IDIAP RESEARCH REPORT

SYNTACTIC PARSING OF MORPHOLOGICALLY RICH LANGUAGES USING DEEP NEURAL NETWORKS

Joël Legrand Ronan Collobert

Idiap-RR-25-2015

JUNE 2015

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Syntactic Parsing of Morphologically Rich Languages Using Deep Neural Networks

Joël Legrand
Idiap Research Institute
Martigny, Switzerland
École Polytechnique Fédérale de Lausanne
Lausanne Switzerland
joel.legrand@idiap.ch

Ronan Collobert *
Facebook AI Research
Menlo Park, CA, USA
Idiap Research Institute
Martigny, Switzerland
ronan@collobert.com

Abstract

This paper presents parsing results for the constituency track of the SPMRL shared task. We use the recurrent neural network model of (Legrand and Collobert, 2015) which leverages a new RNN-based compositional sub-tree representation. Results are provided for different scenarios were models are trained on the full corpus or on a subset of 5k sentences, using either the gold or the predicted POS tags. Our system outperforms the baseline and achieves significant improvements over the state of the art model for three languages, while being very fast due to its greedy nature.

1 Introduction

The SPMRL Shared Task 2013 (Seddah et al., 2013) provides standardized datasets, evaluation metrics and baseline results, in both constituency and dependency parsing, for nine different languages. This paper presents results for the constituency track, using the system introduced in (Legrand and Collobert, 2015). The model used is a greedy parser which leverages a new composition approach to keep a history of what has been predicted so far. The composition performs a syntactic and semantic summary of the contents of a sub-tree in form of a vector representation. The composition is performed along the tree: bottom tree node representations are obtained by composing *continuous word and tag vector representations*, and produces vector representations which are in turn composed together in subsequent nodes of the tree. The composition operation as well as tree node tagging and predictions are achieved with a Recurrent Neural Network (RNN). Both the composition and node prediction are trained *jointly*.

*All research was conducted at the Idiap Research Institute, before Ronan Collobert joined Facebook AI Research.

Both the baseline (Berkeley parser) and the state-of-the-art (Björkelund et al., 2014) models rely on PCFG-based features. The latter uses a product of PCFG with Latent Annotations based models (Petrov, 2010), with a Coarse-to-Fine decoding strategy. The output is then discriminatively reranked (Charniak and Johnson, 2005) to select the best analysis. In contrast, our parser constructs the parse tree in a greedy manner and relies only on word and tag embeddings. Thanks to its greedy nature, our parser is very fast: it is able to parse around 80 (20 for its voting version) sentences per second on average (on a single CPU). Furthermore, as this model relies only on word and tag embeddings, it could be easily enhanced by leveraging unsupervised embeddings.

2 The Model

2.1 Greedy RNN Parsing

$\mathbf{I}_W$	:	Did	you	hear	the	falling	bombs	?
$\mathbf{I}_T$	:	VBD	PRP	VB	DT	VBG	NNS	.
$\mathbf{O}$	:	O	S-NP	O	B-NP	I-NP	E-NP	O
$\mathbf{I}_W$	:	Did	*R1*	hear	*R2*	.		
$\mathbf{I}_T$	:	VBD	NP	VB	NP	.		
$\mathbf{O}$	:	O	O	B-VP	E-VP	.		
$\mathbf{I}_W$	:	Did	*R1*	*R3*	?			
$\mathbf{I}_T$	:	VDB	NP	VP	.			
$\mathbf{O}$	:	B-SQ	I-SQ	I-SQ	E-SQ			

Figure 1: Greedy parsing algorithm (3 iterations), on the sentence "Did you hear the falling bombs ?". $\mathbf{I}_W$, $\mathbf{I}_T$ and $\mathbf{O}$ stand for input words (or composed word representations R_i), input syntactic tags (parsing or part-of-speech) and output tags (parsing), respectively.

