Thang Le Quang
Student at SoICT - HUST,
No 1, Dai Co Viet, Hanoi
lelightwin@gmail.com

Yusuke Miyao
National Informatic Institute,
Tokyo 101-8430, Japan
Yusuke@nii.ac.jp

Hiroshi Noji
National Informatic Institute,
Tokyo 101-8430, Japan
Noji@nii.ac.jp

Abstract

The approach of global discriminative shift-reduce parsing is quite successful, giving state-of-the-art performance both for constituency and dependency parsing problem, but most of existing parsers rely on some approximations such as beam search, which discards the optimality. In this paper, we explain how exact search become possible for constituency shift-reduce parsing and explore the effect of optimality empirically. This is the first work on exact search for shift-reduce parsing in a global discriminative setting. The key components of our system are the new feature templates to reduce the time complexity and several A* heuristics for keeping a tractable runtime. Experiments on the standard Penn Treebank verifies the effectiveness of exact search: our new features is not as strong as richer features with the same beam size, but with A* search, it produces F-score of 91.1% that other beam-based systems cannot reach even with very large beam sizes.

1 Introduction

Transition-based approaches for parsing have gained much popularity in the last decade. Their main advantage is to reduce the parsing problem into successive local classifications, where a correct action in each step would lead to the correct analysis. This design has both pros and cons. Since each decision is based on a local classifier, a parser can exploit arbitrary features from a snapshot in a parsing process that would be useful for disambiguation. The cost of this effectiveness is its limited search quality: a parser can no longer reach the globally optimal solution due to the enormous search space. Commonly, most of existing parsers employ greedy search (Nivre, 2003) or beam search (Zhang and Clark, 2009), which enables linear parsing runtime while there is always a possibility of search errors.

There are many advancement of transition-based parsing which has been done. For dependency parsing, Zhang and Nivre (2011) reported the state-of-the-art accuracy with very rich feature sets from higher-order information which is quite difficult to access by classical CYK-based approach. A similar extension for constituency parsing was studied in Zhu et al. (2013), where larger improvement was reported with many types of external semi-supervised features. The classified-based design also makes it easy to extend to joint modeling of several linguistic analysis, e.g., joint parsing with POS tagging, which further boosts the performance (Hatori et al., 2012; Bohnet et al., 2013; Wang and Xue, 2014). However, as we mentioned earlier, all these extensions have been done by ignoring the search optimality. Due to this approximation, a parser can utilize arbitrary information, which produces very high accuracies.

Unlike these recent advancements, we focus on the *exactness* of search to achieve very accurate parsing. Our work is largely influenced by the best-first dynamic shift-reduce parser introduced in Zhao et al. (2013). However, we would like to perform best first search in a global settings for both training and parsing. Our main strategy can be described as follows: 1) We try to reduce the space complexity of best first search by minimizing elementary features; and 2) to achieve a practical runtime, we improve the search quality with some A* heuristics. The practical interest is its performance compared to inexact method with very rich features. Rather surprisingly, although the performance of the simple feature model is worse than the richer feature model in the same beam size, our A* system with the simple feature model reaches the score that outperforms the other beam-based parsers even with quite large beam sizes such as 64. This result indicates a transition-based parser with exact search could be a practical choice if appropriate features are used. The resulting system is rather similar to the graph-based, or CYK-based approach, but our final system get much higher accuracy than the state-of-the-art CYK-based parsers such as Berkeley parser (Petrov et al., 2006), achieving 91.1% F-score on the standard Penn Treebank test set.

2 Background

Summarizing the basic ideas of previous works here.

2.1 Shift-reduce parsing

- The structure of baseline shift-reduce parsing (stack, queue, state items...)

- The four conventional shift-reduce actions (for both constituent parser and dependency parser):
  - SHIFT.
  - U-REDUCE(X): In dependency parsing, we do not have this one.
  - B-REDUCE$_L$(X): In dependency parsing, we have only one "X" in grammar.
  - B-REDUCE$_R$(X).

