

# A Logical Approach to Sentiment Analysis

Niklas Christoffer Petersen



Kongens Lyngby 2012  
IMM-MSc-2012-????

Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 321, DK-2800 Kongens Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
[reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)  
[www.imm.dtu.dk](http://www.imm.dtu.dk)

IMM-MSc-2012-????

# Summary (English)

---

The goal of the thesis is to ...



# Summary (Danish)

---

Målet for denne afhandling er at ...



# Preface

---

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in partial fulfilment of the requirements for acquiring an MSc in Computer Science and Engineering.

The thesis deals with ...

The thesis consists of ...

Furthermore three project-specific learning objectives for the project concerned should be presented. The following is simply suggestions for those:

- Understand and extend modern techniques for processing of natural language texts using formal logical systems.
- Demonstrate methods for formal reasoning with respect to natural language understanding.
- Present a *proof of concept* system, that is a fully functional implementation of essential theoretical presented methods.

Niklas Christoffer Petersen



# Acknowledgements

---

I would like to thank my....



# Contents

---

<b>Summary (English)</b>	<b>i</b>
<b>Summary (Danish)</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Classical data collection . . . . .	2
1.2 Natural language data collection . . . . .	3
1.3 Related work . . . . .	6
1.4 Using real data sets . . . . .	8
<b>2 Sentiment analysis</b>	<b>9</b>
2.1 Tokenization . . . . .	10
2.2 Lexical-syntactic analysis . . . . .	11
2.3 Mildly context-sensitive grammars . . . . .	12
2.4 Semantic analysis . . . . .	13
2.5 Data acquisition . . . . .	15
2.6 Test dataset . . . . .	15
<b>3 Combinatory categorial grammar</b>	<b>17</b>
3.1 Combinatory rules . . . . .	19
3.2 Coordination . . . . .	23
3.3 Features and agreement . . . . .	24
3.4 Lexicon acquisition . . . . .	25
3.5 Part of speech tagging . . . . .	25
3.6 Annotating the lexicon . . . . .	26

---

<b>4</b>	<b>Semantic networks</b>	<b>27</b>
<b>5</b>	<b>Implementation</b>	<b>29</b>
5.1	WordNet interface . . . . .	30
<b>6</b>	<b>Test results</b>	<b>31</b>
<b>7</b>	<b>Conclusion</b>	<b>33</b>
7.1	Discussion . . . . .	33
<b>A</b>	<b>Appendix A</b>	<b>35</b>
A.1	A naive attempt for lexicon acquisition . . . . .	35
A.1.1	The Brown Corpus . . . . .	35
A.2	Tokenizer and tagger . . . . .	36
A.2.1	Shift-reduce parser . . . . .	36
A.3	Find a good title . . . . .	36
	<b>Bibliography</b>	<b>39</b>

## CHAPTER 1

# Introduction

---

In the UK, online retail sales account for over 10% of purchases, and their growth rate is markedly outstripping store-based sales [13]. Many customers express their opinions about products through online reviews. These reviews are key for marketing intelligence since they contain valuable information. Popular products may have hundreds of reviews making it hard for the customer to find the information they require, and as a result there is a need to automatically classify this data. This can benefit both customers and manufacturers of products. Customers can see what other consumers thought about the products', viewing the products strengths and weaknesses. Manufacturers can then see where their product falls short in order to improve it, and also they can compare their products to other competitive products

Cite, rewrite, *Sentiment Analysis of Customer Reviews ...*

Noisy unstructured text data is generated in informal settings such as online chat, emails, blogs, customer feedbacks and reviews. These texts have the potential to act as rich sources for raw inputs to market research and knowledge discovery. Since Internet is a crucial driving force in today's world, these texts are rich pointers to the collective opinion of the global population on almost every topic.

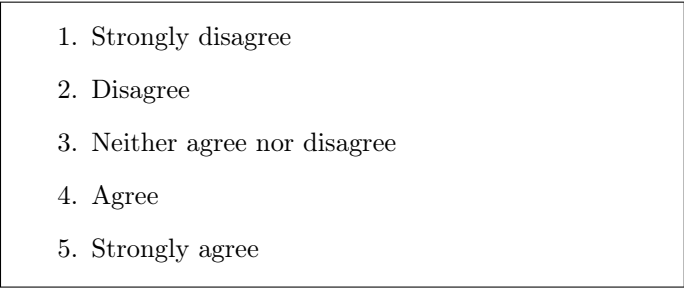
Cite, rewrite, *Opinion Mining From Noisy Text Data*

Feedback, in form of product and service reviews, can thus be highly valuable information for companies. Also political opinions are of high value for both governments and their the oppositions. The interest in such opinions is far from recent, and is a well established subfield of the *psychometrics* and has strong scientific grounds in both psychology and statistics.

