# IT Research Methods
# INFO5993

## Exemplar of research proposal

*(permission to distribute anonymously has been granted by the author)*

# Contents

# Chapter 1

# Introduction

## 1.1 Introduction

Parsing is the recovery of the hierarchical structure from the linear medium of written text. The accurate and efficient recovery of this structure, which describes the relations which hold between the constituents of a sentence, is a key step in any natural language processing application which deals with the semantic content of text. The research we undertake deals broadly with a parsing formalism known as Combinatory Categorial Grammar (hereafter CCG), and in particular, the implementation and evaluation of an extension to the CCG, known as Multi-Modal Combinatory Categorial Grammar (MMCCG).

This chapter is divided broadly into three sections: in this first section we will discuss the aspects of CCG which make it an attractive formalism for the practical problem of wide-coverage parsing, and identify the weaknesses and issues with the basic formalism which motivate the development of subsequent refinements, including the focus of our present work, MMCCG. Having demonstrated the necessity of MMCCG, we will outline the claims made by existing research into MMCCG, and in doing so identify the central thesis of our own work. Furthermore, we will explain the role of our present research in the context of existing work in the field of natural language processing (NLP), and characterise the contributions by which our work will enrich the state of NLP research.

A second section, a research plan, will outline the conceptual stages of our research, together with the dates by which we intend to have achieved each milestone. We will also identify contingencies to be considered if the implementation of the research plan begins to deviate from this schedule, possible causes for such deviations, and the measures and means to mitigate against them.

We will conclude by outlining the approaches and research methods by which we will validate the claims put forward in the first section, justifying the particular research methods we have chosen to arrive at our central thesis. We include a discussion of the evaluation techniques by which we intend to measure the success of our work.

## 1.2 Combinatory Categorial Grammar

The *Combinatory Categorial Grammar* (hereafter CCG) is a highly lexicalised grammatical formalism which unifies syntax and semantics through the act of parsing, achieving this by providing a direct mapping between syntactic structures and the underlying logical (semantic) structure. The formalism itself is an application and extension of the framework of combinatory logic as established by Curry in [16], in which *combinators*, as higher-order functions, transform and compose other functions. The CCG, as developed and described by Steedman in [39], interprets entries in a lexicon as types ('categories'), together with a set of combinators which act upon them to combine them into syntactic units. CCG itself is best understood as a small set of combinatory rules, or type schemata on the categories of words, which unify them into larger units.

The goal of any practical parsing formalism is to provide a means of encoding syntax (the assembly of words into larger units known as constituents), with the implicit goals of enabling the efficient, robust and accurate retrieval of syntactic and semantic structure from natural text.

Purely as a formalism, CCG consists of two components: a method of embedding the description of the syntax of a language into its lexicon, as well as a set of combinatory rules governing the unification of constituents to form more complex structures. CCG is characterised as a *lexicalised* parsing formalism, meaning that the responsibility of specifying syntax is placed in the lexicon, in contrast to formalisms such as the familiar context-free grammar, in which the grammar is a separate component to the lexicon. Lexicalised formalisms enjoy several practical advantages: the computational tasks of disambiguating syntax and disambiguating word senses can be achieved in a single step, and the flexibility afforded by the ability to specify syntax on a per-word basis leads to parsers for which it is easier to specify fine-grained distinctions in grammaticality.

*Combinatory rules* are schemes for transforming a single category into another (unary rules), or combining two categories (binary rules). The successive application of these rules is what induces syntactic structure in the linear order of text. The name hints at the fact that for each rule, there is a corresponding higher-order function (a combinator) which combines the logical representations of its operands, represented as functions in the untyped lambda calculus. This correspondence renders CCG capable of producing a semantic representation in parallel to the parsing act. The ability to parse a query, and in doing so compositionally build the meaning of that query, gives it additional appeal when considered in light of the real-world natural language understanding task of question answering [6].

CCG has a number of distinctions which make it an attractive target for real-world parsing systems: firstly, it is both simple to characterise intuitively as the combination of words through the application of easily-specified combinatory rules, as well as convenient to characterise formally, due to its foundation in the discipline of type theory and formal logic. Secondly, the combination of syntactic and semantic information embedded in the simple concept of 'category' allows for the efficient capture of a wide range of linguistic structures. Third, characterising categories as types renders CCG capable of producing a semantic representation in parallel to the parsing act. The ability to parse a query, and in doing so compositionally build the meaning of that query, gives it additional appeal when considered in light of the real-world natural language understanding task of question answering [6].

## 1.3 Issues with the Combinatory Categorial Grammar

In considering how to apply the pure formalism to parsing various structures found in the languages of the world, Steedman in [39] demonstrates that it although we would otherwise like it to be, pure CCG alone is often insufficient to completely specify a language. In particular, in parsing a given language, it may not be desirable to allow the use of every combinatory rule given by the pure CCG formalism: as an example, one of the combinatory rules, namely *forward crossed composition*, is necessary to capture the free variation in the linear order of certain constituents possessed by languages such as Latin or Japanese, but given the rigidity of verb arguments in English, licensing such a combinatory rule would clearly lead to *overgeneration*, that is, the acceptance of ungrammatical forms due to the unchecked application of particular CCG rules. This single point is one of the main motivations for the multi-modal CCG, as introduced later. Steedman's solution to the overgeneration problem is to permit combinatory rules to be de-licensed for particular languages, or indeed particular structures. The problem with this approach is that in doing so, we have reduced the lexicality of CCG: parsing with the CCG now depends on information outside the lexicon, and we have lost some of the generality which made the CCG an attractive formalism.

The CCG parser which is the starting point for our modifications addresses the restricted applicability of combinatory rules by hard-coding them into the parser, which is an unsatisfactory approach for two reasons. Firstly, such restrictions are language-specific, and extending the parser would require the modification and recompilation of the parser source code. The CCG is intended to embody an invariant parser component, but the formalism does not allow us to encode restrictions in the applicability of combinatory rules in the lexicon. In practical implementations of CCG, this means that restrictions must either be hard-coded into the parser, or else read in as a resource separate to the lexicon. Secondly, embedding the restrictions in the parser incurs considerable overhead, as the parser must determine whether it is parsing an instance of a structure which disallows the application of a given combinatory rule. This must be done for each parsing step, and for each encoded exception. Hence, the promise of MMCCG is to eliminate both shortcomings: the restrictions are moved out of the parser, and the parser no longer has to consider where a rule *cannot* be used, as the categories in MMCCG now encode precisely the set of rules which can.