- The process of training shift-reduce parser with average structured perceptron as in (**?**).

2.2 Models of Shift-Reduce Parsing

- present the structured perceptron model as global training for Shift-Reduce Parsing.

- present the Maxent model as local training for Shift-Reduce Parsing.

2.3 Best-first parsing

- The best-first shift-reduce parser of (**?**) which was locally trained by Maxent model, it did not use dynamic programing.

- The best-first shift-reduce parser of (**?**): also trained by Maxent, but using dynamic programing with lazy expansion technical to improve decoding speed.

2.4 A* parsing

- Give some briefly introduction on A* parsing.

3 Basic Architecture

The goal of this section is to establish our basic parsing system other than our improvements explained in Section 4.

3.1 Unary merging actions

- Modifying the convention shift-reduce actions by merging UNARY action with SHIFT and B-REDUCE actions:
  - SHIFT_UNARY(X): in case of X=NULL, this is a conventional SHIFT action.
  - B-REDUCE(Y)_UNARY(X): in case of X=NULL, this is a conventional B-REDUCE action.

- Advantage:
  - Normalize the number of actions for each parse tree will always be $2n$, with n is the length of input sentence.
  - It is similar to padding methods in (**?**), but more consistent. In addition, the padding method in (**?**) cannot be applied into BFS.
  - In dependency parsing, there is no need for using unary merging actions or padding method.

3.2 Global Linear Best-First Parsing

- The deductive system of our Best-First Parsing.

- Adding an offset to each action score to solve the negative-score problem.

3.3 Search Quality

- Making comparison experiment between the search quality of Best-First Maxent and Best-First structured perceptron. The experiment has been taken on section 22 with

- Based on the results, indicate that Zhao's parser had worked because Maxent model is sparser than Structured perceptron model.

4 Extended Architecture

The goal of this section is to build a parser balancing the model's expressiveness and tractable runtime. Our strategy is changing features to exploit from very different parts from other SR parsers and improving search.

4.1 Span Features

- Discuss the time complexity problem in feature template, we need a simplified features which can guarantee the effectiveness without increasing the complexity.
- Present the span features from (**?**), which focused on span information within a non-terminal node.

4.2 A* search

- Present the *grammar projection* heuristic which is adapted from (**?**). Concerning about the *context summarization*, it cannot be used in incremental parsing, so we ignore it. The idea of how we can use *context summarization* will be the future research.
- Present the *reduced constituent* heuristic: this is our proposed heuristics for shift-reduce design. The key is to reduce the original space into small space with lesser constituents. This one is an efficient heuristic for A* search. The only problem is that it takes time to calculate.
- Present the hierarchical A* heuristic which combined *grammar projection* with *reduced constituent*.

5 Experiment

Describes experimental settings:

- We use Stanford PoS tagger to produce input for parsing.
- Section 22 for experiments on effectiveness of A* and span features. Section 23 for the final results comparing to the others.
- Configuration of computer which runs the experiments.

5.1 Speed and Accuracy Comparison

- Comparing search quality between difference A* heuristic on span features. (Figure 1)
- Comparing search quality between A* parser with beam search parser with different beam size on span features. (Figure 2)
- Total performance comparison between our A* parser with beam search on span features and Zhang's baseline in both accuracy and speed.

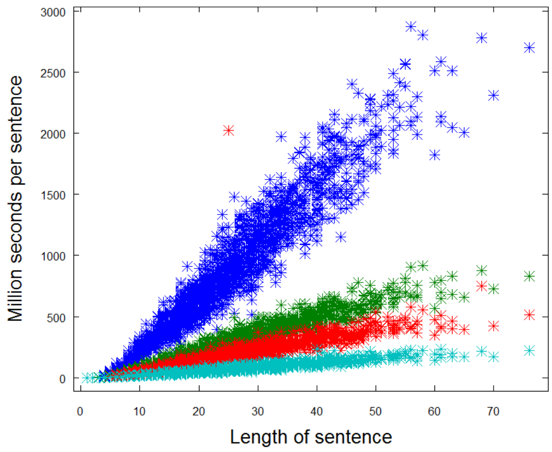

Figure 1: The parsing time comparison between different A* heuristic. The blue one is non-heuristic best first search, the red one is the best first search with reduced-constituent heuristic, the green one is the best first search with grammar-projection heuristic, and the cyan one is the hybrid heuristic between grammar-projection and less-constituent.