Introduce *sentiment analysis* (also sometimes referred as *opinion mining*)

## 1.1 Classical data collection

One of the most used approaches to collect data for opinion analyses is through questionnaire surveys. Most of us are familiar with such surveys, where the subject is forced to answer questions with a fixed scale. For instance, given the statement “The rooms at the Swissôtel Hotel are of high quality.”, a subject must answer by selecting one of a predefined set of answers, e.g. as shown in figure 1.1

- 
1. Strongly disagree
  2. Disagree
  3. Neither agree nor disagree
  4. Agree
  5. Strongly agree

**Figure 1.1:** Likert scale.

Such scales, where the subject indicates the *level of agreement*, are known as *Likert scales*, originally presented by Likert [1932], and has been one of the favourite methods of collection data for opinion analyses cf. [?]. Other scales are also widely used, for instance the *Guttman scale* [?], where the questions are binary (yes/no) and ordered such that answering yes to a question implies the answer yes to all questions ordered

below this. Thus the answer on a Guttman scale can be captured by a single index. An example of an Guttman scale is shown in figure 1.2.

1. I like eating out
2. I like going to restaurants
3. I like going to themed restaurants
4. I like going to Chinese restaurants
5. I like going to Beijing-style Chinese restaurants

**Figure 1.2:** Guttman scale.

Given a set of answers, the result of such surveys are fairly easy to compute. At its simplest it can be an average of the answers, however mostly it is also interesting to connect the questions – for instance how does subjects’ answer to the above statement influent their answer to the statement “The food at the Swissôtel Restaurant are of high quality.”, etc.

List some shortcomings of this approach, e.g. predefined frame for feedback etc., it is often hard to get people to participate, etc.

## 1.2 Natural language data collection

In this thesis it is argued that a far more natural way for subjects to express their opinions is through their most natural communication form, i.e. their language, either in written or spoken form.

The initiative for such data collection could be *opinion seeking queries* as the one shown in (1.1). Such queries are intended to ensure succinct reviews that clearly relate to the *entity* in question (e.g. product or service) with respect to a specific

*topic of interest*. Clearly the queries could be formulated in a more friendly and *call to action* manner, as shown in (1.2)

*Holiday Inn, London: price* (1.1)

*What do you think about pricing at the Holiday Inn, London?* (1.2)

This method might not seem that different from that of the previously mentioned Likert scales, but it still allows the reviewer to answer with a much broader sentiment and lets the reviewer argue for his/hers answer as shown in the examples (1.3-1.4).

*The price is moderate for the service and the location.* (1.3)

*Overall an above average hotel based on location and price but not one for a romantic getaway!* (1.4)

More passive sources could also be considered, including posts on social networking services and microblogging services (e.g. Facebook<sup>1</sup> and Twitter<sup>2</sup>). This though introduces the need for efficient candidate filtering, as the posts in general of cause are not constrained to a specific entity or topic of interest. However it also significantly increases the data quantity, which in turn can yield a more precise analysis. Since the author of the post might never realize that the post is being used for the purpose of opinion analysis it also raises ethical issues. Larger texts, such as blog posts, could also be considered, however the contextual aspects of large, contiguous texts often makes interpretation extramly complex, thus making it a most difficult task to extract opinions on a specific entity.

It is also worth mentioning the possibility to collect answers such as (1.3, 1.4) in spoken form, which would give perhaps the most natural interaction. However for the purpose of this thesis it is proposed to solely focus on language in written form, as spoken form introduces a lot of complexity due to the speech recognition needed, for instance efficient audio sampling and analysis, speaker dependence (e.g. dialects), etc. Further more the interpretation of spoken language is also highly complex, since the emotional perception of the speaker must be considered in order to detect for instance ironic statements.

The overall goal is thus to perform a *sentiment analysis* of a set of small, succinct, review texts with respect to a subject, and yield an normalized score.