As mentioned before, the above observation relating to overgeneration is the key point behind Baldridge's work in [2] introducing the *multi-modal combinatory categorial grammar* (hereafter MMCCG), the foundation for our own work. Overgeneration in the pure CCG necessitates the selective licensing of combinatory rules, but the issue is that in doing so, we move some of the specificational responsibility out of the lexicon. A natural way to address this divergence is to identify those restrictions that must be captured to prevent overgeneration, and encode them in the lexicon. Baldridge achieves this by requiring each category, and hence each lexical item, to specify the set of combinatory rules in which it can validly participate. Consequently if applying a given rule would lead to overgeneration in a given structure, the categories of the participants in that structure would bar that rule from ever being considered. Its emphasis on constructing the extension of MMCCG from the underlying Categorial Type Logic with a reduction from CCG couches it in the tradition of formal categorial logic, while Baldridge derives insights from the discipline of generative linguistics to provide an account for constraints on the space of possible categories. The additional specificational power yielded by MMCCG is demonstrated in

[2] through a multi-modal CCG-informed analysis of *extraction asymmetries* (restrictions on the movement of constituents) in the natural languages Tagalog and Toba Batak. Baldridge also describes his proof-of-concept implementation of a MMCCG parser, known as Grok, as a demonstration of the practicality of MMCCG. We intend to evaluate not just the theoretical practicality of MMCCG, but its value as applied to wide-coverage parsing, by incorporating MMCCG into an existing parser capable of parsing general text.

**Thesis 1.** *Multi-Modal Combinatory Categorial Grammar yields theoretical benefits, including increased generality, the ability to specify finer-grained combinatory constraints, and reduced parser ambiguity, leading to practical benefits including increased parser efficiency, more accurate analyses for a range of grammatical constructions, and the ability to remove language-dependent constraints from the parser.*

The primacy of the lexicon is emphasised all the more in MMCCG, since the goal is to restore the parser to a set of invariant rules, returning the responsibility of specifying applicational restrictions to the lexicon. Therefore it becomes even more important to develop a MMCCG lexicon which accurately captures those restrictions which are required to prevent overgeneration. Although Baldridge discusses an approach given in [24, 26] of deriving standard CCG categories from a Government and Binding analysis, no suggestions are given as to deriving the modes (the innovation introduced by MMCCG). These modes must be known *a priori* as they are part of the lexicon, so this is an issue to be addressed by any implementation of the MMCCG with the goal of wide-coverage parsing, as there exists no MMCCG corpus corresponding to Hockenmaier's CCGbank, a CCG corpus derived automatically from the Penn Treebank, a dataset of 2.8 million parsed English sentences. Our proposed contribution to the field is to develop a technique for inducing a mode-annotated corpus, building upon CCGbank. Such a corpus has two benefits: firstly, it would enable us to quantitatively explore the claim made in [2] that MMCCG could yield benefits in parsing efficiency: the applicational restrictions MMCCG allows us to specify may prevent the consideration of ungrammatical parses, leading to a reduction in parsing ambiguity. Secondly, the availability of a MMCCG corpus would fulfill a role analogous to the one CCGbank plays in CCG research, enabling future research to build upon the refinements introduced by MMCCG.

In the course of our work, we will develop the means for extracting a MMCCG corpus from Hockenmaier's CCGbank, and apply this resource to the parser we will develop in our investigation of Thesis 1. We will subsequently demonstrate that MMCCG also greatly simplifies the internals of a CCG parser by shifting specificational responsibility out of the parser component.

**Thesis 2.** *It is possible to induce a* MMCCG *corpus by largely automatic means from an existing corpus.*

**Thesis 3.** MMCCG *simplifies the implementation of a parser component relative to* CCG*, and enables in conjunction with the wide-coverage* MMCCG *corpus developed in verifying Thesis 2 an efficient and general solution to the wide-coverage parsing problem.*

The proof-of-concept implementation of a MMCCG parser in Grok is posed as a verification of the ability of MMCCG to capture the structures Baldridge discusses, as well as an informal demonstration that practical MMCCG parsing is possible with existing techniques. Baldridge constructs by hand small-scale lexicons for English, Dutch, Tagalog and Turkish, with a test suite of 191 sentences in total. Since the size of the test suite is small, Baldridge is able to evaluate the parser output by inspection, and evaluation methods for MMCCG are neither developed nor introduced. A disadvantage resulting from the absence of a formal evaluation is the inability to validate the attractiveness of MMCCG over the pure CCG, or indeed other formalisms of similar generative power.

As we will be unable to evaluate the results of our own general-purpose MMCCG parser by inspection, it is clear that the selection of evaluation metrics appropriate to our parsing task is a concern we must address in our own work. A consequence of our own work will be the validation of hypotheses in [2] relating to improvements in parsing efficiency and discriminative power, along with the potential to compare MMCCG to CCG and other parsing formalisms.

**Thesis 4.** *Once we have a wide-coverage parser trained on an* MMCCG *corpus, we will develop the evaluation framework enabling it to be compared to parsers of similar power, such as the unmodified* CCG. *This evaluation will allow us to evaluate the suitability of* MMCCG *as the underlying formalism of a wide-coverage parser, and enable us to discuss whether the theoretical refinements presented by* MMCCG *correspond to practical benefits.*

## 1.4 Claims of the thesis

Our proposed research is the first application of MMCCG to a wide-coverage parser, the first time a wide-coverage MMCCG lexicon has been generated, and the first evaluation of a MMCCG parser on the wide-coverage parsing task. The resulting parser allows for the lexicon writer to specify restrictions on the space of combinatory operations on a word-by-word basis, reducing ambiguity and increasing parser efficiency. As the formalism of CCG does not allow for these constraints to be encoded in the lexicon, existing parsers based on CCG either hard-code these constraints, or embed them in a resource separate to the lexicon. MMCCG unifies these constraints into each lexical item, and allows finer-grained control over their specification.

The second contribution is the development of automatic or semi-automatic techniques for obtaining an MMCCG corpus, a prerequisite for our evaluation of the practicality of parsers based on the formalism. We will be performing entirely novel research in the development of such methods, and consequently the production of such a corpus. A product of our research will be a wide-coverage MMCCG corpus for the English language, derived from an existing corpus known as CCGbank ([21]).

Just as Hockenmaier's development of the CCGbank enabled the successful application of CCG to wide-coverage parsing as investigated by [12], we believe that our development of a MMCCG corpus will be a key contribution to the advancement of CCG research in general, and a foundation for further research into MMCCG.

