High–level methodologies for grammar engineering. Introduction to the special issue

Denys Duchier and Yannick Parmentier
Laboratoire d'Informatique Fondamentale d'Orléans,
Université d'Orléans, France

ABSTRACT

Grammar engineering is the task of designing and implementing linguistically motivated electronic descriptions of natural language (so-called grammars). These grammars are expressed within well-defined theoretical frameworks, and offer a fine-grained description of natural language. While grammars were first used to describe syntax, that is to say, the relations between constituents in a sentence, they often go beyond syntax and include semantic information. Grammar engineering provides precise descriptions which can be used for natural language understanding and generation, making these valuable resources for various natural language applications, including textual entailment, dialogue systems, or machine translation. The first attempts at designing large-scale resource grammars were costly because of the complexity of the task (Erbach 1990) and of the number of persons that were needed (see e.g. Doran *et al.* 1997). Advances in the field have led to the development of environments for semi-automatic grammar engineering, borrowing ideas from compilation (grammar engineering is compared with software development) and machine learning. This special issue reports on new trends in the field, where grammar engineering benefits from elaborate high-level methodologies and techniques, dealing with various issues (both theoretical and practical).

Denys Duchier, Yannick Parmentier

1 GRAMMAR ENGINEERING

Grammar engineering, the task of designing and implementing linguistically motivated electronic grammars, has been an active field for decades, following seminal work by Chomsky (1957) on formal languages. The kind of deep structures produced in this field contain rich information, which makes them a valuable resource for various NLP applications, including natural language parsing / generation, textual entailment, or dialogue systems. Among the reasons which make grammar engineering a complex task, one may cite the variety of theoretical frameworks that are used to represent linguistic information, and the intrinsic complexity coming from interactions between rules within large grammars.

In this context, the classical model of hand-crafted grammar has been replaced with more advanced techniques, which we may call semi-automatic grammar production. These techniques vary depending on the target formalism, the target language, or the target linguistic dimensions (e.g. syntax, semantics, morphology, etc.). In the following, we first report on the production of formal grammars (Section 1.1). We then report on the main resource grammars that are available (Section 1.2). We then give a brief overview of the current issues in grammar engineering (Section 1.3). Finally, in Section 2, we summarise the contributions of this special issue, and in Section 3, we conclude about the current status of grammar engineering.

1.1 *Semi-automatic production of formal grammars*

Over the last decades, several approaches to formally describing natural language syntax have been proposed, starting with relatively basic string rewriting systems such as Context-Free Grammar (Chomsky 1956), to continue with more elaborate constraint-based systems such as Head-driven Phrase Structure Grammar (Sag and Pollard 1987). These formal grammars differ in terms of expressive power and computational complexity. While theoretical research on formal grammar addresses the question of what expressive power is needed to describe natural language syntax (and of what computational cost it implies),[1] more practical research is concerned with issues arising from building large grammars.

---

[1] See e.g. Joshi 1985.

Large grammars often exhibit a high structural redundancy, especially when using *lexicalised* formalisms where each grammar rule is associated with at least one lexical item.[2] This high redundancy heavily affects grammar production and maintenance. Indeed, some representation choice applies to many grammatical structures. Should it be modified, a costly revision of the grammar would be required.

In order to facilitate grammar engineering, two main approaches have been considered. The first approach, which we may call knowledge-driven grammar engineering, aims at formally describing the structures belonging to a grammar (which in turn describes natural language syntax). Such a formal description is defined by linguists using a description language.[3] Examples of such description languages include the PATR II (Shieber 1984), the DATR (Evans and Gazdar 1996), and more recently the XMG (Crabbé *et al.* 2013) languages.

Such languages offer a well-defined syntax and semantics to express the relations between grammar structures. These relations are then automatically processed to build a set of structures (i.e., the target grammar). While working with description languages, grammar engineering becomes similar to software engineering. Indeed, both rely on developers working together on a source code, which can be processed to produce some information (e.g. some binary code in one case, or some syntactic structures in the other case).

