A Generation Framework for Grammar Development

Shashi Narayan
School of Informatics
University of Edinburgh
shashi.narayan@ed.ac.uk

Claire Gardent
CNRS, LORIA, UMR 7503
Vandoeuvre-lès-Nancy, F-54500, France
claire.gardent@loria.fr

Abstract

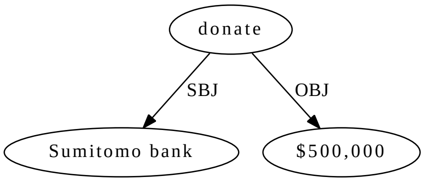
We developed a general framework for the development of a symbolic (hand-written) feature-based lexicalised tree-adjoining grammar (FB-LTAG). We choose natural language generation, surface realisation in particular, to question the capabilities of the grammar in terms of both accuracy and robustness. Our framework combines an optimised surface realiser with efficient error mining techniques. While generating from a large data set provided by the Generation Challenge Surface Realisation task, we improve both accuracy and robustness of our grammar significantly.

1 Introduction

We present a framework for the development of a hand-written feature-based lexicalised tree-adjoining grammar(FB-LTAG) for English. We use an XMG (Crabbé et al., 2013) based FB-LTAG (Alahverdzhieva, 2008) in our experiments. Unlike XTAG (The XTAG Research Group, 2001) where each rule is described manually, XMG based grammar uses a compact way of grammar writing using a meta grammar. Only rules in the meta grammar are described manually, these rules are later combined to generate the FB-LTAG grammar. Our grammar consists of roughly 1000 trees with a linguistic coverage similar to that of XTAG.

(1) **Input:** Shallow dependency structure

```
SROOT 1 0 donate cpos=vbd
SBJ 2 1 Sumitomo bank cpos=nn
OBJ 3 1 $500,000 cpos=cd
```



Output: Sumitomo bank donated $500,000.

In the literature, parsing has been proposed for testing symbolic grammars in terms of both accuracy and coverage (Thurmair, 1990; van Noord, 2004; Sagot and de la Clergerie, 2006; de Kok et al., 2009). In this paper, we choose natural language generation (NLG), surface realisation in particular, to question the capabilities of the grammar while generating from a large set of shallow dependency trees (similar to the input shown in Example 1) provided by the Generation Challenge: Surface Realisation Shared task (SR Task, in short) (Belz et al., 2011). This SR benchmark consists of 26,725 sentences varying from minimum length 1 to maximum length 134 with an average length of 22. The purpose behind using this dataset was to check the robustness and the accuracy of the grammar, and also the efficiency of the surface realiser on a large benchmark.

2 Our Framework

Our framework consists of an optimised surface realisation algorithm together with error mining techniques. We describe them briefly in this section.

Structure-driven Lexicalist Generation. Symbolic surface realisation is very prone to the combinatorial problem (Kay, 1996) because of (i) strong lexical ambiguity, (ii) the lack of order information in the input, and (iii) intersective modifiers. We developed an optimised algorithm (TDBU-PAR, (Narayan and Gardent, 2012b)) which combines techniques and ideas from the head-driven (Shieber et al., 1990) and the lexicalist approaches (Espinosa et al., 2010; Carroll and Oepen, 2005; Gardent and Kow, 2005). On the one hand, rule selection is guided, as in the lexicalist approach, by the elementary units present in the input rather than by its structure. On the other hand, the structure of the input is used to provide top-down guidance for the search and thereby restrict the combinatorics. To further improve efficiency, the algorithm integrates three additional optimisation techniques: (i) polarity filtering from the lexicalist approach (Bonfante et al., 2004; Gardent and Kow, 2007); (ii) the use of a language model to prune competing intermediate substructures; and (iii) simultaneous rather than sequential parallelised top-down predictions.

Table 1 shows the advantages of the proposed algorithm. We compared our proposed system (TDBU-PAR) with a baseline system (BASELINE, (Narayan, 2011)). We saw that whereas BASELINE times out for longer sentences, the newly proposed system TDBU-PAR remains stable. Our system successfully terminates on all the SR Task input (26,725 sentences) with a coverage a of 38.73% and with a BLEU score of 0.675 for covered sentences.

Sentence				Algorithm BASELINE	TDBU-PAR
Length range (L)	$S(0-5)$	Total	1084	0.85	1.53
		Succ	985	0.87	1.66
	$S(6-10)$	Total	2232	10.90	2.56
		Succ	1477	10.76	3.28
	$S(11-20)$	Total	5705	110.07	2.66
		Succ	520	97.52	4.14
	$S(All)$	Total	13661	-	2.57
		Succ	2744	-	2.78

Table 1: Comparison between generation times (seconds). To make these comparisons possible, average maximum arity of words present in dependency trees is 3.

Meaningful Error Mining for Tree-structured Data. In recent years, error mining approaches (van Noord, 2004; Sagot and de la Clergerie, 2006; de Kok et al., 2009) were developed to help identify the most likely sources of parsing failures in parsing systems using handcrafted grammars and lexicons. However the techniques they use to enumerate and count n-grams build on the sequential nature of a text corpus and do not easily extend to structured data. In addition, they generate a flat list of suspicious forms ranked by decreasing order of suspicion. There is no clear overview of how the various suspicious forms interact and as a result, the linguist must analyse all error cases one by one instead of focusing on improving sets of related error cases.