## 1.5 Methodology for the validation of claims

The 'success' of a corpus is determined by how well it covers the raw text, so we intend to evaluate lexicon induction by how often it yields incorrect MMCCG derivations for a given CCGbank derivation, or fails to account for the corresponding derivation at all. We will perform an evaluation of the corpus derivation process by selecting elements from the source corpus to form an evaluation set. [21] contains a similar account of structures which CCGbank does or does not account for; we intend to consider each of these phenomena (eg negative inversion, clefting), manually annotating each of these with its correct MMCCG analysis. In particular, we will also evaluate the ability of the corpus in accurately capturing instances of extraction asymmetries [1, 2, 14], the correct treatment of which is one of the key arguments in [2] for the necessity of the MMCCG. The coverage of the corpus is defined as a simple proportion of correct analyses to the size of the evaluation set.

The second evaluation task is to determine the value of MMCCG as a general-purpose parsing formalism over the pure CCG and other similar formalisms. We consider a number of attributes to be desirable in a wide-coverage parser:

- The ability to process text efficiently with respect to time and memory
- The ability to robustly provide accurate parses for a broad spectrum of text
- Ease of supporting new languages, while maintaining a clean division between the lexicon (grammar) and the parser

To evaluate our parser, we will employ a *dependency-sensitive* metric as introduced by Clark in [13], which compares the dependency relations captured by a candidate parse to those encoded in a gold standard. Intuitively, such a metric favours semantic correspondence over structural correspondence, since multiple semantically equivalent CCG derivations may exist for a given text. We believe that performing our evaluation with respect to such a metric has several benefits: as argued in the same paper, dependency-sensitive metrics are a better determinant of correspondence to a gold standard for the CCG parsing problem than the PARSEVAL metric, a structural metric commonly used in the literature for parser evaluation tasks. Furthermore, employing a dependency-sensitive evaluation allows us to directly compare the values computed for the output of our modified parser against the evaluation performed for the unmodified C&C parser ([12]). We will divide the MMCCG corpus derived in the previous stage of our research into three sets: a training set, constituting the majority of the data, a tuning set, a small set to be used once prior to final testing, and the test set, from which we will gather and compute the evaluation metrics we are about to describe.

Following the convention established in [13], and followed in the paper presenting C&C in [12], we will compute the precision, recall and $F$-score with respect to the category of each captured dependency relation, comparing these statistics to those obtained from the unmodified C&C parser on the corresponding sections of the CCGbank. We intend to conduct significance tests to validate any observed difference between the ability of our MMCCG-based parser and the unmodified parser in accurately capturing dependency relations.

The ability of our modified parser to capture a range of syntactic phenomena is difficult to quantify automatically. However, we propose a similar approach to that undertaken by [2] in implementing his proof-of-concept MMCCG parser on top of the Grok CCG parsing system. Baldridge maintains a small test corpus built by hand containing instances of several structures which MMCCG is expected to capture. This serves two main purposes: firstly, it enables regression testing during parser development, and secondly, excerpts from the corpus can be referenced in an evaluation to demonstrate the ability of the parser to process a range of syntactic structures. It is likely that we will develop this small corpus incrementally during both the corpus extraction and the parser development phases of our research.

# Chapter 2

# Literature review

## 2.1 Introduction

The ability to accurately and efficiently parse text is the foundation of any system built for natural language understanding. In improving and extending a wide-coverage parser already integrated into a *question answering system* for interpreting natural-language queries, the work we propose will fulfill three key goals. The first goal is to extend the parser to accommodate a refinement of its parsing formalism, with the effect of increasing its generality, parsing speed and discriminative power. The second goal is to develop the techniques to acquire a *corpus* or body of text with which to train the augmented system, and to apply these methods in transforming an existing corpus. The third goal is to validate the refined formalism by selecting appropriate evaluation metrics, and surveying the parsing performance of the updated parser (the product of our first goal) as trained on the newly induced corpus (the product of the second).

This document is a critical consideration of the literature relevant to the development and implementation of the *Multi-modal Combinatory Categorial Grammar*. We will identify the major considerations pertinent to researchers in the field, survey the contributions and methodologies of selected works, and discuss the ways in which they relate to our own research topic.

## 2.2 Overview of the field

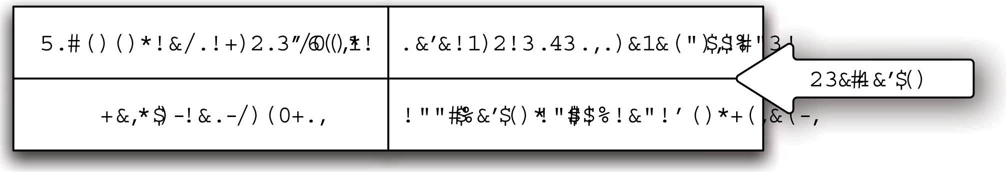

Figure 2.1: Five centres of CCG research

The current state of CCG research encompasses a number of research questions:

- The application of CCG analyses to questions in the study of language, including semantics and information structure [1, 39, 40]: *How can we use* CCG *to shed light on how humans process and understand text, and conversely how do we capture those distinctions through parsing?*

- Augmentations and refinements to the CCG formalism [2, 28, 29, 39]: *how can we improve* CCG*'s ability to discriminate between grammatical and ungrammatical sentences?*

- The representation and acquisition of critical parsing resources such as lexicons or grammars [17, 21, 22, 27, 41]: *how can we develop or improve resources to better inform the parsing process?*

- Methods for the efficient processing of text through CCG (encompassing tagging, parsing and parse selection), and the implementation issues inherent in designing a CCG parser [11, 12, 15, 18, 21]: *How can we build a* CCG *parser for robust, efficient and accurate text processing?*

- Meta-research into evaluating aspects of CCG systems [9, 13]: inadequacies and trade-offs inherent in existing evaluation methodologies. *How should we determine the 'goodness' of a* CCG *system, and compare them to parsers built on* CCG *as well as those based on other formalisms?*

Three of these areas are of particular importance to our research task: refining the CCG formalism, acquiring lexicons, and evaluating a CCG system. These correspond to the three conceptual goals or milestones in our work. As we explore each of these areas in turn, and how they interrelate, it becomes clear that there is not a complete split between them: as an example, the foundation for our present project, as described in [2], is just as much motivated by improving the practical performance and generality of a CCG system as it is an enrichment of the underlying theoretical formalism. Very often, theoretical research informs practical research, which is in turn informed by meta-research.

## 2.3 Motivating the CCG

Why, in the space of parsing formalisms, is the CCG a desirable direction for research? Section 2.2 refers to CCG as a *highly lexicalised* formalism; in the context of parsing formalisms, this indicates that the description of *syntax* (the definition of phrasal structure in a language) is given in the lexicon, in contrast to formalisms such as the familiar *context-free* or *phrase-structure grammar* which separates the production rules defining a language's syntax from its lexicon. Such formalisms enjoy several benefits: in particular, the two stages of lexicon acquisition and grammar acquisition are reduced to a single conceptual step, and the parser component is easier to generalise to the processing of arbitrary languages.