The model of (Legrand and Collobert, 2015) is entirely based on neural networks and performs parsing in a greedy recurrent way. Our approach is a bottom-up iterative procedure: the tree is built

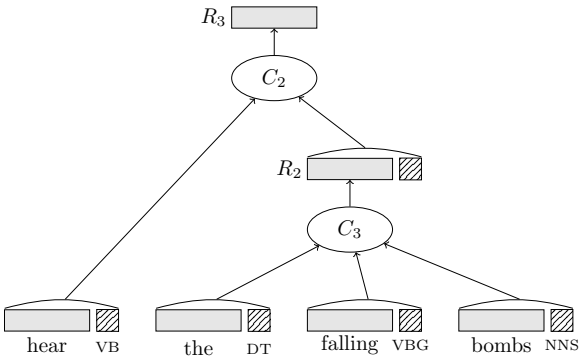

Figure 2: Recurrent composition of the sub-tree *(VP (VB hear) (NP (DT the) (VBG falling) (NNS bombs)))*. The representation R2 is first computed using the 3-inputs module C_3 with *the/DT falling/VBG bombs/NNS* as input. R3 is obtained by using the 2-inputs module C_2 with *hear/VB R2/NP* as input

starting from the terminal nodes (sentence words), as shown in Figure 1. This greedy procedure is explained in detail in (Legrand and Collobert, 2015).

2.2 Word and Tag Embeddings

Each iteration of our parsing can be seen as a simple sequence tagging task. This is done using the model introduced in (Collobert and Weston, 2008) on various NLP tasks. This model relies on word and tag embeddings. Each word (resp. tag) in a finite dictionary $\mathcal{W}$ (resp. $\mathcal{T}$), is assigned a continuous vector representation which is, as all parameters of our architecture, trained by back-propagation. Note that we did not use pre-trained word embeddings as there were not any available for every language.

More formally, each word (resp. tag) is embedded in a D-dimensional (resp. T-dimensional) vector space by applying a lookup-table operation $LT_X(n) = X_n$, where X is a $D \times |\mathcal{W}|$ (resp. $T \times |\mathcal{T}|$) matrix of parameters to be train. The column X_n corresponds to the vector embedding of the n^{th} word (resp. tag) in our dictionary $\mathcal{W}$ (resp. $\mathcal{T}$).

2.3 Word-Tag Composition

At each step of the parsing procedure, the tagger is fed with word and node representations. The node representation is a summary of the corresponding sub-tree. As shown in Figure 2, the vector representation is obtained by a simple recurrent procedure which outputs a representation living in the same space as the word representations (dimension D).

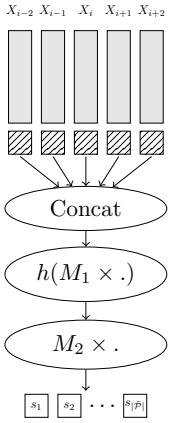

Figure 3: A constituent X_i is tagged by considering a fixed size context window of size K (here $K = 5$). The concatenated output of the compositional history and constituent tags is fed as input to the tagger (a standard two-layer neural network). It outputs a score for each BIOES-prefixed parsing tag.

Compositional networks take as input both the merged node word or node representations (dimension D) and predicted tag representations (dimension T). There is one different network C_k for each possible node with a number of k merged constituent. In practice most tree nodes do not merge more than a few constituents. In our case, denoting $z \in \mathbb{R}^{(D+T)\times k}$ the concatenation of the merged constituent representations (k vectors of tags and constituent representations), the compositional network is simply a matrix-vector operation followed by a non-linearity

$$C_k(z) = h(M^k z)\,,$$

where $M^k \in \mathbb{R}^{D\times(k(D+T))}$ is a matrix of parameters to be trained, and $h()$ is a simple non-linearity such as a pointwise hyperbolic tangent.

As the node and word representations are embedded in the same space, the compositional networks C_k can compress information coming both from leaves and sub-trees. Similarly, the tagger network can be fed indifferently with word or sub-tree representations.

2.4 Sliding Window BIOES Tagger

As illustrated in Figure 3, the tagging module of our architecture (see Figure 3) is a two-layer neural network which applies a sliding window of size K over the input constituent representations (as computed in Section 2.3), as well as the input constituent tag representations. Considering