The second approach to grammar engineering, which we may call data-driven, aims at acquiring the structures belonging to a grammar from annotated corpora (so-called treebanks) (Abeillé 2003). The complexity of grammar engineering is moved from designing grammar rules to designing learning algorithms. Examples of such grammar induction include for instance work by Charniak (1994, Chapter 7), Villavicencio (2002), or Cahill *et al.* (2005). As is the case with statistical approaches in general, grammar learning suffers from the sensitivity to the corpus used to infer the grammar, not mentioning the fact that it requires large annotated corpora which may be lacking when

---

[2] Such lexicalised formalisms are particularly interesting for the lexicon and can be seen as a mapping between a word and its various uses in a sentence, and parsing complexity is reduced since only the grammar rules associated with the input words need to be considered.

[3] In this respect, one may consider that the grammar description itself is a linguistically motivated description of natural language.

working on under-resourced languages. Still, data-driven approaches to grammar learning showed promising results for English, in particular in terms of coverage (Cahill *et al.* 2008).

These two approaches can also be seen as complementary. While knowledge-based methods make it possible to design precision grammars where one can integrate various extra information (e.g. semantic structures), they often hardly scale up so far as describing unrestricted text. On the other hand, while data-driven methods allow to build robust grammars, which can achieve very good results in terms of coverage, these automatically acquired grammars sometimes fail at describing linguistic phenomena which are very infrequent.

1.2   *Resource grammars*

Among the many formalisms which have been proposed to describe natural language syntax, some have been used in practice to develop core or large grammars for a wide range of languages. Formalisms for which there exist available electronic grammars include (by chronological order of publication): Tree Adjoining Grammar (TAG, Joshi *et al.* 1975), Lexical Functional Grammar (LFG, Kaplan and Bresnan 1982), Head-driven Phrase Structure Grammar (HPSG, Sag and Pollard 1987), Combinatory Categorial Grammar (CCG, Steedman 1987), Interaction Grammar (IG, Perrier 2000), or Property Grammar (PG, Blache 2005).

Many efficient description-language-based integrated grammar development environments have been created for these formalisms, such as XLE (Butt *et al.* 1999) for LFG, ALE (Carpenter and Penn 1999), TRALE (Meurers *et al.* 2002) and LKB (Copestake 2002) for HPSG, or DotCCG (Baldridge *et al.* 2007) and GF (Ranta 2011) for CCG. Such environments made it possible to develop large grammars for several languages, see Table 1. Such grammars have been used in practical applications such as machine translation (Lønning and Oepen 2006), textual adventure games (Benotti 2009), or second language learning (Perez-Beltrachini *et al.* 2012).

At the same time, efficient learning algorithms have been developed to induce large grammars from annotated corpora for some of these formalisms, see e.g. Xia (1999) for TAG, Cahill *et al.* (2002) for LFG, Miyao *et al.* (2005) for HPSG, or Hockenmaier and Steedman (2002) for CCG. These automatically learned grammars have been

Table 1: Available electronic grammars (non-exhaustive)

Type	Grammar	Reference
TAG	XTAG (English)	`http://www.cis.upenn.edu/~xtag/`
	XTAG using XMG (English)	`http://homepages.inf.ed.ac.uk/s0896251/XMG-basedXTAG/titlepage.html`
	FrenchTAG	`https://sourcesup.renater.fr/scm/viewvc.php/trunk/METAGRAMMARS/FrenchTAG/?root=xmg`
	GerTT (German)	`http://www.sfs.uni-tuebingen.de/emmy/res-en.html`
LFG	Parallel Grammar (Norwegian, Japanese, etc.)	`http://pargram.b.uib.no/`
	HunGram (Hungarian)	`http://hungram.unideb.hu`
	Urdu ParGram	`http://ling.uni-konstanz.de/pages/home/pargram_urdu`
	POLFIE (Polish)	`http://zil.ipipan.waw.pl/LFG`
HPSG	English Resource Grammar	`http://lingo.stanford.edu/erg.html`
	GG (German Grammar)	`http://gg.dfki.de/`
	JaCY Grammar (Japanese)	`http://jacy.opendfki.de/`
	Korean Resource Grammar	`http://web.khu.ac.kr/~jongbok/projects/krg.html`
	Modern Greek Resource Grammar	`http://www.delph-in.net/mgrg/`
	NorSourceGrammar (Norwegian)	`http://wiki.delph-in.net/moin/NorsourceTop`
	Spanish Resource Grammar	`http://svn.emmtee.net/trunk/upf/srg/`
	Berligram (German), Danish, Chinese, Persian	`https://hpsg.fu-berlin.de/Software/`
CCG	openCCG (English)	`http://www.utcompling.com/wiki/openccg/openccg-grammar-writing`
	Grail (French, Dutch)	`http://www.labri.fr/perso/moot/Corpus/`
	GF (29 languages)	`http://www.grammaticalframework.org/lib/doc/synopsis.html`
IG	FriGram	`http://wikilligramme.loria.fr/doku.php/frig:frig`
PG	FrenchPG	`http://prost.jeanphilippe.free.fr/resources/grammaireGP13.JPP.these.xml`

evaluated on test suites, and often used in practical applications such as semantic construction (Bos *et al.* 2004), dialogue systems (Foster *et al.* 2005), or machine translation (Birch *et al.* 2007). [4]