We commence our survey of the state of CCG research with a number of works which are at the core of our own work as well as that of the research community at large.

As Steedman argues in [39], just as most programming languages are devised to map a well-defined syntax onto a well-defined set of semantics to be interpreted by a machine, the analogous process of syntax mapping onto semantics is a major force in the human development and acquisition of natural languages. Thus, a major goal of CCG is to span the divide between representation and meaning in natural language, by characterising a fragment of language as precisely a function from its constituent words to a logical representation of its meaning. In unifying the naturally ambiguous and subjective concept of 'meaning' with the formal background provided by type theory and combinatory logic, CCG reduces the construction of meaning to the computational act of parsing.

Steedman has left four areas out of the scope of his survey: firstly, the implementation of a CCG parser is only discussed briefly, and only in the abstract. Issues which face all real-world implementations of the CCG, including *spurious ambiguity*: the existence of semantically identical, but syntactically distinct parses (an approach for dealing with this phenomenon is described in [18]), are not satisfactorily addressed. Steedman, in considering spurious ambiguity, considers it to be sometimes desirable, in that apparently 'spurious' parses may encode suprasegmental differences in information structure, without proposing a solution for the extant computational problem it causes. The second area absent from this work is the methodology behind inducing a CCG lexicon, which is arguably the aspect of producing a CCG system for which human intuition and effort is needed most. Since the ability to parse CCG is fundamentally dependent on the ability to correctly assign categories to lexical items, this is by all means an important consideration. The contributions of Hockenmaier in [21, 22, 27], to be introduced later, are important works in this area. The final area is evaluating the benefits of the CCG in the context of other formalisms of similar generative power. Steedman proves the weak equivalence in generative power between CCG and several other formalisms including LIG (linear indexed grammar) and TAG (tree adjoining grammar), but the practical benefits of CCG over such formalisms is left implicit. Lastly, no evaluation is considered with respect to the ability of CCG to reliably and accurately capture dependencies, its efficiency when applied to the general task of parsing, or its robustness in the face of the vast extent of structures encountered in human languages. It would be unusual for a seminal work such as *The Syntactic Process* to anticipate each of the above observations, but subsequent research addresses each of these issues, described hereafter.

## 2.4 Lexicons and data

Statistical methods for parsing depend critically upon the availability of an appropriately represented corpus. The advancement of CCG research has hinged critically on the availability of a wide-coverage lexicon produced for the CCG formalism. Similarly, since the refinements made through MMCCG mean that existing CCG resources are unsuitable for direct use in MMCCG applications, a prerequisite for the adoption of MMCCG as a general-purpose parsing discipline is the availability of a lexicon built for MMCCG. Several approaches including [17, 41], all employing the methodology of deriving a CCG lexicon from an existing corpus, have been successful in rectifying the absence of a wide-coverage CCG lexicon. We will achieve the goal of developing an MMCCG corpus by adopting the same methodology, beginning with the CCGbank, whose derivation from the Penn Treebank is described by Hockenmaier in [21].

*Data and Models for Statistical Parsing with Combinatory Categorial Grammar* [21] is immediately relevant to our own work, being one of two major sources of documentation, along with the CCGbank Users' Manual in [27], of the lexicon induction process developed and undertaken by Hockenmaier in the semi-automatic induction of CCGbank from the Penn Treebank. This work has two repercussions for our own research: firstly, we hope to model our own approach in deriving a MMCCG corpus after the structural techniques employed by Hockenmaier. Secondly, since our point of origin is the CCGbank, the key works [21, 27] serve as both a rationale for the choices made in the representation of CCGbank, as well as a reference for the CCGbank format, which is the input and output format for the MMCCG lexicon induction process we will ourselves devise. With reference to the discussion of the state of CCG research, [21] contributes to two areas: the advancement of CCG parsing techniques, as well as lexicon acquisition. The area most germane to our own work is Hockenmaier's description of the lexicon acquisition process, so in considering [21] we will focus on this aspect of the discussion.

The methodology employed in [21] for the development of CCGbank first involves a procedure which captures the general case of converting a (not necessarily binary) Penn Treebank tree into a binary-branching CCG derivation. A category assignment procedure then labels the nodes in the induced derivation. However, the Penn Treebank specially annotates a large number of special constructions in English which signal the application of various transformations acting upon syntactic structure. For example, the familiar *inversion* of the auxiliary with respect to its subject when forming an English question (eg: *Did I bring them to the office?*) results in a sentence-level tag of SQ rather than that of a declarative sentence S. Hockenmaier considers each of the structures tagged by the Penn Treebank and specifies transformations on the Penn derivation where necessary, to allow a CCG derivation to be generated with the aforementioned process of tree binarisation and category assignment.

A large number of problems in natural language processing can be characterised as modelling problems. Just as CCG itself is a *model* accounting for the complexity of natural language, [21] aims to model the data embodied in the Penn Treebank with a CCG analysis. Just as a researcher producing a mathematical model may choose to omit known factors which complicate the analysis, Hockenmaier identifies certain structures as too hard to detect, or render amenable to CCG analysis. When devising a mathematical model, simplifications made must be identified during the analysis and their effect on the evaluation discussed. Failing to capture a given syntactic structure has obvious effects on the ability of any system trained on that data to discriminate and correctly analyse an instance of it. In the pragmatic setting of wide-coverage parsing, resource limitations motivate parser implementers to identify those structures which occur most often, and concentrate their efforts on capturing the most common cases. However, omitting a given construction from the analysis by appealing to its paucity in the Penn Treebank is not entirely satisfactory. The Penn Treebank, although derived from a variety of sources including Dow Jones articles, sentences drawn from a technical manual, and the Brown corpus (itself intended to be drawn from a spectrum of American English text types), still comes with no guarantee of its representativeness.

A second issue is that [21] fails to evaluate the success of the CCGbank derivation process; no evaluation is given as to the coverage of the derived corpus, whether the derivation process yielded incorrect derivations, or the impact of the absence of syntactic structures the process does not account for. Therefore, a dependency-sensitive analysis conducted with metrics detailed in Section 2.5 against the output of a parser trained on CCGbank is the only evaluation given of its success as a corpus. This is not ideal in light of our focus on CCGbank not as a parser component but as the foundation for a derived corpus. Furthermore, no comparisons are drawn between the resulting corpus and CCG corpora derived by a similar methodology from source corpora other than the Penn Treebank.

