Thang Le Quang
SoICT - HUST
lelightwin@gmail.com

Yusuke Miyao
National Informatic Institute
Yusuke@nii.ac.jp

Hiroshi Noji
National Informatic Institute
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 feature model is not as strong as richer feature model with the same beam size, but with A* search, it reaches the score that other beam-based systems cannot reach even with very large beam sizes.

1 Introduction

- Introduce the transition-based approaches for parsing and its limitation with search errors.
- Confirm the purpose of this paper is to focus on exact search, and present the summarization of key ideas.
- Briefly discuss about some previous work on exact search such as (Zhao et al., 2013), (Huang and Sagae, 2010), (Sagae and Alon, 2006)... present the difference between their system and ours. Strongly affirm that their system are either local model or inexact search, whereas our system is the global model with exact search.

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 (Collins and Roark, 2004).

2.2 Best-first search

- The best-first shift-reduce parser of (Sagae and Alon, 2006) which was locally trained by Maxent model, it did not use dynamic programing.
- The idea of using dynamic programing of (Huang and Sagae, 2010).
- The best-first shift-reduce parser of (Zhao et al., 2013): also trained by Maxent, but using dynamic programing with lazy expansion technical to improve decoding speed.

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 (Zhu et al., 2012), but more consistent.
  - 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 (Hall et al., 2014), which focused on span information within a non-terminal node.

4.2 A* search

- Give some briefly introduction on previous research on A* parsing such as: (Klein and Manning, 2003a).
- Present the *grammar projection* heuristic which is adapted from (Klein and Manning, 2003b). 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)

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	89.57	89.9	20	89.3	90.2	17
	beam = 64	89.7	90.2	10.6	89.55	90.15	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

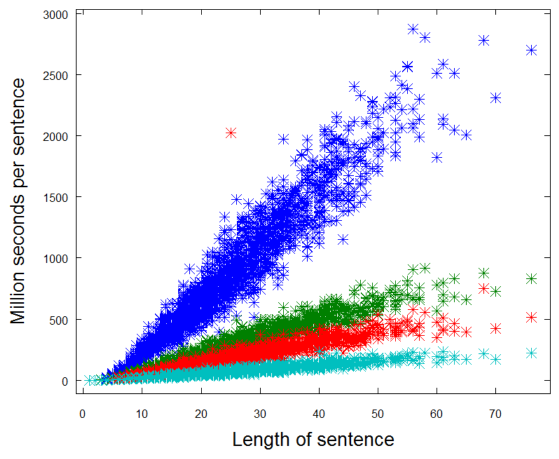

Figure 1: The parsing time comparison between different A* heuristic. The blue one is non-heuristic best first search, the red one is the best first search with reduced-constituent heuristic, the green one is the best first search with grammar-projection heuristic, and the cyan one is the hybrid heuristic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

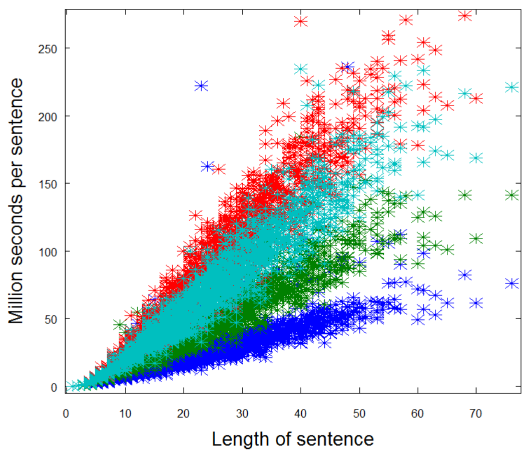

Figure 2: The parsing time comparison between A* and beam search. The blue one is beam size = 16, the green one is beam size = 32, the red one is beam size = 64, and the cyan one is our hybrid A*.

- 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.

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-

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**

constituent approach) which can apply arbitrary feature sets without taking too much parsing time.

7 Conclusion

- Conclude about our research.
- Some future works.

References

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*, Stroudsburg, PA, USA.

David Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 228–337, Baltimore, Maryland.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL 2010.*

Dan Klein and Chris Manning. 2003a. A* parsing: Fast exact viterbi parse selection. In *Proceedings of HLT-NAACL.*

Dan Klein and Chris Manning. 2003b. Accurate unlexicalized parsing. In *Proceedings of ACL.*

Kenji Sagae and Lavie Alon. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL*, pages 691–698.

Kai Zhao, Jame Cross, and Liang Huang. 2013. Optimal incremental parsing via best-first dynamic programming. In *Proceedings of EMNLP 2013.*

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2012. 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.