---

<sup>1</sup>Facebook, <http://www.facebook.com/>

<sup>2</sup>Twitter, <http://www.twitter.com/>



Fit

We will present the problematics that arises when trying to extract the semantic of texts from a language with a large vocabulary, and present several computational approaches for solving (at least partly) these issues.

## 1.3 Related work

In the following notable related work on sentiment analysis are briefly presented, and based on this it is argued that there are two main approaches for sentiment analysis of written texts, namely using *formal systems* and respectively using *statistical methods*.

- **Formal approaches** With this approach the language of the texts to analyse is modeled as a formal language, i.e. using a formal grammar. This allows a syntactical analysis of the texts, yielding the structures of the texts, e.g. sentences, phrases, and words. Semantic information is then extractable by augmenting and inspecting these structures.

The result of the semantic analysis is then subject to the actual sentiment analysis ...

Cites

- **Statistical approaches** With a statistical approach a probabilistic model is constructed, and trained against a training data set, configuring the parameters of the model (of which there can be an tremendous amount of). The model is then applied to the actual data set of which an analysis is desired.

can extract features from the input text. These features can then be analyzed in order to extract semantic properties.

Cites: Language models, positional language model, HHM

Both of these has been considered as the foundation of the analysis of the text reviews, however it is proposed to follow a logic approach for several reasons:

- Given that it is a formal system, it can be modeled closely and hopefully also efficiently in the *proof of concept* implementation.
- It is questionable whether the entropy of each topic in the chosen dataset actually allows feature extraction on a significant level. Given a set of text on a topic is divided into a training set and a test set, it is doubtful that the trained model would be able to capture features from the test set, since the topics only contains approximately a hundred texts.
- The undersigned has far more experience in the field of logic systems, than in the field of statistics, thus a higher level

Notice that even though it is proposed to follow a logic approach, it is still the intention to retain the proposed solutions for misspellings and minor grammatical

---

errors, which utilize some probabilistic properties, e.g. *match scores*, etc. However these should be seen solely as preprocessing steps needed, in case the input text cannot be analyzed directly, e.g. as a failover precaution. Thus for perfect texts, the analysis will be purely logic.

## 1.4 Using real data sets

For the presented solution to be truly convincing we wish to present a *proof of concept* implementation that shows at least some of the desired capabilities. However, for such product to be demonstrated properly, real data is required. Testing in on some tiny pseudo data set constructed for the sole purpose of this demonstration would not be convincing.

Move to chap. 2?

Dealing with major grammatical errors, such as wrong word order is a much harder problem, since even small changes in for instance the relative order of subject, object, verb etc. may result in an major change in interpretation. Thus it is proposed, only to focus on minor grammatical errors such as incorrect form. Such corrections could be made achievable by linking different forms of the same word.

Blog texts on the other hand suffer from ill-composed sentences, arbitrary punctuations or insertions of characters, irrational capitalization etc. Spelling correction, abbreviation and case restoration mechanisms for noisy text have been addressed in [3, 4]. We have employed variations of these techniques for our work.

Should be 3-5 pages: Motivation, Project goals, and Thesis structure

# Sentiment analysis

---

For the purpose of this thesis a *sentiment analysis* is defined as follows:

**DEFINITION 2.1** A sentiment analysis  $\mathcal{A}$  is a computation on a review text  $T \in \Sigma^*$  with respect to a subject  $s \in S$ , where  $\Sigma^*$  denotes the set of all permissible texts in the language. The result is an normalized score as shown in (2.1). The yielded score should reflect the *polarity* of the subject in the text, i.e. whether the overall opinion is positive (a score close to  $\iota$ ), negative (a score close to  $-\iota$ ), or neutral (a score close to 0).

$$\mathcal{A} : \Sigma^* \rightarrow S \rightarrow [-\iota; \iota] \quad (2.1)$$

It should be evident that this computation is far from trivial, and constitutes the cornerstone of this thesis. There are several steps needed, if such computation should yield any reasonable result. As mentioned in the introduction the goal is a logical approach for achieving this. The following outlines the overall steps to be completed, their associated problematics in this process, and succinctly presents different approaches to solve each step. The chosen approach for each step will be presented in much more details in later chapters. Finally ...

Should  $\Sigma^*$  be a string, and then include  $\perp$  as result?