In summary, the structure of our own project mirrors that of [21], which also consists of devising a corpus, applying it to an existing parser, and applying evaluation methods to the resulting parser. In the following section, we survey approaches to parser evaluation as applied to the CCG parsing task.

## 2.5 Approaches to evaluation

As identified in Section 2.1, our approach involves the derivation of a MMCCG corpus from the CCG corpus CCGbank, which is in turn derived from the Penn Treebank. Derivations in the Penn Treebank encode *phrase structure*, with special empty annotations for the positions originally filled by constituents moved by linguistic transformations such as wh-inversion or fronting, as well as part of speech (POS) tags on terminal nodes. Structurally, internal nodes may have an unbounded number of children. Parsing systems which employ the Penn Treebank directly as training or evaluation data often produce output similar in form to it, permitting evaluation to employ the Penn Treebank itself as the *gold standard* (output judged to be correct, against which parser output is compared and evaluated) with a cross-validation or holdout methodology. A very popular evaluation metric in systems which use the Penn Treebank is the PARSEVAL metric, as described in [5]. This metric calculates precision and recall with respect to the number of *groupings* which the candidate parse shares with a gold standard parse, so intuitively, PARSEVAL is intended to measure the similarity of a candidate parse to the gold standard with respect to tree structure.

[13] identifies the shortcomings of PARSEVAL with respect to CCG systems, and introduces two evaluation metrics in the context of CCG which capture various aspects of *dependency*, that is, the correct subordination of constituents in a candidate parse. This work is directly relevant to the evaluation section of our own work, since our overarching goal is to evaluate an MMCCG system against the plain CCG parser, as well as against parsers built for similar formalisms, in order to validate MMCCG as a practical choice of formalism for the general parsing task. The second dependency-sensitive evaluation metric described in [13] has been adopted as a *de facto* standard in evaluating CCG systems, so the evaluation of our own system must support the metric described in this paper.

The goal of any evaluation metric is to act as a model of the 'goodness' of a candidate parse. Clark's approach in proposing two dependency-sensitive evaluation metrics is first to identify the shortcomings of the existing PARSEVAL metric in relation to measuring the correctness of the parses issued by a CCG system, then to determine metrics which better capture the concept of 'goodness' in a way that matches our own intuition. The selection of an evaluation method can be posed as a modelling problem, in the sense that the success of an evaluation metric is determined by how well it reflects a gold standard, which in the parse evaluation task is necessarily that of a human. If a parse is 'good' or 'bad' by human standards, this should be reflected by the choice of metric.

As identified in [13], PARSEVAL is inappropriate for CCG evaluation because different sequences of combinator applications can yield derivations which are structurally distinct but semantically equivalent. PARSEVAL will punish a derivation which is structurally different from the gold standard, although it may be equivalent with respect to semantic structure. Furthermore, PARSEVAL ignores category assignments, only accounting for the equivalence of the lexical items at grouped leaf nodes. This implies that PARSEVAL cannot directly differentiate between syntactically distinct senses of the same word, for example the transitive verb *has* $\vdash (S\backslash NP)/NP$ and the perfect aspect auxiliary *has* $\vdash (S\backslash NP)/(S[pt]\backslash NP)$. The two dependency-sensitive metrics are designed to address both points, by emphasising the criterion of *semantic similarity* over syntactic similarity. Two trees are similar if they encode the same dependencies: that is, if the head-dependent relations in the gold standard hold in the candidate parse. Also, both dependency-sensitive metrics are based on category assignments rather than the textual correspondence of grouped elements.

Two prominent issues arise from the introduction of evaluation metrics for CCG: firstly, the 'goodness of fit' of a metric, that is, how well it matches the human ability to discriminate between correct and incorrect parses, depends on the application, as argued by [9]. As such, the choice of evaluation metric is less clear-cut and straightforward than, for example, the use of the F-score as a standard evaluation metric in a standard classification or retrieval problem.

| | ↓ Actual class | |
|---|---|---|
| ↓ Assigned | True | False |
| True | TP | FP |
| False | FN | TN |

Figure 2.2: Contingency matrix for classification

Secondly, again consider the standard binary classification/retrieval problem, whose evaluation is based on placing each test case in one of four bins according to its assigned class (an attribute which defines the very problem), as in Figure 2.2. By comparison, a dependency-sensitive evaluation relies upon an attribute of the output which is not always represented. Even in those parsers which do output dependency information, there is the problem of lost or inconsistent information when converting the candidate to the representation expected by the chosen evaluation method. [9] suggests that parsers in general be built with the explicit support of a parser- and framework-neutral representation to address this issue. In particular, although it is not currently a problem, since the only large-scale corpus of CCG is the aforementioned CCGbank, the possibility that various training corpora may represent categories differently raises questions of whether the second dependency-sensitive metric, which is a macro-average of F-scores calculated for each dependency relation, is valid when comparing systems founded on different representations of a given language. Naturally, we expect an evaluation metric to allow us to compare both intra-system performance as well as inter-system performance.

Despite these observations, we will target as the baseline for our evaluation a version of the original CCG parser which does not incorporate our MMCCG extensions. In doing so, we ensure that the output of our own parser retains enough information to allow its output to be compared with that of the original parser through a dependency-sensitive metric, although the exploration of parser- and framework-neutral evaluation methods is a beneficial direction for further research.

# Chapter 3

# Research plan

This section describes the conceptual steps in demonstrating the theses presented in Section 1.3, as well as the identification of contigencies which may arise at each stage of the research, the factors which may cause deviations from the schedule, and the measures we will take to mitigate against them.

**1** Creation of MMCCG corpus from the CCGbank corpus

**Weeks 1-2: 23/7–3/8**
Identifying methods for automatic or semi-automatic annotation of CCGbank with modes (statistical, or linguistically motivated methods)

**Weeks 3-5: 6/8–24/8**
Iteratively, until a desired level of corpus coverage is reached:

Determining structures which the above method fails to capture

Updating the annotation process to allow them to be analysed

**2** Modification of the C&C CCG parser to support the parsing of text through the MMCCG formalism

**Week 6:** Investigation of the structure of the C&C parser and implementation decisions made

Determining the structural modifications to the parser necessary to support MMCCG

*ALTA paper submission deadline in mid-September*

**Week 7-10: 3–26/9** Modifying parser to support MMCCG input (by extending the parser's input handling for CCGbank data)

Modifying parser to respect modes during parsing

Removal of existing hard-coded combinatory rule restrictions

**3** Training and evaluation

**Week 11-12: 29/9–2/10** Using the MMCCG corpus as input, train the modified C&C parser to yield a MMCCG parser built on a wide-coverage lexicon

Developing an evaluation framework which permits direct comparison with the evaluation for the unmodified C&C parser

Conducting selected evaluation methods and interpretation of results

Figure 3.1: Research plan overview

## 3.1 Development of automatic methods for MMCCG corpus extraction

**Description of task**

As embodied in Thesis 2, the most fundamental aspect of the investigation of MMCCG we intend to perform is the development of an MMCCG corpus. Such a corpus is required both for the training of the MMCCG parser we will develop in the second phase of our work, and of course for the evaluation of that parser against the unmodified C&C CCG parser which forms the third phase of our work.