1.3   *Current issues*

As mentioned above, the field of grammar engineering has been active for several decades. It succeeded in providing the research community with both large resources for a wide range of languages, and techniques for efficient grammar production. By efficient, it is meant that knowledge-based approaches now offer expressive and modular description languages, together with tools for computer-aided grammar design. [5] In the same spirit, data-driven approaches now offer generic algorithms and frameworks which can be applied to the induction of grammars for many formalisms and languages (provided there exist available treebanks for these languages).

Still, the field has a lot more to offer, on-going projects aim at removing existing barriers in grammar engineering, such as *the lack of enhanced grammar development techniques and tools*, which would facilitate grammar debugging, grammar evaluation, or collaborative grammar design. Attempts at providing such techniques include work by Gardent and Kruszewski (2012) on debugging and by Hoetmer (2005) and Sygal and Wintner (2011) on grammar design.

Another current issue in grammar engineering concerns *parsing efficiency*. Indeed parsing complexity depends not only on the length of the input sentence, but also on the grammar size. In order to parse sentences using large grammars, several options have been considered, including the on-line (symbolic or probabilistic) selection of a sub-part of the grammar (Zhang *et al.* 2009; Gardent *et al.* 2014), [6] or parsing using factorised grammars (Carroll *et al.* 2000; Villemonte De La Clergerie 2010).

Other challenges include *multilingual and cross-framework grammar engineering*. While there exist several projects aiming at building

---

[4] Some of these automatically acquired grammars are available on-line, see e.g. `http://web.engr.illinois.edu/~juliahmr/CCGlexicon/index.html` for CCG, or `http://lfg-demo.computing.dcu.ie/lfgparser.html` for LFG.

[5] See e.g. the Matrix for HPSG (Bender *et al.* 2010).

[6] Following seminal work by Bangalore and Joshi (1999), this selection is often called supertagging.

parallel grammars (see e.g. Butt *et al.* 2002; Flickinger *et al.* 2012), cross-framework grammar engineering did not (yet) achieve the same results. One may cite seminal work by Clément and Kinyon (2003) on the description of parallel TAG-LFG from a common abstract description (called metagrammar), or more recent work by Crabbé *et al.* (2014) on the design of a constraint-based description language which could be applied to the description of grammars belonging to distinct formalisms. In the latter, the authors show how to enrich the description language to support several target formalisms, while in the former the authors show how to project a common description to several target formalisms (the metagrammar could be seen to some extent as a universal grammar). Without going as far as designing a universal grammar, *grammar reusability* (i.e., sharing information between grammars) remains an important challenge.

Another interesting topic concerns *grammar interfaces*. One of the motivations behind grammar engineering is the possibility to build rich semantic representations. The definition and implementation of a syntax / semantics interface within large grammars is an active field (see e.g. Gardent 2008; Kallmeyer and Osswald 2013), for both theoretical (definition / selection of an adequate semantic formalism) and technical (limited grammar readability and extensibility) reasons.

Describing under-resourced languages is also an active field within the grammar engineering community. The objective is twofold. Grammar engineering can help to (i) better understand e.g. minority languages (by implementing linguistic theories and checking how this implementation compares with field data), and also (ii) provide electronic grammars (which would make it possible to develop NLP applications for these languages, and/or build core treebanks, which could in turn be beneficial to grammar engineering). Recent knowledge-based attempts at creating linguistic resources for under-resourced languages include work by Bender (2008) and Duchier *et al.* (2012).

Last but not least, in order to improve *grammar coverage*, novel ideas are needed. As mentioned above, a first step towards a better grammar coverage was to automatically learn the grammar from annotated corpora. In order to get grammars with a better coverage while keeping a high precision, hybrid techniques involving both knowledge-based and data-driven methods are needed. Seminal work by Baldwin *et al.* (2005) expresses the same concerns.

