Multilingual Lexicalized Constituency Parsing with Word-Level Auxiliary Tasks

Maximin Coavoux[1,2] and **Benoît Crabbé**[1,2,3]
[1]Univ. Paris Diderot, Sorbonne Paris Cité
[2]Laboratoire de linguistique formelle (LLF, CNRS)
[3]Institut Universitaire de France
{mcoavoux,bcrabbe}@linguist.univ-paris-diderot.fr

Abstract

We introduce a constituency parser based on a bi-LSTM encoder adapted from recent work (Cross and Huang, 2016b; Kiperwasser and Goldberg, 2016), which can incorporate a lower level character bi-LSTM (Ballesteros et al., 2015; Plank et al., 2016). We model two important interfaces of constituency parsing with auxiliary tasks supervised at the word level: (i) part-of-speech (POS) and morphological tagging, (ii) functional label prediction. On the SPMRL dataset, our parser obtains above state-of-the-art results on constituency parsing without requiring either predicted POS or morphological tags, and outputs labelled dependency trees.

1 Introduction

Recent work has shown the efficacy of bidirectional long short-term memory network (bi-LSTM) encoders in parsing (Kiperwasser and Goldberg, 2016; Cross and Huang, 2016b; Cross and Huang, 2016a). In these parsers, a bi-LSTM encodes the sentence and constructs context-aware embeddings for each word. Then a standard transition-based parser uses these embeddings as input to score parsing actions. In such architectures, the bi-LSTM component lends itself to auxiliary tasks of sequence prediction at the word level as illustrated for multilingual POS tagging by Plank et al. (2016).

In this paper, we present a constituency parsing model based on a bi-LSTM encoder, and use the bi-LSTM component to model two natural interfaces of constituency parsing — morphology and functional labelling — as word-level auxiliary tasks.

Morphological information is crucial for phrase structure parsing of morphologically rich languages (Seddah et al., 2013; Björkelund et al., 2013; Crabbé, 2015). Most multilingual parsers use a morphological tagger as the first step of a pipeline approach. As a first auxiliary task, we perform morphological analysis (prediction of the POS tags and of additional language-specific morphological attributes such as case, tense). We compare the resulting model to a pipeline approach.

As the second auxiliary task, we predict the functional label that links each word to its head. Overall, we evaluate to which extent these auxiliary tasks can both improve parsing and enrich the output of the parser. This paper makes the following contributions:

1. We introduce a single greedy parser which does not need predicted POS tags or morphological tags at inference time, and yet outperforms the best published results on the SPMRL dataset (Björkelund et al., 2014).[1]
2. We present the first experiments with multitask learning for multilingual lexicalized constituency parsing.
3. We further observe that a lexicalized constituency parser produces surprisingly accurate labelled dependency trees in a multilingual context.

2 Constituent Parsing with bi-LSTMs

Lexicalized transition-based constituent parsing generally derives from the work of Sagae and Lavie (2005) and subsequent work (Sagae and Lavie, 2006; Zhu et al., 2013, among others). We use the set of parse actions described by Sagae and Lavie (2005). It is a standard shift-reduce transition system which distinguishes left- and right- re-

[1]The code of the parser is available for download at https://github.com/mcoavoux/mtg/.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 331–336,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

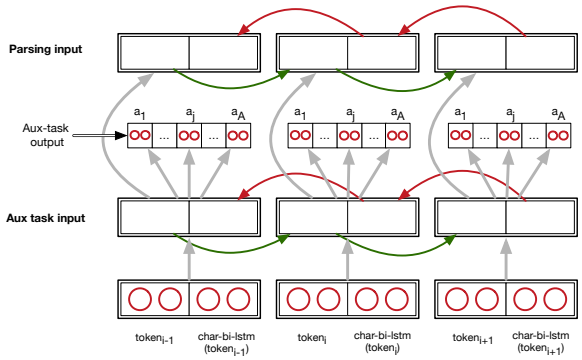


Figure 1: Deep bi-LSTM encoder with auxiliary tasks supervised at the first layer.

duce actions to assign heads to new constituents. We present the algorithm as a deduction system in Figure 3 of Appendix A.

Each action has a set of preconditions to make sure that the transition system always terminates and always outputs a well-formed lexicalized tree (Table 3 of Appendix A). For exemple, it is impossible to shift if B is empty.