**Time allocated for this stage**

We intend to spend two weeks (Weeks 1–2, 23/7–3/8) selecting an approach, refining it, and proposing transformations which together should offer a good initial level of coverage. After we have generated an initial iteration of the corpus, we will evaluate it by inspection, noting for which derivations the process has generated correct analyses. We will modify the transformation process to account for missed or incorrectly analysed cases, and return to the corpus generation stage. We intend for this to be an iterative process with a very tight cycle, although this assumption may be problematic as we identify below in 'Risk factors'. The corpus should demonstrate coverage and accuracy sufficient to the problem by the end of Week 6 (31/8), ready to be applied to the parser in the second stage.

**Objectives**

The goal of this first phase is to develop a conversion scheme which reads in a corpus encoded in the CCGbank format as documented in [27], and produces a MMCCG corpus, in an automatic, or mostly automatic fashion. Major criteria for evaluating such a corpus are coverage (the proportion of the source corpus which has been analysed), and how well the process derives correct MMCCG analyses for a broad spectrum of English text.

**Overview of methodology**

Recall that MMCCG allows us to specify restrictions on the set of rules by which a given category can be combined with others, by adding annotations known as *modes* to the slashes (/ and \) which introduce compound categories (functors). The objective of this stage is to supplement each category which occurs in the CCGbank corpus with these annotations. Although the precise methodology for this transformation is yet to be determined, we will initially investigate a statistical approach in which we consider the combinatory operations which do occur in the CCGbank. Assuming reasonable coverage, the fact that particular combinatory rules are uncommon, or do not occur at all in the context of a particular lexical item is evidence that those rules should be blocked by a combinatory rule restriction. Incorrect analyses will be isolated in a semi-automatic manner: we will verify a hypothesised set of annotated categories by determining whether our candidate annotations break existing analyses encoded in CCGbank. This process is semi-automatic, because noisy data in CCGbank may block a correct MMCCG analysis.

Since we do not expect our first pass over the data with a conversion algorithm to yield sufficient coverage and accuracy, we propose an iterative approach to corpus extraction, illustrated by the below pseudocode:

```
while corpus coverage is deemed insufficient do
    determine incorrectly analysed derivations
    modify the transformation process to account for these cases
    apply the corpus transformation process to CCGbank
end while
```

**Risk factors**

Two classes of factors may lead to this phase requiring more time than it is assigned. A first possibility is that the process given above of incremental development converges too slowly to our coverage goals. This may be due to the difficulty of accommodating a given analysis, or result from issues encountered in developing the extraction methodology. A possible cause of the process converging too slowly is that each cycle of the corpus generation process actually takes a non-trivial amount of time to complete due to the cost of the analysis performed as part of the process.

A second possibility is that our chosen extraction methodology is flawed, and it becomes necessary to reconsider our approach to the problem. For example, it may eventuate that CCGbank alone does not yield sufficient information to produce the annotations that we need. Otherwise, it may be possible that the statistical approach we sketch in the 'Overview' section above does not yield sufficiently accurate analyses with tokens which do not participate frequently in derivations in CCGbank.

**Mitigating risks**

We will address the impact of each of the factors identified above, and discuss measures to prevent them from eventuating. To mitigate against the risk that the process converges too slowly, we will employ a 'checklist' approach in obtaining sufficient corpus coverage. We will implement this approach by using the first two weeks of this stage analysing CCGbank to determine how commonly particular constructions occur. To obtain the greatest increases in coverage in each iteration, we will concentrate on handling the most common cases first, using the order of this checklist as a guide. Furthermore, we believe that insight acquired through handling each case will feed back into solving other cases of the general problem. Through this approach we hope to mitigate the risk of overly slow convergence.

The second risk identified is that our approach as formulated above in 'Overview of methodology' ends up being unsuitable to the problem for any reason: unsatisfactory coverage, incorrect analyses, or theoretical unsoundness. What we do not want, is for these flaws to be discovered in the process only after we have committed time and resources to its implementation, which we have scheduled as above for Weeks 3–6. This implies that we have to identify these flaws as early as possible: in the planning phase (Weeks 1–2). We will discover the flaws as soon as possible by implementing on a 'pilot' basis the approaches we intend to use. Although we clearly don't hope to achieve anywhere near the required level of coverage with such an implementation, it may uncover gross deficiencies in the approach as well as alerting us to cases which our full implementation will have to cover. A benefit of producing this proof-of-concept implementation at an early stage is that a successful implementation can be used as a prototype, or the foundation of the implementation which is the product of Weeks 3–6 of this stage.

We also have a final fallback in the event that all of the approaches to corpus extraction that we employ fail: the set of categories recognised by CCG numbers approximately 400; if all else fails, we can manually annotate this relatively small quantity of categories. Naturally, we lose the ability to generalise a method to arbitrary CCGbank-style corpora, but manual annotation would allow us to proceed to the third phase.

**Conference paper deadline**

We intend to submit a paper to the Australasian Language Technology Association (ALTA) workshop, whose submission deadline we expect to be set for mid-September. This paper should encompass a description of our approach and an evaluation for the corpus extraction phase of our research. We believe that feedback acquired from researchers in the NLP community will be highly valuable in refining our approach, and the content of the conference paper will inform the chapter in our thesis on corpus extraction.

## 3.2 Modifying the C&C CCG parser

**Description of task**

In this phase, we modify the implementation of the C&C parser, an efficient and practical implementation of a wide-coverage CCG parser [7, 11], to support the MMCCG extension. Having derived the MMCCG corpus in the first phase, through the course of this second phase we will apply a MMCCG parser for the first time to wide-coverage parsing through our modified version of the C&C parser package.