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	avg
1) gold POS / full training set										
IMS:SZEGED:CIS	**82.20**	**90.04**	**83.98**	82.07	91.64	**92.60**	86.50	88.57	**85.09**	86.97
BASE:BKY+POS	80.76	76.24	81.76	80.34	**92.20**	87.64	82.95	88.13	82.89	83.66
BASE:BKY:RAW	79.14	69.78	80.38	78.99	87.32	81.44	73.28	79.51	78.94	78.75
This Model	80.75	82.65	81.06	81.47	91.65	89.54	86.12	93.16	81.06	85.27
This Model (v4)	82.06	84.42	82.38	**82.79**	91.95	90.56	**87.48**	**93.55**	81.93	86.34
2) gold POS / 5K training set										
IMS:SZEGED:CIS	**79.47**	**88.45**	**82.25**	74.78	91.64	**91.87**	80.10	88.18	**85.09**	84.65
BASE:BKY+POS	77.54	74.06	78.07	71.37	**92.20**	86.74	72.85	87.91	82.89	80.40
BASE:BKY:RAW	75.22	67.16	75.91	68.94	87.32	79.34	60.40	78.30	78.94	74.61
This Model	77.34	78.83	78.57	75.25	91.65	88.01	79.12	93.12	81.06	82.55
This Model (v4)	79.16	80.82	79.21	**76.97**	91.95	89.00	**80.82**	**93.43**	81.93	83.69
3) predicted POS / full training set										
IMS:SZEGED:CIS	**81.32**	**87.86**	**81.83**	**81.27**	89.46	**91.85**	**84.27**	87.55	**83.99**	85.49
BASE:BKY+POS	78.66	74.74	79.76	78.28	85.42	85.22	78.56	86.75	80.64	80.89
BASE:BKY:RAW	79.19	70.50	80.38	78.30	86.96	81.62	71.42	79.23	79.18	78.53
This Model	78.55	81.00	79.27	79.29	91.65	87.00	81.08	92.23	79.47	82.28
This Model (v4)	79.82	82.54	80.62	80.79	**91.95**	88.36	82.93	**92.67**	79.61	84.37
4) predicted POS / 5K training set										
IMS:SZEGED:CIS	**78.85**	**86.65**	**79.83**	73.61	89.46	**90.53**	**78.47**	87.46	**83.99**	83.21
BASE:BKY+POS	74.84	72.35	76.19	69.40	85.42	83.82	67.97	87.17	80.64	77.53
BASE:BKY:RAW	74.57	66.75	75.76	68.68	86.96	79.35	58.49	78.38	79.18	74.24
This Model	75.41	77.37	77.09	73.23	91.65	85.63	72.69	92.18	79.47	80.52
This Model (v4)	77.01	78.99	77.57	**75.01**	**91.95**	87.00	75.09	**92.66**	79.61	81.65

Table 1: Parseval results (the predicted POS were automaticaly predicted and provided with the corpus)

N input constituents $X_1, \ldots, X_N$, for each constituent X_n, the network tagger is fed with the concatenation of the constituent's word and tag representations and its $\frac{K}{2}-1$ left and right neighbors' word and tag representations. The network output a score for every possible BIOES-prefixed parsing tag.

2.5 Coherent BIOES Predictions

The next module of our architecture aggregates the BIOES-prefixed parsing tags from our tagger module in a coherent manner. It is implemented as a Viterbi decoding algorithm over a constrained graph G, which encodes all the possible valid sequences of BIOES-prefixed tags over constituents: e.g. B-A tags can only be followed by I-A or E-A tags, for any parsing label A. Each node of the graph is assigned a score produced by the previous neural network module (score for each BIOES-prefixed tag, and for each word). The score $S([t]_1^N, [X]_1^N, \theta)$ for a sequence of tags $[t]_1^N$ in the lattice G is simply obtained by summing scores along the path ($[X]_1^N$ being the input sequence of constituents and θ all the parameters of the model). This decoding is present only to ensure coherence in the predicted sequence of tags.

Both the composition network and tagging networks are trained by maximizing a likelihood over the training data using stochastic gradient ascent. For detailed information about the training procedure and how the training set is built please read (Legrand and Collobert, 2015).

The score $S([t]_1^N, [X]_1^N, \theta)$ of the true sequence of BIOES-prefixed tags $[t]_1^N$, given the input node sequence $[X]_1^N$ can be interpreted as a conditional probability by exponentiating this score (thus making it positive) and normalizing it with respect to all possible path scores. The log-probability of a sequence of tags $[t]_1^N$ for the input sequence of constituents $[X]_1^N$ is given by:

$$\log P([t]_1^N | [X]_1^N, \theta) = S([t]_1^N, [X]_1^N, \theta) - \log \left[\sum_{\forall [t']_1^N} \exp S([t']_1^N, [X]_1^N, \theta) \right] . \quad (1)$$

The second term of this equation (which correspond to the normalisation term) can be computed in linear time thanks to a recursion similar to the Viterbi algorithm (Rabiner, 1989).