In order to improve grammar coverage, one major issue needs to be addressed, namely, *Multi-Word Expressions*. Such expressions are often ignored when designing core grammars, while they frequently appear in unrestricted text. Work on MWE detection for enhancing parsing with HPSG has been done by Zhang *et al.* (2006), where authors use parsing error mining techniques to detect whether unknown words belong to some MWE which is in turn included in the lexicon.[7] Further work in this field is needed to improve grammar precision and coverage.[8]

2 CONTRIBUTIONS TO THIS SPECIAL ISSUE

This special issue contains contributions dealing with several aspects of grammar engineering, namely description languages (Clément *et al.*), grammar extraction (Le-Hong *et al.*), syntax / semantics interface (Lichte and Petitjean), grammar coverage (Moot), multilingual grammars (Müller), and grammar development and maintenance (Perrier and Guillaume).

Clément, Kirman, and Salvati present a logic-based grammar description formalism. They use this formalism to describe both mildly context-sensitive grammars and their semantic interpretation. As an illustration, this formalism is applied to the (syntactic and semantic) description of several linguistic phenomena related to extraction in Dutch, English, and German.

Le-Hong, Roussanaly and Nguyen present the development of a linguistic resource for Vietnamese using the TAG formalism. The authors first show how to semi-automatically extract such a grammar from a treebank of Vietnamese. In a second step, they use this grammar for deep parsing. In particular, they present a complete pipeline for parsing Vietnamese sentences to produce constituent and dependency structures.

---

[7] As is the case in lexicalised formalisms, the term *lexicon* is used here to refer to the grammar entries.

[8] Better support of MWEs within lexicons, grammars, and applications is among the topics of the current PARSEME international initiative, EU COST Action IC1207, see `http://www.parseme.eu`.

Lichte and Petitjean present an extension of the XMG description language with a new linguistic dimension based on semantic frames. In their approach, the authors aim at offering a description language, which can be used to express various constraints on types. They apply this formalism to the definition of a syntax / semantics interface within an English TAG.

Moot reports on the development of a type-logical treebank for French, and its use for wide-coverage syntactic and semantic parsing. This article contains information about the various tasks involved in the development of a competitive type-logical parser for French using an automatically-extracted broad-coverage type-logical grammar.

Müller presents the CoreGram project, which aims at providing HPSG grammars for various typologically distinct languages. In this approach, a multilingual grammar is used to represent a common core shared by these languages. This article gives theoretical linguistic motivations behind multilingual grammars, along with theoretical grammar development concepts, and information about the concrete implementation of the corresponding HPSG grammars.

Perrier and Guillaume present FriGram, a broad-coverage French IG, which relies on a modular architecture and can be interfaced with various lexicons. This article also addresses grammar design and maintenance issues by presenting grammar-consistency principles which are implemented within FriGram. The authors also report on the current status of the grammar (coverage, comparison with other resource grammars for French, evaluation).

3 CONCLUSION

In this introduction, we gave an overview of past and recent advances in the field of grammar engineering. We presented the main approaches for semi-automatic grammar production, namely knowledge-based approaches, which rely on linguistically motivated descriptions of formal grammar designed by experts, and data-driven approaches, which rely on robust broad-coverage grammars extracted from large annotated corpora.

We also reported on existing available resource grammars for various languages and grammar formalisms, and summarised current issues in grammar engineering. These issues include the lack of tech-

niques and tools for easier grammar extension and maintenance (e.g. debugging facilities), a sometimes low parsing efficiency when dealing with large grammars, the limited coverage of hand-crafted grammars (especially regarding multi-word expressions), the difficulties to interface syntax with other linguistic dimensions, and a weak reusability between grammars belonging to different formalisms or describing different languages.

We finally gave a brief overview of the contributions to this special issue, which cover both knowledge-based and data-driven approaches, along with several grammar formalisms (namely CCG, HPSG, TAG, IG), several linguistic dimensions (syntax and semantics), and several languages (including English, Dutch, German, French, Danish, Persian, etc.).

ACKNOWLEDGEMENTS

REFERENCES

Anne Abeillé (2003), *Treebanks: Building and Using Parsed Corpora*, Text, Speech and Language Technology, Springer.