**Time allocated for this stage**

We have allocated four weeks (Weeks 6–10) to the second phase of our research. The first week will be spent familiarising ourselves with the existing codebase of the C&C parser, identifying the modules to which we need to make structural modifications. This will be done in close co-ordination with our supervisor, one of the developers of the parser, who will be able to answer questions on implementation decisions requiring explanation. The following three weeks we will dedicate to the implementation of multi-modal CCG in the C&C parser. Roughly speaking, we foresee this process to have three sub-stages: modifying the input handling, modifying the parser to respect modes, and lastly the removal of existing hard-coded restrictions on combinatory rules, the undesired presence of which was one of the motivating factors behind the MMCCG.

**Objectives**

At the conclusion of this phase, we will have a modified version of the C&C parser capable of both accepting the output of the first phase (a MMCCG annotated corpus) as training data, and parsing text in a manner which respects the restrictions afforded by modes. We should also develop an evaluation framework in this phase which allows the product of this phase to be compared in a meaningful way to characteristics of the unmodified C&C parser including efficiency and accuracy, in the final phase of our research.

**Overview of methodology**

This is the major implementation phase of our research. We will use a split of the MMCCG data available into three sets in preparation for the final phase of this project: a development set, a test set and a training set. We will create a small regression testing set from the development set, and devise by hand correct analyses for selected corpus sentences. As mentioned in Section 2.5, CCG does not place as great a degree of emphasis on structural equivalence as a means of evaluating correctness, since there may exist multiple derivations which encode the same dependency structure, so the data in the regression testing set will likely be specified in terms of dependency structures. As we develop the parser, we will ensure that any impact caused by modifications to the parser is well documented by comparing its output against the gold standard encoded in the regression testing set when changes are made to the codebase.

The exact nature of the development/test/training set split is given by convention in the natural language parsing research community, so for want of a reason to deviate from it, we will employ the same splitting convention.

**Risk factors**

The major risk in this phase is that the changes we need to make to the parser to support MMCCG may require major structural changes to the C&C parser codebase. As a parser written for practical parsing, in contrast to a proof-of-concept implementation of CCG and MMCCG such as that embodied by Grok [2], efficiency is placed above strict adherence to Steedman's formalism as it appears in [39], and it is unlikely that the implementation of the C&C parser was carried out with the extensibility of the underlying formalism in mind. Accordingly, we may find as we implement MMCCG in the C&C parser that existing implementation decisions impede our efforts to an unforeseen degree.

**Mitigating risks**

Our approach to alleviating this risk is to interleave aspects of this phase of our research with our work in the previous phase. That is, our decisions in developing the input for the modified C&C parser should be informed by the parser's structure and implementation. This gives us an opportunity to become acclimatised to the parser internals at an earlier stage, rendering the learning curve in this phase less steep. We believe that early exposure to the codebase will allow us to identify at an early stage issues which may lead to implementation problems in this phase. We believe it to be unlikely that we will encounter show-stopping issues, in the sense that any issues identified should not unreasonably impede our work in the second phase, although their presence may increase our workload for this phase. However, we will consider potential courses of action to be taken if aspects of C&C render it unsuitable for the modifications we seek to perform.

If implementation decisions embodied in C&C make MMCCG unreasonably difficult to implement, it will be necessary to simplify the formalism in a way which enables us to implement it within the four weeks we have allocated for the task. In particular, this is likely to involve implementing only a subset of the combinatory rule restrictions which MMCCG defines. Even if we identify show-stopping issues in the first phase, we will generate an MMCCG corpus for the full formalism instead of doing so for a simplified and incomplete form of MMCCG, since we do not wish to diminish the usefulness of a MMCCG corpus for use by other researchers. Instead, we would map the MMCCG corpus onto the restricted subset of MMCCG as required by our circumstances, as a post-processing step.

## 3.3 Training and evaluation

**Description of task**

This phase ties together the first phase (the development of a MMCCG corpus) and the second phase (the implementation of a MMCCG parser) by training this parser on the corpus we have derived. Through the evaluation of the modified C&C MMCCG parser against the unmodified CCG parser, we want to determine the impact which MMCCG has on the characteristics of a parsing system, and in doing so verify the hypotheses and benefits which Baldridge ascribes to MMCCG over CCG in [2].

**Time allocated for this stage**

We have designated the remaining three weeks for refining the evaluation framework, performing the evaluation and interpreting the results. Conducting our evaluation replicating the methodology of [11] to allow us to compare it meaningfully to the modified parser is a step which will take significant wall-clock time. Therefore, we have allowed two weeks for the acquisition of the metrics by which we will determine the success of our research, and consequently the utility of MMCCG. While experiments are being conducted, we hope to refine the thesis document, ensuring that it accurately reflects our work.

In the final week of the project, we will collate the experimental results, and interpret them in the light of those obtained for the unmodified C&C parser, employing statistical tests of significance so we can make meaningful statements about observed changes in performance. We will examine the results keeping in mind Baldridge's claims on MMCCG, evaluating the degree to which these claims are supported by the results we have obtained.

**Objectives**

We propose three main goals for this final phase of the research:

- Making the modifications necessary to allow the direct comparison of the unmodified implementation C&C with a version incorporating our MMCCG extensions
- Conducting experiments on our parser in conditions which replicate the experimental setup undertaken in [11]
- Evaluating Baldridge's claims in [2] with the above comparison as evidence

**Overview of methodology**

An overview and justification of our proposed evaluation methodology is presented in Section 1.5. A summary of our approach is to perform the same evaluation as done for the unmodified C&C parser in [11], replicating the dataset and choice of parameters employed for that evaluation.

We will employ a dependency-based evaluation, to allow us to conduct our evaluation against that of the C&C parser, and also because we believe it to be more suitable for the problem of evaluating a CCG-based parser, as argued in Chapter 1.

**Risk factors**

Although the majority of the workload is concentrated in the first two phases of the project, given the possibility that our suite of experiments may have to be repeated numerous times, there is considerable room for slippage at this stage. We also believe the process of adapting the existing evaluation framework employed for C&C to MMCCG to be straightforward, but modifications to the evaluation framework required to handle MMCCG categories may introduce additional workload at this late stage. Finally, it may be possible that our experimental results do not appear to support Baldridge's claims, an outcome whose causes we will have to analyse and explain.

**Mitigating risks**

The aspects of the experimental setup which we need to duplicate are restricted to methodological concerns: as long as we can implement the same metrics employed in the evaluation of C&C, we believe that we can meaningfully compare them to our results. If it eventuates that the dependency-based evaluation employed for CCG is invalid for MMCCG, we will have to modify the evaluation method such that it admits both a CCG analysis as well as a MMCCG analysis, and reproduce the experiment on the original C&C parser instead of employing the evaluation already performed in [11]. To prevent such issues from causing slippage at such a critical juncture, we intend to implement the evaluation framework as part of the second phase. As well as alerting us earlier to issues with the evaluation framework, we ensure that implementation decisions made in modifying the parser are made with the evaluation task in mind. For example, we should ensure that the modified parser is able to produce the dependency representation expected by the dependency-sensitive evaluation metric we intend to use.