3 Experiments

3.1 Corpus

The corpus used to conduct our experiments is the Statistical Parsing of Morphologically Rich Languages (SPMRL) corpus provided for the shared task 2014 (Seddah et al., 2014). It provides sentences and tree annotations for 9 different languages (Arabic, Basque, French, German, Hebrew, Hungarian, Korean, Polish,Swedish), coming from various sources (Sima'an et al., 2001; Tsarfaty, 2010; Goldberg, 2011; Tsarfaty, 2013;

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	avg
1) gold POS / full training set										
IMS:SZEGED:CIS	**88.61**	**94.90**	**92.51**	89.63	**92.84**	**95.01**	**91.30**	94.52	**91.46**	92.31
BASE:BKY+POS	87.85	91.55	91.74	88.47	92.69	92.52	90.82	92.81	90.76	91.02
BASE:BKY:RAW	87.05	89.71	91.22	87.77	91.29	90.62	87.11	90.58	88.97	89.37
This Model	87.77	81.83	90.44	92.58	84.62	94.62	84.19	95.48	88.44	88.89
This Model (v4)	87.50	82.24	91.03	**92.94**	84.55	94.75	84.66	**95.61**	88.34	89.07
2) gold POS / 5K training set										
IMS:SZEGED:CIS	**86.68**	**94.21**	**91.56**	85.74	**92.84**	**94.79**	**88.87**	94.17	**91.46**	91.15
BASE:BKY+POS	86.26	90.72	89.71	84.11	92.69	92.11	86.75	92.91	90.76	89.56
BASE:BKY:RAW	84.97	88.68	88.74	83.08	91.29	89.94	81.82	90.31	88.97	87.53
This Model	84.73	80.80	89.13	90.03	84.62	94.01	81.63	94.43	88.44	87.54
This Model (v4)	85.71	81.37	89.44	**90.76**	84.55	94.13	88.22	**95.56**	88.34	88.68
3) predicted POS / full training set										
IMS:SZEGED:CIS	**88.45**	**94.50**	**91.79**	89.32	**91.95**	**94.90**	**90.13**	94.11	**91.05**	91.80
BASE:BKY+POS	86.60	90.90	90.96	87.46	89.66	91.72	89.10	92.56	89.51	89.83
BASE:BKY:RAW	86.97	89.91	91.11	87.46	90.77	90.50	86.68	90.48	89.16	89.23
This Model	86.61	81.30	89.52	91.66	84.46	93.94	82.04	95.44	87.56	87.54
This Model (v4)	86.34	82.24	90.07	**92.32**	84.63	93.86	82.77	**95.38**	87.12	88.67
4) predicted POS / 5K training set										
IMS:SZEGED:CIS	**86.69**	**93.85**	**90.76**	85.20	**91.95**	**94.05**	**87.99**	93.99	**91.05**	90.61
BASE:BKY+POS	84.76	89.83	89.18	83.05	89.66	91.24	84.87	92.74	89.51	88.32
BASE:BKY:RAW	84.63	88.50	89.00	82.69	90.77	89.93	81.50	90.08	89.16	87.36
This Model	84.37	80.26	88.33	89.23	84.46	92.93	79.05	94.34	87.57	88.30
This Model (v4)	84.62	81.00	88.71	**89.98**	84.63	93.06	80.04	**95.45**	87.12	87.17

Table 2: Leaf-ancestor Results (the predicted POS were automaticaly predicted and provided with the corpus)

Choi et al., 1994; Choi, 2013; Vincze et al., 2010; Habash and Roth, 2009; Habash et al., 2009; S. Green and Manning, 2010; Maamouri et al., 2004; Brants et al., 2002; Seeker and Kuhn, 2012; I.Aduriz et al., 2003; Csendes et al., 2005; Abeillé et al., 2003; Świdziński and Woliński, 2010; Nivre et al., 2006).

For each language, the *gold* part-of-speech tags are provided as well as part-of-speech tags *predicted* by state-of-the-art taggers. A sub-corpus of 5k sentences (which correspond to the size of the smallest corpus) is also provided for each language. This leads to 4 different possible scenarios (see Table 1 and 2). For each of these scenarios, we evaluated our models using the Parseval (labelled f1-score) and the Leaf-ancestor metric.

