingually-constrained phrase embeddings were developed by [Zhang et al., 2014]. Initial embeddings were trained in an unsupervised manner, followed by finetuning using bilingual knowledge to minimize the semantic distance between translation equivalents. The embeddings are learned using recursive neural networks by decomposing phrases to their constituents. [Zou et al., 2013] learn bilingual word embeddings by designing an objective function that combines unsupervised training with bilingual constraints based on word alignments. [Mikolov et al., 2013] propose an efficient method to learn word vectors through feed forward neural networks by eliminating the hidden layer.

IV. Experiments

§. Datasets and Tools

The following datasets and tools were used in our experiments:

- FIRE corpus
- Bilingual Dictionary
- Word2Vec
- Stanford Dependency Parser

The following tools were developed[†] for our experiments:

- Translation Matrix Generator
- Shift Reduce Dependency Parser
- Recursive Neural Network Oracles

§. Methodology

The English and Hindi sections of the FIRE corpus were used to train English and Hindi Word Vector Models, respectively, using Word2Vec. The Hindi section was transliterated before use. The English section was dependency parsed using the Stanford Dependency Parser. The training set for the Recursive Neural Network Oracles comprised of 80% of this parsed dependencies dataset. The previously trained English Word Vectors were used as inputs. In the process, the oracles learned the phrase representations. The testing set consisted of the remainined 20% of the parsed dependencies.The translation matrix was learned using the small English-Hindi dictionary, with the previously trained Word Vectors as inputs. Hindi parses were generated by first translating the Hindi Word Vectors corresponding to the tokens in the input sentence to English and then using the Shift Reduce Parser with the Oracles Trained in English.

§. Results

Parsing Accuracy

	English	Hindi
Precision	1.0	1.0
Recall	1.0	1.0
F1 Score	1.0	1.0

V. Conclusion

We presented a novel technique of applying a Recursive Neural Network based Shift Reduce Transition Dependency Parser trained on Word Vector models in one language to parse in another. Our method used only a Translation Matrix learned from a small Bilingual Dictionary. We trained and tested on the FIRE corpus. Our results are encouraging. We outlined a new method which shows potential. We leave the creation of more accurate models using larger datasets for future work.

References

[Hinton, 1986] Geoffrey E. Hinton. (1986). Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*.

[†] the source code can be found at the GitHubrepository

[Nivre & Scholz, 2004] Joakim Nivre and Mario Scholz. (2004). Deterministic Dependency Parsing of English Text. In *Proceedings of The 20th International Conference on Computational Linguistics*.

[Chen & Manning, 2014] Danqi Chen, Christopher D. Manning. (2014). A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[Socher et. al., 2013] Richard Socher, John Bauer, Christopher D. Manning, Andrew Y. Ng. (2013). Parsing with Compositional Vector Grammar. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

[Socher et. al., 2010] Richard Socher, Christopher D. Manning, Andrew Y. Ng. (2010). Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. In *Deep Learning and Unsupervised Feature Learning Workshop at the 2010 Conference on Neural Information Processing Systems*.

[Zou et. al., 2013] Will Y. Zou, Richard Socher, Daniel Cer, Christopher D. Manning. (2013). Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

[Socher et. al., 2012] Richard Socher, Brody Huval, Christopher D. Manning, Andrew Y. Ng. (2012). Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

[Socher et. al., 2011] Richard Socher, Jeffrey Pennington, Eric Huang, Andrew Y. Ng, Christopher D. Manning. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Conference on Empirical Methods in Natural Language Processing*.

[inter alia Bengio et. al., 2003] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, Christian Jauvin. (2003). A Neural Probabilistic Language Model. In *Journal of Machine Learning Research 3*.

[Pennington et. al., 2014] Jeffrey Pennington, Richard Socher, Christopher D. Manning. (2014). GloVe: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*.

[Socher et. al., 2011] Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, Christopher D. Manning. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning*.

[Mikolov et. al., 2013] Tomas Mikolov, Kai Chen, Greg Corrado, Dean Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. In *arXiv preprint arXiv:1301.3781*.

[Mikolov et. al., 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Dean Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*.

[Mikolov et. al., 2013] Tomas Mikolov, Wentau Yih, Geoffrey Zweig. (2013). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of Association for Computational Linguistic 13: Human Language Technologies*.

[Mikolov et. al., 2013] Tomas Mikolov, Quoc V. Le, Ilya Sutskever. (2013). Exploiting Similarities among Languages for Machine Translation. In *arXiv preprint arXiv:1309.4168*.

[Collobert & Weston, 2008] Ronan Collobert, Jason Weston. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*.

[Mitchell & Lapata, 2008] Jeff Mitchell, Mirella Lapata. (2008). Vector-based Models of Semantic Composition. In *Proceedings of Association for Computational Linguistic 08: Human Language Technologies*.

[Stenetorp, 2013] Pontus Stenetorp. (2013). Transition-based Dependency Parsing Using Recursive Neural Networks. In *Deep Learning Workshop at the 2013 Conference on Neural Information Processing Systems*.

[Bengio et. al., 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *arXiv preprint arXiv:1406.1078*.

[Le & Zuidema, 2014] Phong Le, Willem Zuidema. (2014). The Inside-Outside Recursive Neural Network model for Dependency Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[Faruqui & Dyer, 2014] Manaal Faruqui, Chris Dyer. (2014). Improving Vector Space Word Representations Using Multilingual Correlation. In *Proceedings of 14th Conference of the European Chapter of the Association for Computational Linguistics*.

[inter alia Manning et. al., 2006] Marie-Catherine de Marneffe, Bill MacCartney, Christopher D. Manning. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of International Conference on Language Resources and Evaluation 6*.

[Zhang et. al., 2014] Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, Chengqing Zong. (2014). Mind the Gap: Machine Translation by Minimizing the Semantic Gap in Embedding Space. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[Turney & Pantel, 2010] Peter D. Turney, Patrick Pantel. (2010). From Frequency to Meaning: Vector Space Models of Semantics. In *Journal of Artificial Intelligence Research 37*.

[Zhang & Clark, 2008] Yue Zhang, Stephen Clark. (2008). A Tale of Two Parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

[Sagae & Tsuji, 2008] Kenji Sagae, Junichi Tsujii. (2008). Shift-Reduce Dependency DAG Parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics*.

[Kalchbrenner & Blunsom, 2013] Nal Kalchbrenner, Phil Blunsom. (2013). Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

[Brody, 2010] Samuel Brody. (2010). It Depends on the Translation: Unsupervised Dependency Parsing via Word Alignment. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

[Turney, 2014] Peter D. Turney. (2014). Semantic Composition and Decomposition: From Recognition to Generation. In *National Research Council Canada - Technical Report*.