Jason Baldridge, Sudipta Chatterjee, Alexis Palmer, and Ben Wing (2007), DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG, in Tracy Holloway King and Emily M. Bender, editors, *Proceedings of the GEAF07 Workshop*, pp. 5–25, Center for the Study of Language and Information (CSLI), Stanford, California, http://csli-publications.stanford.edu/GEAF/2007/geaf07-toc.html.

Timothy Baldwin, John Beavers, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen (2005), Beauty and the Beast: What running a broad-coverage precision grammar over the BNC taught us about the grammar

— and the corpus, in Stephan KEPSER and Marga REIS, editors, *Linguistic Evidence: Empirical, Theoretical, and Computational Perspectives*, pp. 49–70, Mouton de Gruyter, Berlin.

Srinivas BANGALORE and Aravind K. JOSHI (1999), Supertagging: An Approach to Almost Parsing, *Computational Linguistics*, 25(2):237–262.

Emily M. BENDER (2008), Evaluating a Crosslinguistic Grammar Resource: A Case Study of Wambaya, in *Proceedings of ACL-08: HLT*, pp. 977–985, Association for Computational Linguistics, Columbus, Ohio.

Emily M. BENDER, Scott DRELLISHAK, Antske FOKKENS, Michael Wayne GOODMAN, Daniel P. MILLS, Laurie POULSON, and Safiyyah SALEEM (2010), Grammar Prototyping and Testing with the LinGO Grammar Matrix Customization System, in *Proceedings of the ACL 2010 System Demonstrations*, pp. 1–6, Association for Computational Linguistics, Uppsala, Sweden.

Luciana BENOTTI (2009), Frolog: an Accommodating Text-Adventure Game, in *Proceedings of the Demonstrations Session at EACL 2009*, pp. 1–4, Association for Computational Linguistics, Athens, Greece.

Alexandra BIRCH, Miles OSBORNE, and Philipp KOEHN (2007), CCG Supertags in Factored Statistical Machine Translation, in *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 9–16, Association for Computational Linguistics, Prague, Czech Republic.

Philippe BLACHE (2005), Property Grammars: A Fully Constraint-Based Theory, in Henning CHRISTIANSEN, Peter Rossen SKADHAUGE, and Jørgen VILLADSEN, editors, *Constraint Solving and Language Processing*, volume 3438 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Berlin Heidelberg.

Johan BOS, Stephen CLARK, Mark STEEDMAN, James R. CURRAN, and Julia HOCKENMAIER (2004), Wide-coverage Semantic Representations from a CCG Parser, in *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pp. 1240–1246, Association for Computational Linguistics, Stroudsburg, PA, USA.

Miriam BUTT, Helge DYVIK, Tracy Holloway KING, Hiroshi MASUICHI, and Christian ROHRER (2002), The Parallel Grammar Project, in *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pp. 1–7, Taipei, Taiwan.

Miriam BUTT, Tracy H. KING, Marma-Eugenia NIÑO, and Frédérique SEGOND (1999), *A Grammar Writer's Cookbook*, Center for the Study of Language and Information (CSLI), Stanford, California.

Aoife CAHILL, Michael BURKE, Ruth O'DONOVAN, Stefan RIEZLER, Josef VAN GENABITH, and Andy WAY (2008), Wide-Coverage Deep Statistical Parsing Using Automatic Dependency Structure Annotation, *Computational Linguistics*, 34(1):81–124.

Aoife CAHILL, Martin FORST, Michael BURKE, Mairead MCCARTHY, Ruth O'DONOVAN, Christian ROHRER, Josef VAN GENABITH, and Andy WAY (2005), Treebank-Based Acquisition of Multilingual Unification Grammar Resources, *Journal of Research on Language and Computation; Special Issue on "Shared Representations in Multilingual Grammar Engineering"*, 3(2):247–279.

Aoife CAHILL, Mairéad MCCARTHY, Josef VAN GENABITH, and Andy WAY (2002), Parsing with PCFGs and automatic f-structure annotation, in *Proceedings of the 7th International Conference on LFG*, pp. 76–95, Center for the Study of Language and Information (CSLI), Palo Alto, California.

Bob CARPENTER and Gerald PENN (1999), ALE 3.2 User's Guide, Technical Memo CMU-LTI-99-MEMO, Carnegie Mellon Language Technologies Institute.