To improve our coverage (38.73%), we introduced two error mining algorithms in our framework: one, an algorithm for mining trees (Gardent and Narayan, 2012) which we apply to detect the most likely sources of generation failure and two, an algorithm that structures the output of error mining into a tree (called, suspicion tree) (Narayan and Gardent, 2012a), highlighting the relationships between suspicious forms. The first algorithm adapts from van Noord (2004) and from a complete and computationally efficient algorithm developed by Chi et al. (2004) for discovering frequently occurring subtrees in a database of labelled unordered trees. The second algorithm resembles that of a decision tree building using ID3 algorithm (Quinlan, 1986).

Our error mining algorithms allow us to do error analysis in a linguistically meaningful way and permit identifying not only errors in the generation system (grammar, lexicon) but also mismatches between the structures contained in the input and the input structures expected by our generator as well as a few idiosyncrasies/error in the input data.

S-ALL (26725)		
Before	Fail (Failed %)	16374 (61.2%)
	BLEU	0.675
After	Fail (Failed %)	5078 (19%)
	BLEU	0.72

Table 2: Number of errors before and after.

Further improvements While efficient surface realisation and error mining helped improve coverage and precision, they also uncovered two additional sources of errors: (i) a formal error related to how multiple adjunction is represented in FB-TAG (Gardent and Narayan, 2015) and (ii) some errors with generating elliptical sentences (Gardent and Narayan, 2013). Addressing these issues lead to further improvements (Table 2) whereby, on the testset provided by the SR shared task, the final system has a BLEU score of 0.72 and a coverage of 0.81.

3 Conclusion

Efficient surface realisation and focused error mining allowed us to develop a large scale natural language generation system which is efficient, robust and accurate while integrating a symbolic grammar and lexicon. While this system was evaluated on newspaper text, because it relies on a symbolic grammar and lexicon, it should straightforwardly extend to other text genre. Similarly, while it was developed for unordered dependency tree inputs, it could also be of use in NLG applications which take as input tree-shaped data.

References

Katya Alahverdzhieva. 2008. XTAG using XMG. Master's thesis, Université de Nancy.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, Nancy, France.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2004. Polarization and abstraction of grammatical formalisms as methods for lexical disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland.

John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP)*, Jeju Island, Korea.

Yun Chi, Yirong Yang, and Richard R. Muntz. 2004. Hybridtreeminer: An efficient algorithm for mining frequent rooted trees and free trees using canonical form. In *Proceedings of the 16th International Conference on and Statistical Database Management (SSDBM)*, pages 11–20, Santorini Island, Greece. IEEE Computer Society.

Benoît Crabbé, Denys Duchier, Claire Gardent, Joseph Le Roux, and Yannick Parmentier. 2013. XMG : eXtensible MetaGrammar. *Computational Linguistics*, 39(3):1–66, September.

Daniël de Kok, Jianqiang Ma, and Gertjan van Noord. 2009. A generalized method for iterative error mining in parsing results. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 71–79, Suntec, Singapore. Association for Computational Linguistics.

Dominic Espinosa, Rajakrishnan Rajkumar, Michael White, and Shoshana Berleant. 2010. Further meta-evaluation of broad-coverage surface realization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 564–574, Cambridge, MA, October. Association for Computational Linguistics.

Claire Gardent and Eric Kow. 2005. Generating and selecting grammatical paraphrases. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)*, Aberdeen, Scotland, Aug.

Claire Gardent and Eric Kow. 2007. A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 328–335, Prague, Czech Republic, June. Association for Computational Linguistics.

Claire Gardent and Shashi Narayan. 2012. Error mining on dependency trees. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL, Volume 1: Long Papers)*, pages 592–600, Jeju Island, Korea, July. Association for Computational Linguistics.

Claire Gardent and Shashi Narayan. 2013. Generating elliptic coordination. In *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG)*, pages 40–50, Sofia Bulgaria, August. Association for Computational Linguistics.

Claire Gardent and Shashi Narayan. 2015. Multiple adjunction in Feature-based Tree-Adjoining Grammar. *Computational Linguistics*, 41(1).

Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 200–204, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2012a. Error mining with suspicion trees: Seeing the forest for the trees. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 2011–2026, Mumbai, India, December. The COLING 2012 Organizing Committee.

Shashi Narayan and Claire Gardent. 2012b. Structure-driven lexicalist generation. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 2027–2042, Mumbai, India, December. The COLING 2012 Organizing Committee.

Shashi Narayan. 2011. RTG based surface realisation from dependency representations. Master's thesis, University of Nancy 2 and University of Malta. Erasmus Mundus Master "Language and Communication Technology".

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, pages 81–106, March.

Benoît Sagot and Éric de la Clergerie. 2006. Error mining in parsing results. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 329–336, Sydney, Australia.

Stuart M. Shieber, Gertjan Van Noord, Fernando C. N. Pereira, and Robert C. Moore. 1990. Semantic-head-driven generation. *Computational Linguistics*, 16.

The XTAG Research Group. 2001. A lexicalised tree adjoining grammar for english. Technical report, Institute for Research in Cognitive Science, University of Pennsylvannia.

Gregor Thurmair. 1990. Parsing for grammar and style checking. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, pages 365–370.

Gertjan van Noord. 2004. Error mining for wide-coverage grammar engineering. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, pages 446–453, Barcelona, Spain.