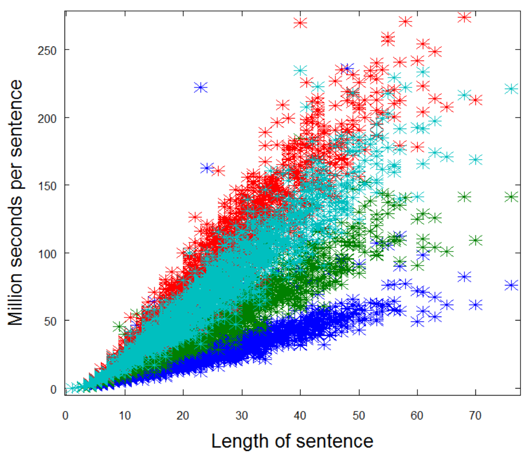

Figure 2: The parsing time comparison between A* and beam search. The blue one is beam size = 16, the green one is beam size = 32, the red one is beam size = 64, and the cyan one is our hybrid A*.

Table 1: The results of experiment comparing between A* parser with different beam search systems

System		Baseline Features			Span Features		
		F-score		*Speed*	*F-score*		*Speed*
		non-DP	*DP*	*(sentence/s)*	*non-DP*	*DP*	*(sentence/s)*
Beam Search	beam = 16	89.1	90.1	34.6	88.6	89.9	31.9
	beam = 32	89.6	89.9	20	89.3	90.2	17
	beam = 64	89.7	90.2	10.6	89.6	90.2	9.1
A* Search		N/A	N/A	N/A	N/A	90.7	13.6
Best First Search		N/A	N/A	N/A	N/A	90.7	1.12

Table 2: The final results on section 23 of English Penn Treebank

System	**Speed (sentence/s)**	**LR**	**LP**	**F1**
Johnson and Charniak (2005)	2.1	91.2	91.8	91.5
Mc Closky (2006)	1.2	92.2	92.6	92.4
Berkeley parser	6.1	90.1	90.3	90.2
Stanford parser (2014)	3.3	90.3	90.7	90.5
Zhang Yue (2012) - baseline	107.5	90.0	89.9	89.9
Zhang Yue(2012) - baseline+padding	93.4	90.2	90.7	90.4
Zhang Yue(2012) - baseline+padding+semi	47.6	91.1	91.5	91.3
Sagae & Lavie (2005)*	3.7	86.0	86.1	86.0
Sagae & Lavie(2006)*	2.2	88.1	87.8	87.9
This paper	**13.6**	**90.9**	**91.2**	**91.1**

5.2 Final Results

The results of final experiment to show that our parser can achieve a state-of-the-art performance with only simplified feature template.

6 Discussion

It is clearly to see that inexact search cannot exploit the potential of such feature template like exact one. In the future, we will study how to design a good A* parser (hierarchical A* with reduced-constituent approach) which can apply arbitrary feature sets without taking too much parsing time.

7 Conclusion

- Conclude about our research.
- Some future works.

References

Bernd Bohnet, Joakim Nivre, Igor M. Boguslavsky, Richárd Farkas, and Jan Hajič. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1(Oct):429–440.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1045–1053, Jeju Island, Korea, July. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiguo Wang and Nianwen Xue. 2014. Joint pos tagging and transition-based constituent parsing in chinese with non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 733–742, Baltimore, Maryland, June. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris, France, October. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.

Kai Zhao, James Cross, and Liang Huang. 2013. Optimal incremental parsing via best-first dynamic programming. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 758–768, 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 *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 434–443, Sofia, Bulgaria, August. Association for Computational Linguistics.