John CARROLL, Nicholas NICOLOV, O. SHAUMYAN, M. SMETS, and D. WEIR (2000), Engineering a wide-coverage lexicalized grammar, in *Proceedings of the Fifth International Workshop on Tree Adjoining Grammars and Related Frameworks*, pp. 55–60, Paris, France.

Eugene CHARNIAK (1994), *Statistical Language Learning*, MIT Press, Cambridge, Massachusetts.

N. CHOMSKY (1956), Three models for the description of language, *Information Theory, IEEE Transactions on*, 2(3):113–124.

Noam CHOMSKY (1957), *Syntactic Structures*, Mouton, The Hague.

Lionel CLÉMENT and Alexandra KINYON (2003), Generating Parallel Multilingual LFG-TAG Grammars from a MetaGrammar, in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 184–191, Sapporo, Japan.

Ann COPESTAKE (2002), *Implementing Typed Feature Structure Grammars*, Center for the Study of Language and Information (CSLI), Stanford, California.

Benoît CRABBÉ, Denys DUCHIER, Claire GARDENT, Joseph LE ROUX, and Yannick PARMENTIER (2013), XMG : eXtensible MetaGrammar, *Computational Linguistics*, 39(3):591–629.

Benoît CRABBÉ, Denys DUCHIER, Yannick PARMENTIER, and Simon PETITJEAN (2014), Constraint-driven Grammar Description, in Philippe BLACHE, Henning CHRISTIANSEN, Verónica DAHL, Denys DUCHIER, and Jørgen VILLADSEN, editors, *Constraints and Language*, pp. 93–121, Cambridge Scholar Publishing.

Christine DORAN, Beth HOCKEY, Philip HOPELY, Joseph ROSENZWEIG, Anoop SARKAR, Srinivas BENGALORE, Fei XIA, Alexis NASR, and Owen RAMBOW (1997), Maintaining the Forest and Burning out the Underbrush in XTAG, in *Proceedings of the ACL Workshop on Computational Environments for Grammar Development and Language Engineering (ENVGRAM)*, pp. 30–37, Madrid, Spain.

Denys DUCHIER, Brunelle MAGNANA EKOUKOU, Yannick PARMENTIER, Simon PETITJEAN, and Emmanuel SCHANG (2012), Describing Morphologically-rich Languages using Metagrammars: a Look at Verbs in Ikota, in *Workshop on "Language technology for normalisation of less-resourced languages", 8th SALTMIL Workshop on Minority Languages and the 4th workshop on African Language Technology*, pp. 55–60, Istanbul, Turkey, http://aflat.org/files/saltmil8-aflat2012.pdf.

Gregor ERBACH (1990), Grammar Engineering: Problems And Prospects, report on the Saarbrücken Grammar Engineering Workshop.

Roger EVANS and Gerald GAZDAR (1996), DATR: A Language for Lexical Knowledge Representation, *Computational Linguistics*, 22(2):167–213.

Dan FLICKINGER, Valia KORDONI, Yi ZHANG, Ant BRANCO, K. SIMOV, Petya OSENOVA, Catarina CARVALHEIRO, Francisco COSTA, and S CASTRO (2012), ParDeepBank : multiple parallel deep treebanking, in *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT11)*, pp. 97–108, Lisbon, Portugal.

Mary E. FOSTER, Michael WHITE, Andrea SETZER, and Roberta CATIZONE (2005), Multimodal Generation in the COMIC Dialogue System, in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 45–48, Association for Computational Linguistics, Ann Arbor, Michigan.

Claire GARDENT (2008), Integrating a unification-based semantics in a large scale Lexicalised Tree Adjoininig Grammar for French, in *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pp. 249–256, Manchester, United Kingdom.

Claire GARDENT and German KRUSZEWSKI (2012), Generation for Grammar Engineering, in *INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference*, pp. 31–39, Association for Computational Linguistics, Utica, Illinois.

Claire GARDENT, Yannick PARMENTIER, Guy PERRIER, and Sylvain SCHMITZ (2014), Lexical Disambiguation in LTAG using Left Context, in Zygmunt VETULANI and Joseph MARIANI, editors, *Human Language Technology. Challenges for Computer Science and Linguistics. 5th Language and Technology Conference, LTC 2011, Poznan, Poland, November 25-27, 2011, Revised Selected Papers*, volume 8387, pp. 67–79, Springer.