To make the algorithm deterministic, we use a neural network to score actions at each parsing step. The first component of the network is a bi-LSTM encoder (Hochreiter and Schmidhuber, 1997) which builds contextual representations for every token in the sentence. The second component uses these representations as input to produce a distribution over possible actions at each parsing step. Both components are trained simultaneously.

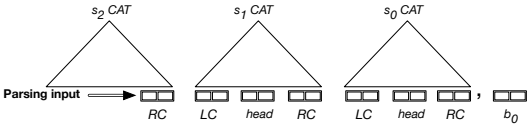
2.1 Bi-LSTM representation of the input

The use of a bi-LSTM encoder in parsing was proposed independently by Kiperwasser and Goldberg (2016) and Cross and Huang (2016a). Its role is to provide contextual representations for each token. In transition-based parsing, bi-LSTMs can give a finite representation of the potentially unbounded buffer (Dyer et al., 2015), and model span (Cross and Huang, 2016b).

Each token is a tuple of typed symbols, consisting minimally of a word-form. The other types of symbols are POS tags and language-dependent morphological attributes. Each type of symbol has its own embedding look-up table.

In our architecture (Figure 1), the input to the bi-LSTM encoder at step i is the concatenation of the embeddings of each typed symbol composing token i. The output for the same token is the concatenation of the forward and backward LSTM

Parser configuration:



Template set: s_0.*CAT*, s_0.*LC*, s_0.*RC*, s_0.*head*, s_1.*CAT*, s_1.*LC*, s_1.*RC*, s_1.*head*, s_2.*CAT*, s_2.*RC*, b_0

Figure 2: Parsing Templates. s and b respectively address symbols in the stack and the buffer.

states at step i. We use a two-layer bi-LSTM encoder, the input to the second layer being the output of the first one. Intuitively, the lower layer encodes a representation suitable for the word-level auxiliary tasks while the upper layer builds a representation for the parsing task itself.

On some experimental setups, we also use a single-layer of character bi-LSTM encoder for each word form, using the sequence of its characters, and concatenate its output to the input of the higher-level bi-LSTM, as has been done by Plank et al. (2016), Ballesteros et al. (2015), among others.

2.2 Output layers

To compute transition scores, we use a simple two-layer feedforward neural network. The input of this network consists of embeddings extracted from symbols in the stack (S) and the buffer (B). The symbols used are presented as feature templates in Figure 2.

These features are either instantiated with non-terminal embeddings or by the contextual token embedding produced by the bi-LSTM encoder. For example, s_0.L(eft)C(orner) is instantiated by the bi-LSTM output of the left-most token encompassed by the constituent s_0.

2.3 Auxiliary Tasks

We use the bi-LSTM states of the lower layer of the encoder to predict word-level attributes. Intuitively, the auxiliary tasks should make the lower layer representations good at predicting some word-level attributes known to be informative for parsing. The upper layer constructs more abstract features from these intermediate representations.

We experiment with two types of auxiliary tasks: morphology and functional labels.

		Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	Avg
Experimental conditions	Decoding	Development F1 (EVALBSPMRL)									
TOK+CLSTM	greedy	82.97	86.88	81.97	87.91	88.43	89.91	86.12	92.13	77.08	85.93
TOK+CLSTM+M	greedy	83.03	87.93	82.0	88.32	89.42	89.98	86.71	92.8	78.4	86.51
TOK+CLSTM+M+D	greedy	83.04	87.93	82.19	88.7	89.64	90.52	86.78	93.23	79.14	**86.8**
TOK	greedy	80.97	76.28	79.93	85.52	85.82	81.88	72.97	82.8	72.95	79.9
TOK+MMT	greedy	82.75	88.25	82.5	88.5	90.31	91.22	86.53	93.53	79.39	87.0
TOK+MMT+D	greedy	83.07	88.35	82.35	88.75	90.34	91.22	86.55	94.0	79.64	**87.14**
Experimental Conditions		Test F1 (EVALBSPMRL)									
TOK+CLSTM+M+D	greedy	**82.92**	87.87	82.1	85.12	89.19	90.95	85.89	92.67	83.44	86.68
TOK+MMT+D	greedy	82.77	**88.81**	82.49	**85.34**	**89.87**	**92.34**	**86.04**	**93.64**	84.0	**87.26**
Björkelund et al. (2014)	ens+reranker	81.32[a]	88.24	**82.53**	81.66	89.80	91.72	83.81	90.50	**85.50**	86.12

Table 1: Results on development and test corpora (SPMRL evaluator). [a]Björkelund et al. (2013).