Something about data acquisition of test-data

## 2.1 Tokenization

In order to even start processing natural language texts, it is essential to be able to identify the elementary parts, i.e. *lexical units* and *punctuation marks*, that constitutes a text. However even identifying the different sentences in a text can yield a difficult task. Consider for instance the text (2.2) which is taken from the Wall Street Journal (WSJ) corpus [Paul and Baker, 1992]. There are six periods in it, but only two of them indicates sentence boundaries, and delimits the text into its two sentences.

*Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.* (2.2)

The domain of small review texts allows some restrictions and assumptions, that at least will ease this issue. For instance it is argued that the review texts will be fairly succinct, and thus it seems like a valid assumption that they will consist of at few sentences. It is argued that this indeed is achievable by sufficient instructing and constraining the reviewers doing data collection, e.g. only allowing up to a certain number of characters. This allows sentences in such phrases to be processed independently (i.e. as two separate review texts).

Even with this assumption, the process of identifying the sentences, and the lexical units and punctuation marks within them, is not a trivial task. Webster and Kit [1992] criticizes the neglect of this process, as most NLP studies focus purely on analysis, and assumes this process has already been performed. Such common assumptions might derive from English being a relatively easy language to tokenize. This is due to its space marks as explicit delimiters between words, as opposed to other languages, e.g. Chinese which has no delimiters at all. This might hint that tokenization is very language dependent. And even though English is considered simple to tokenize, a naive approach like segmenting by the occurrence of spaces fails for the text (2.3), which is also from the WSJ corpus, as it would yield lexical units such as “(or”, “perceived,” and “rate),”. Simply consider all groups of non-alphanumerics as punctuation marks does not work either, since this would fail for i.a. ordinal numbers, currency symbols, and abbreviations, e.g. “Nov.” and “Elsevier N.V.” in text (2.2). Both of these methods also fail to recognize “Pierre Vinken” and “Elsevier N.V.” as single proper noun units, which is arguably the most sane choice for such.

*One of the fastest growing segments of the wine market is the category of superpremiums – wines limited in production, of exceptional quality (or so perceived, at any rate), and with exceedingly high prices.* (2.3)

Padró *et al.* [2010] presents a framework of analytic tools, developed in the recent years, for various NLP tasks. Specially interesting is the morphological analyzer, which applies a cascade of specialized (i.e. language dependent) processors to solve exactly the tokenization. The most simple use pattern matching algorithms to recognize numbers, dates, quantity expressions (e.g. ratios, percentages and monetary amounts), etc. More advanced processing are needed for proper nouns, which relies on a two-level solution: first it applies a fast pattern matching, utilizing that proper nouns are mostly capitalized; and secondly a more advanced ... TODO Cite: (Named entity extraction using adaboost). These recognize proper nouns with accuracy of respectively 90% and over 92%. The analyzer also tries to identify lexical units that are composed of multiple words, e.g. proper nouns and idioms.

It is thus possible, by the use of this framework, to preprocess the raw review texts collected from users, and ensure that they will be tokenized into segments that are suitable for the lexical-syntactic analysis. Thus more details on the tokenizer as regards design will not be presented.

## 2.2 Lexical-syntactic analysis

The syntactic analysis determines the grammatical structure of the input texts with respect to the rules of the English language. It is expected that the reader is familiar with English grammar rules and syntactic categories, including phrasal categories and lexical categories (also called parts of speech), otherwise (TODO: APPENDIX?). As mentioned earlier it is essential that the chosen solution is able to cope with *real* data, collected from actual review scenarios. This implies a robust syntactic analysis accepting a large vocabulary and a wide range of sentence structures. In order to calculate the actual polarity it is essential to have semantic annotations on the lexical units. It is argued that a feasible and suitable solution is to use a grammar that is *lexicalized*, i.e. where the rules are essentially language independent, and the syntactic properties are derived from a lexicon. Thus the development of a lexicalized grammar is mainly a task of acquiring a suitable lexicon for the desired language.

Even though the task of syntactic analysis now is largely reduced to a task of lexicon acquisition, which will be addressed in chapter 3, there are still general concerns that are worth acknowledging. Hockenmaier *et al.* [2004, p. 108-110] identifies several issues in being able to efficiently handle natural language texts solely with lexicalized grammars, mainly due to the need for entries for various combinations of proper nouns, abbreviated terms, dates, numbers, etc. Instead they suggest to use pattern matching and statistical techniques as a preprocessing step, for which efficient components exist, which translate into reduced complexity for the actual syntactic analysis.