# Bibliography


[1] J. Baldridge. A categorial account of extraction asymmetries in Tagalog and Toba Batak. *Manuscript, Division of Informatics, University of Edinburgh*, 1999.

[2] J. Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2002.

[3] J. Baldridge and G.-J. M. Kruijff. Multi-modal combinatory categorial grammar. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 211–218, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[4] S. Bangalore, A. Sarkar, C. Doran, and B. Hockey. Grammar and parser evaluation in the XTAG project. In *Proceedings of the First Conference on Language Resources and Evaluation*, 1998.

[5] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, et al. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the February 1991 DARPA Speech and Natural Language Workshop*, volume 10, Pacific Grove, CA., 1991.

[6] J. Bos. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS*, volume 6, pages 42–53, 2005.

[7] J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 1240–1246. Association for Computational Linguistics Morristown, NJ, USA, 2004.

[8] R. Çakıcı. Automatic induction of a CCG grammar for Turkish. *ACL Student Research Workshop*, pages 73–78, 2005.

[9] J. Carroll, T. Briscoe, and A. Sanfilippo. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 447–454, 1998.

[10] S. Clark and J. R. Curran. Log-linear models for wide-coverage CCG parsing. In *Proceedings of the EMNLP Conference*, pages 97–104, 2003.

[11] S. Clark and J. R. Curran. Parsing the WSJ using CCG and log-linear models. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 104–111, Barcelona, Spain, 2004. Association for Computational Linguistics.

[12] S. Clark and J. R. Curran. Wide-Coverage Statistical Parsing with CCG and Log-Linear Models. *To appear in* Computational Linguistics 2007. *Unpublished draft manuscript*, 2006.

[13] S. Clark and J. Hockenmaier. Evaluating a Wide-Coverage CCG Parser. In *Proceedings of the LREC 2002 Beyond PARSEVAL workshop*, pages 60–66, 2002.

[14] S. Clark, M. Steedman, and J. R. Curran. Object-extraction and question-parsing using CCG. In *Proceedings of the SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, pages 111–118, 2004.

[15] J. R. Curran and S. Clark. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Meeting of the EACL*, pages 91–98, Budapest, Hungary, 2003.

[16] H. Curry, R. Feys, and W. Craig. *Combinatory logic*. North-Holland Publishing Company, 1958.

[17] C. Doran and B. Srinivas. Bootstrapping a Wide-coverage CCG from FB-LTAG. Technical report, Université Paris, 1994. Technical Report TALANA-RT-94-01, TALANA.

[18] J. Eisner. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 79–86. Association for Computational Linguistics Morristown, NJ, USA, 1996.

[19] R. A. Frost. Realization of natural language interfaces using lazy functional programming. *ACM Comput. Surv.*, 38(4):Article 11, 2006.

[20] J. Hockenmaier. Statistical parsing for CCG with simple generative models. In *ACL (Companion Volume)*, pages 7–12, 2001.

[21] J. Hockenmaier. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2003.

[22] J. Hockenmaier. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 505–512. Association for Computational Linguistics Morristown, NJ, USA, 2006.

[23] J. Hockenmaier, G. Bierner, and J. Baldridge. Providing robustness for a CCG system. *Unpublished manuscript, University of Edinburgh*, 2000.

[24] J. Hockenmaier, G. Bierner, and J. Baldridge. Extending the Coverage of a CCG System. *Research on Language & Computation*, 2(2):165–208, 2004.

[25] J. Hockenmaier and M. Steedman. Generative models for statistical parsing with combinatory categorial grammar. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 335–342, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[26] J. Hockenmaier and M. Steedman. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1974–1981, 2002.

[27] J. Hockenmaier and M. Steedman. CCGbank: Users' manual. Technical report, Technical Report MS-CIS-05-09, Computer and Information Science, University of Pennsylvania, 2005.

[28] B. Hoffman. A CCG approach to free word order languages. In *Proceedings of the 30th conference on Association for Computational Linguistics*, pages 300–302. Association for Computational Linguistics Morristown, NJ, USA, 1992.

[29] B. Hoffman. Integrating "free" word order syntax and information structure. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 245–252, 1995.

[30] B. Hoffman. *The computational analysis of the syntax and interpretation of free word order in Turkish*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1996.

[31] G. Kruijff. A brief introduction to specifying grammars for OpenCCG. `http://www.coli.uni-sb.de/cl/courses/CG-03/materials/openccg.introduction.pdf`, 2003. Draft, Accessed 29 March 2007.

[32] G. Kruijff and J. Baldridge. Generalizing Dimensionality in Combinatory Categorial Grammar. In *Proceedings of the 20th international conference on Computational Linguistics*, 2004. Article 191.

[33] Y. Kubota and E. Smith. Syntax and Semantics of Japanese Nonconstituent Clefting in Combinatory Categorial Grammar. *Unpublished manuscript, to appear in Proceedings of the 36th North East Linguistic Society*, 2006.

[34] McConville, Mark. Incremental natural language understanding with combinatory categorial grammar. Master's thesis, School of Cognitive Science, Division of Informatics, University of Edinburgh, 2001.

[35] M. Steedman. Dependency and Coördination in the Grammar of Dutch and English. *Language*, 61(3):523–568, sep 1985.

[36] M. Steedman. Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, 5(3):403–439, 1987.

[37] M. Steedman. Type-raising and directionality in combinatory grammar. In *Proceedings of the 29th conference on Association for Computational Linguistics*, pages 71–78. Association for Computational Linguistics Morristown, NJ, USA, 1991.

[38] M. Steedman. A very short introduction to CCG. *Unpublished tutorial paper. Available at* `http://www.cogsci.ed.ac.uk/steedman/papers.html`, 1996. Accessed 27 March 2007.

[39] M. Steedman. *The Syntactic Process*. MIT Press, 2001.

[40] M. Steedman. Information-Structural Semantics for English Intonation. *Draft workshop paper for the* LSA Summer Institute Workshop on Topic and Focus, Santa Barbara, 2003. Available on author's site at `ftp://ftp.cogsci.ed.ac.uk/pub/steedman/prosody/santabarbara.pdf`. Accessed 14 April 2007.

[41] A. Villavicencio. Building a wide-coverage Combinatory Categorial Grammar. Master's thesis, Cambridge University, 1997.

[42] M. White and J. Baldridge. Adapting chart realization to CCG. In *Proceedings of EWNLG9*, 2003.