		Arabic	Basque	French[c]	German	Hebrew[c]	Hungarian	Korean[c]	Polish[c]	Swedish[c]
Experimental Conditions	Decoding	Development results – POS-Tagging[d]								
TOK+CLSTM+M	greedy	97.66	95.7	97.58	98.39	95.71	98.06	94.42	97.02	96.88
MarMoT[a]	CRF+lexicons	97.38	97.02	97.61	98.10	97.09	98.72	94.03	98.12	97.27
		Test results – UAS/LAS								
TOK+CLSTM+M+D	greedy	81.5/78.7	75.8/68.9	88.0/83.1	67.1/64.1	84.5/75.3	74.5/69.5	89.9/87.3	88.2/80.0	86.3/76.5
TOK+MMT+D	greedy	81.3/78.6	76.8/71.2	**87.8/83.5**	67.2/64.7	**85.8/77.3**	75.9/72.0	**89.6/87.5**	**89.6/83.1**	**86.7/78.5**
Ballesteros et al. (2015)	greedy	**86.1/83.4**	**85.2/78.6**	86.2/82.0	**87.3/84.6**	80.7/72.7	**80.9/76.3**	88.4/86.3	87.1/79.8	83.4/76.4
Best published[b]	ens+reranker	88.3/86.2	90.0/85.7	89.0/85.7	91.6/89.7	87.4/81.7	89.8/86.1	89.1/87.3	91.8/87.1	88.5/82.8

Table 2: Dependency and tagging results. [a]Uses external morphological lexicons (Björkelund et al., 2013). [b]Either Björkelund et al. (2013) or Björkelund et al. (2014). [c]Languages with few head mismatches between the dependency and the constituency corpora (Crabbé, 2015). [d]Tagging is evaluated with the dependency treebanks (the tagsets used in the constituency treebanks might differ).

Morphology Each token is annotated with its tag and a sequence of language-specific morphological attributes such as gender, case or tense. Whereas the tagging has often been addressed with parsing as a joint task, to the best of our knowledge, no model has proposed to perform full morphological analysis in a multi-task framework. For this task, we use one softmax output layer per available morphological attribute, including POS tags (Figure 1).

Functional Labels Both to improve constituency parsing and to enrich constituency trees with functional information, we propose a novel auxiliary task consisting in predicting the functional label of a token, i.e. its syntactic role with respect to its head. This task is constructed as a simple sequence prediction task without any information about the parse tree.

2.4 Loss function

The objective function for a single sentence w_1^n whose gold derivation is the sequence of actions a_1^T is defined as follows:

$$L(a_1^T, w_1^n; \boldsymbol{\theta}) = \sum_{i=1}^{T} \log p(a_i | a_1, \dots a_{i-1}; w_1^n, \boldsymbol{\theta}) + \sum_{i=1}^{n} \sum_{j=1}^{A} \log p(w_{i,j} | w_1^n; \boldsymbol{\theta})$$

where $w_{i,j}$ denotes the attribute j of token i, A is the total number of attributes used as auxiliary tasks and $\boldsymbol{\theta}$ is the set of all parameters.

3 Experiments

Our model combine constituency parsing with two of its natural interfaces, morphology and functional structure. We designed experiments to assess to which extent modelling these interfaces as auxiliary tasks can improve parsing and enrich the output of the parser.

We performed two sets of experiments to handle two questions: we compare the integration of morphological information as respectively provided by an external tagger in a pipeline architecture or as an auxiliary prediction task for the neural model. For each of those setup, we test to which

extent we can also accurately predict functional labels as an auxiliary task.

In a first set of experiments, we evaluated the model with a character-level bi-LSTM and either no auxiliary task (TOK+CLSTM), morphological tagging as an auxiliary task (TOK+CLSTM+M), or morphological tagging and dependency label predictions as auxiliary tasks (TOK+CLSTM+M+D). In those three models, the input to the sentence-level bi-LSTM is the concatenation of a word embedding and a character-based embedding.