3.2 Training details

Our systems are trained using a stochastic gradient descent over the available training data. Hyper-parameters were tuned on the validation set. The dimension for the words embedding and tag embeddings were respectively 100 and 20. The window size for the tagger is $K = 7$ (3 neighbours from each side). The size of the tagger's hidden layer were $H = 500$. All parameter were initialized randomly. As suggested in (Plaut and Hinton, 1987), the learning rate was divided by the size of the input vector of each layer. We used the same dropout regularization and voting procedure as in (Legrand and Collobert, 2015).

3.3 Results

Table 1 and 2 present the results obtained for the Parseval and the Leaf-ancestor metrics. We included a voting procedure using several models trained starting from different random initializations. At each iteration of the greedy parsing procedure, the BIOES-tag scores are averaged and the new node representations are computed for each model by composing the sub-tree representations corresponding to the given model, using its own compositional network.

We compare our system with the baseline provided with the task (Berkeley parser trained in two modes: with provided POS Tags (gold or predicted depending on dataset) and in Raw mode where the parser do its own POS tagging) and with the best (and only) participant of the task (Björkelund et al., 2014) which uses a product of PCFG-LA based model (Petrov, 2010) followed by a discriminative reranking (Charniak and Johnson, 2005).

4 Conclusion

In this paper, we showed that a simple greedy RNN-based model is able to outperform the baseline systems. Furthermore, we achieve significant improvements over the state-of-the-art model for several languages.

References

A. Abeillé, L. Clément, and F. Toussenel. 2003. Building a treebank for french. In Anne Abeillé, editor, *Treebanks*. Kluwer.

A. Björkelund, O. Çetinouglu, A. Faleńska, R. Farkas, T. Müller, W. Seeker, and Z. Szántó. 2014. Introducing the IMS-Wroclaw-Szeged-CIS entry at the SPMRL 2014 Shared Task: Reranking and Morpho-syntax meet Unlabeled Data. In *Proc. of SPMRL-SANCL*.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*.

E. Charniak and M. Johnson. 2005. Coarse-to-fine N-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

K. Choi, Y. Han, G. Young Han, and O. W. Kwon. 1994. Kaist tree bank project for korean: Present and future development. In *Proceedings of the International Workshop on Sharable Natural Language Resources*.

J. D. Choi. 2013. Preparing korean data for the shared task on parsing morphologically rich languages. In *Proceedings of the EMNLP 2013 Workshop of Statistical Parsing of Moprhologically-Rich Languages (SPMRL 2013)*.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor. 2005. The Szeged treebank. In *Text, Speech and Dialogue: Proceedings of TSD 2005*.

Y. Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ph.D. thesis, Ben Gurion University of the Negev.

N. Habash and R. Roth. 2009. Catib: The columbia arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*.

N. Habash, R. Faraj, and R. Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*.

I.Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank.

J. Legrand and R. Collobert. 2014. Recurrent greedy parsing with neural networks. In *European Conference on Machine Learning*.

J. Legrand and R. Collobert. 2015. Joint rnn-based greedy parsing and word composition. In *Proceedings of ICLR*, May.

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*, pages 1392–1395, I. Genoa.

S. Petrov. 2010. Products of random latent variable grammars. In *NAACL-HLT*.

D. C. Plaut and G. E. Hinton. 1987. Learning sets of filters using back-propagation. *Computer Speech and Language*.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286.

Spence S. Green and C. Manning. 2010. Better arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*.

D. Seddah, R. Tsarfaty, S. K´ubler, M. Candito, J Choi., R Farkas, J. Foster, I. Goenaga, K. Gojenola, Y. Goldberg, S. Green, N. Habash, M. Kuhlmann, W. Maier, J. Nivre, A. Przepiorkowski, R. Roth, W. Seeker, Y. Versley, V. Vincze, M. Woliński, A. Wróblewska, and E. Villemonte de la Clérgerie. 2013. Overview of the spmrl 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the spmrl 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*.

W. Seeker and J. Kuhn. 2012. Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.

K. Sima'an, A. Itai, Y. Winter, A. Altman, and N. Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In *Traitement Automatique des Langues*.

M. Świdziński and M. Woliński. 2010. Towards a bank of constituent parse trees for Polish. In *Text, Speech and Dialogue: 13th International Conference (TSD)*, Lecture Notes in Artificial Intelligence, pages 197—204. Springer.

R. Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.

R. Tsarfaty. 2013. *A Unified Morpho-Syntactic Scheme of Stanford Dependencies*. Proceedings of ACL.

V. Vincze, D. Szauter, A. Almási, G. Móra, Z. Alexin, and J. Csirik. 2010. Hungarian dependency treebank. In *LREC*.