Julia HOCKENMAIER and Mark STEEDMAN (2002), Generative Models for Statistical Parsing with Combinatory Categorial Grammar, in *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 335–342, Association for Computational Linguistics, Philadelphia, Pennsylvania.

Kenneth HOETMER (2005), *Higher-Order Types for Grammar Engineering*, Master's thesis, University of Toronto, Department of Computer Science, http://www.cs.toronto.edu/~hoetmer/hoetmerthesis.pdf.

Aravind K. JOSHI (1985), Tree Adjoining Grammars: How much Context - sensitivity is Required to Provide Reasonable Structural Descriptions?, in David R. DOWTY, Lauri KARTTUNEN, and Arnold ZWICKY, editors, *Natural Language Parsing*, pp. 206–250, Cambridge University Press, Cambridge.

Aravind K. JOSHI, Leon S. LEVY, and Masako TAKAHASHI (1975), Tree Adjunct Grammars, *Journal of Computer and System Sciences*, 10(1):136–163.

Laura KALLMEYER and Rainer OSSWALD (2013), Syntax-Driven Semantic Frame Composition in Lexicalized Tree Adjoining Grammars, *Journal of Language Modelling*, 1(2):267–330.

Ronald M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A Formal System for Grammatical Representations, in *The Mental Representation of Grammatical Relations*, pp. 173–281, MIT Press.

Jan Tore LØNNING and Stephan OEPEN (2006), Re-Usable Tools for Precision Machine Translation, in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 53–56, Association for Computational Linguistics, Sydney, Australia.

W. Detmar MEURERS, Gerald PENN, and Frank RICHTER (2002), A Web-based Instructional Platform for Contraint-Based Grammar Formalisms and Parsing, in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 19–26, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA.

Yusuke MIYAO, Takashi NINOMIYA, and Jun'ichi TSUJII (2005), Corpus-Oriented Grammar Development for Acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank, in Keh-Yih SU, Jun'ichi TSUJII, Jong-Hyeok LEE, and OiYee KWONG, editors, *Natural Language Processing – IJCNLP 2004*, volume 3248 of *Lecture Notes in Computer Science*, pp. 684–693, Springer, Berlin, Heidelberg.

Laura PEREZ-BELTRACHINI, Claire GARDENT, and German KRUSZEWSKI (2012), Generating Grammar Exercises, in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pp. 147–156, Association for Computational Linguistics, Montréal, Canada.

Guy PERRIER (2000), Interaction Grammars, in *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pp. 600–606, Saarbruecken, Germany.

Aarne RANTA (2011), *Grammatical Framework: Programming with Multilingual Grammars*, Center for the Study of Language and Information (CSLI), Stanford, California.

Ivan A. SAG and Carl J. POLLARD (1987), Head-Driven Phrase Structure Grammar: An Informal Synopsis, CSLI Report 87–79, Stanford University.

Stuart M. SHIEBER (1984), The Design of a Computer Language for Linguistic Information, in *Proceedings of the Tenth International Conference on Computational Linguistics*, pp. 362–366, Stanford, California.

Mark STEEDMAN (1987), Combinatory grammars and parasitic gaps, *Natural Language & Linguistic Theory*, 5(3):403–439.

Yael SYGAL and Shuly WINTNER (2011), Towards Modular Development of Typed Unification Grammars, *Computational Linguistics*, 37(1):29–74.

Aline VILLAVICENCIO (2002), The acquisition of a unification-based generalised categorial grammar, Technical Report UCAM-CL-TR-533, Number 533, Computer Laboratory, University of Cambridge.

Éric VILLEMONTE DE LA CLERGERIE (2010), Building factorized TAGs with meta-grammars, in *The 10th International Conference on Tree Adjoining Grammars and Related Formalisms - TAG + 10*, pp. 111–118, New Haven, Connecticut, https://hal.inria.fr/inria-00551974.

Fei XIA (1999), Extracting Tree Adjoining Grammars from bracketed corpora, *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, pp. 398–403.

Yao-zhong ZHANG, Takuya MATSUZAKI, and Jun'ichi TSUJII (2009), HPSG Supertagging: A Sequence Labeling View, in *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pp. 210–213, Association for Computational Linguistics, Paris, France.

Yi ZHANG, Valia KORDONI, Aline VILLAVICENCIO, and Marco IDIART (2006), Automated Multiword Expression Prediction for Grammar Engineering, in *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pp. 36–44, Association for Computational Linguistics, Sydney, Australia.