In a second set of experiments, the input to the sentence-level bi-LSTM is either a word embedding (TOK) or the concatenation of a word embedding and embeddings for each available morphological tag (TOK+MMT), predicted by a morphological tagger (Mueller et al., 2013, MarMoT). We compare the latter setup with an additional functional prediction as auxiliary task (TOK+MMT+D).

This last model will give upper-bound accuracies against which we can compare the model with all auxiliary tasks (TOK+CLSTM+M+D), which is the focus of the paper.

Data We evaluate our models on the SPMRL dataset (Seddah et al., 2013). This dataset contains constituency and dependency treebanks aligned at the word level for 9 morphologically righ languages. These treebanks are annotated with POS tags and morphological attributes (such as case, mood, tense, number).

In the experiments where morphology is predicted as an auxiliary task, we use the gold tags and morphological annotations at training time and none of this information at test time.

In the other experiments, we use the POS and morphological tags predicted by MarMoT (Mueller et al., 2013),[2] for training and parsing. Following Björkelund et al. (2013), we used fine pos-tags for all languages except Korean.

As our parsing model is lexicalized, each constituent in the training set must be annotated with its head. We used the procedure described by Crabbé (2015) to do so. This procedure uses the alignement between constituency trees and dependency trees to determine the head of each phrase, and uses heuristics to solve mismatch cases.[3] We binarize trees with an order-0 head-Markovization and collapse unary productions except those which produce pre-terminals.

Protocol We trained every model with ASGD (Polyak and Juditsky, 1992) and shuffle the training set before each iteration. When using auxiliary losses, we repeat the two following steps. First, we make predictions for every auxiliary task, assign POS tags to tokens (POS tags of tokens are non-terminals once shifted onto S), then backpropagate and update the parameters. In the second step, we compute the primary loss (over the gold sequence of actions for the current sentence), then backpropagate the gradient and update the parameters.

For each model, we calibrated the learning rate and the number of iterations on the development set, but did not do any other hyperparameter tuning. The complete list of hyperparameters used is shown in Table 4 in Appendix A.

Results and Discussion Results on development and test sets are presented in Table 1. First we observe that our baseline (TOK+CLSTM) is nearly as accurate as the best published results on the SPMRL dataset. The use of morphology as auxiliary tasks (TOK+CLSTM+M) improves the baseline by 0.5 F1 on average on the test sets. While being greedy, and needing neither predicted POS nor morphological tags, the resulting parser outperforms the product of grammar and reranker combination of Björkelund et al. (2014).

Furthermore, on average, it is only 0.5 F1 behind the model which uses predicted morphology as input to the bi-LSTM (TOK+MMT). Across languages, the performance difference between the two models can be partly explained by the difference in tagging accuracy (Table 2). The TOK+CLSTM+M model matches MarMoT tagging results for several languages, but is not as good overall. MarMoT uses morphological lexicons as an additional source of information, which might be crucial for languages such as Basque.

Second, the dependency label auxiliary task improves constituency parsing by a small but consistent margin. As our model is lexicalized, it is able to output unlabelled dependency trees. As a byproduct of this task, we can obtain labelled dependency trees instead. Thus, we also evaluate the output of our parser against the dependency corpora using the evaluator provided with the shared task. Results are shown in Table 2. Our parser outperforms Ballesteros et al. (2015), the best pub-

[2] These are available on MarMoT website.

[3] Mismatches could be caused by irreducible structure difference between both treebanks (Crabbé, 2015).

lished results with a greedy parser, on 5 languages out of 9. Unsurprisingly, these languages correspond to the corpora, identified by Crabbé (2015), which contain very few mismatch cases between the dependency and the constituency treebank.

This result is in keeping with Cer et al. (2010) who has shown that constituency parsers are very good at recovering dependency structures for English. Our experiments confirm this finding in a novel multilingual setting where labelled dependency trees are directly predicted by the parser, rather than obtained by conversion of predicted constituency trees.

4 Conclusion

We have investigated to which extent modelling morphological analysis and functional label prediction as auxiliary tasks could benefit parsing. The parser we described does not need predicted morphological information at test time, and yet obtains state-of-the-art results in constituency parsing. Since the parser is lexicalized, it models both constituency and dependency and can therefore output directly labelled dependency trees without involving any additional conversion heuristic.

Acknowledgments

We thank Djamé Seddah, Héctor Martínez Alonso and Chloé Braud for helpful comments. This work was partially funded by the Agence Nationale de la Recherche (ParSiTi project, ANR-16-CE33-0021).

A Supplemental Material

Action	Conditions
SH	B is not empty.
U-X	The last action is SHIFT. X is an axiom iff this is a one-word sentence.
(R|L)-X	S has at least 2 elements. X is an axiom iff B is empty, and S has exactly one element. If X is a temporary symbol and if B is empty, s_2 must not be a temporary symbol.
R-X	s_1 is not a temporary symbol.
L-X	s_0 is not a temporary symbol.

Table 3: List of preconditions on actions. Temporary symbols are symbols introduced by the binarization process.

$$\text{SH(IFT)} \quad \frac{\langle S, w|B\rangle}{\langle S|w, B\rangle}$$

$$\text{(REDUCE-)U(NARY)-X} \quad \frac{\langle S|s_0[h], B\rangle}{\langle S|X[h], B\rangle}$$

$$\text{(REDUCE-)R(IGHT)-X} \quad \frac{\langle S|s_1[h]|s_0[h'], B\rangle}{\langle S|X[h'], B\rangle}$$

$$\text{(REDUCE-)L(EFT)-X} \quad \frac{\langle S|s_1[h]|s_0[h'], B\rangle}{\langle S|X[h], B\rangle}$$

Figure 3: Lexicalized shift-reduce transition system. $X[h]$ denotes a non-terminal X and its head h. Each action has a set of preconditions to make sure that the transition system always terminates and always outputs a well-formed lexicalized tree. These preconditions are described in Table 3 of Appendix A.

Hyperparameters	Values
Optimisation	
Iterations	$\{4, 8, 12, \ldots 28, 30\}$
Initial learning rate	$\{0.01, 0.02\}$
Learning rate decay constant	10^{-6}
Hard gradient clipping	5.0
Gaussian noise σ	0.01
Parameter initialisation	Xavier initialisation
Embedding initialisation	Uniform($[-0.01, 0.01]$)
Output layers	
Number of hidden layers	2
Size of hidden layers	128
Activation	rectifiers
Word level bi-LSTM	
Depth	2
Size of LSTM states	128
Word embeddings[a]	32
Non-terminal embeddings	16
Morphological embeddings[b]	4, 8 or 16[c]
Char-level bi-LSTM[a]	
Depth	1
Size of LSTM states	32
Character embeddings	32

Table 4: Hyperparameters.

[a] Following Kiperwasser and Goldberg (2016), we stochastically replace a word by an unknown symbol with probability $p(w) = \frac{\alpha}{\#\{w\}+\alpha}$, where $\#\{w\}$ is the raw frequency of w in the training corpus. Following Cross and Huang (2016b), we used $\alpha = 0.8375$.

[b] When applicable.

[c] Depending on number of possible values for this attribute.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.

Anders Björkelund, Özlem Çetinoğlu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, Seattle, Washington, USA, October. Association for Computational Linguistics.

Anders Björkelund, Özlem Çetinoğlu, Agnieszka Faleńska, Richárd Farkas, Thomas Mueller, Wolfgang Seeker, and Zsolt Szántó. 2014. Introducing the ims-wrocław-szeged-cis entry at the spmrl 2014 shared task: Reranking and morpho-syntax meet unlabeled data. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 97–102, Dublin, Ireland, August. Dublin City University.

Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*.

Benoit Crabbé. 2015. Multilingual discriminative lexicalized phrase structure parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1847–1856, Lisbon, Portugal, September. Association for Computational Linguistics.

James Cross and Liang Huang. 2016a. Incremental parsing with minimal features using bi-directional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 32–37, Berlin, Germany, August. Association for Computational Linguistics.

James Cross and Liang Huang. 2016b. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas, November. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics – Volume 4, Issue 1*, pages 313–327.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August. Association for Computational Linguistics.

B. T. Polyak and A. B. Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July.

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132. Association for Computational Linguistics.

Kenji Sagae and Alon Lavie. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 691–698. Association for Computational Linguistics.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *ACL (1)*, pages 434–443. The Association for Computer Linguistics.