Even though it can be argued that the use of proper nouns and specific dates is fairly limited in review text, in that a context have already been established for the reviewer cf. section 1.2, it is still

However the domain of small review texts also introduce concerns that are absent from other domains, including the possibility of incorrect grammar and spelling, since the texts comes unedited from humans with varying English skills. A solution that would only work on *perfect texts* (i.e. texts of sentences with completely correct grammar and spelling) would not be adequate. In order to at least try to handle minor misspellings it is intended to use algorithms that can select alternatives from the lexicon. Reasons for this could be that word is simply unresent from the system's vocabulary (e.g. misspelled), or on a grammatical incorrect form (e.g. wrong person, gender, tense, case, etc.).

## 2.3 Mildly context-sensitive grammars

There exists formal proofs that some natural language structures requires formal power beyond *context-free grammars* (CFG), i.e. [Shieber, 1985] and [Bresnan *et al.*, 1982]. Thus the search for grammars with more expressive power has long been a major study within the field of computational linguistics. The goal is a grammar that is so restrive as possible, allowing efficient syntactic analysis, but still capable of capturing these structures. The class of *mildly context-sensitive grammars* are conjectured to be powerful enough to model natural languages while remaining efficient with respect to syntactic analysis cf. [Joshi *et al.*, 1990].

Different grammar formalisms from this class has been considered, including *Tree Adjunct Grammar* (TAG) [Joshi *et al.*, 1975], in its lexicalized form (LTAG), *Head Grammar* (HG) [Pollard, 1984] and *Combinatory Categorical Grammar* (CCG) [Steedman, 1998]. It has been shown that these are all equal in expressive power by Vijay-Shanker and Weir [1994]. The grammar formilism chosen for the purpose of this thesis is *Combinatory Categorical Grammar* (CCG), pioneered largely by Steedman [2000]. CCG adds a layer of combinatory logic onto pure Categorical Grammar, which allows an elegant and succinct formation of *higher-order* logical expressions directly from the syntactical analysis. It is this capability, which is very usefull with respect to the goal of sentiment analysis, that mainly made the choice. Chapter 3 will formally introduce the CCG in much more detail.

CCG IS LOGICAL!!! - i.e. is uses combinatory logic



## 2.4 Semantic analysis

The overall process of semantic analysis of a text in the context of sentiment analysis is to identify the polarity of the subjects appearing in the text. The approach is to *annotate* the lexical units of adjectives and adverbs with suitable polarities, and then fold these onto the phrasal structures, yielded by the syntactical analysis, in order to identify the bindings of these polarities, i.e. which subjects they modify directly or indirectly.

There exists datasets that tries to bind a general polarity to each word in a lexicon, e.g. [Esuli and Sebastiani, 2006] and [Baccianella *et al.*, 2010]. While such might be fine for very general sentiment analyses, it is argued that better results can be achieved by using a context specific annotation. For instance the adjective “huge” might be considered positive for a review describing rooms at a hotel, while negative for a review describing cell phones.

adj?

As already mentioned, the use of a lexicalized syntactic analysis allows the annotation to appear directly on the entries in the lexicon. However, a manual annotation of a large lexicon is evidently not a feasible approach. Furthermore the solution must also be generic enough so it can be applied to different review contexts with minimum efforts. The concept of *semantics networks*, as [Russell and Norvig, 2009, p. 454–456]

### Semantic networks

A semeantic network is ...

A semantic network, or frame network, is a network which represents semantic relations between concepts. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges.[1]

Adjectives

Adverbs - intensifiers (very)

### Using real data-sets

Continue...

## 2.5 Data acquisition

In order to successfully perform the proposed computation, and thus the sentiment analysis

### Tagged corpora / Part-of-speech tagging

Since English is

*As with English around the world, the English language as used in the United Kingdom and the Republic of Ireland is governed by convention rather than formal code: there is no equivalent body to the Académie française or the Real Academia Española, and the authoritative dictionaries (for example, Oxford English Dictionary, Longman Dictionary of Contemporary English, Chambers Dictionary, Collins Dictionary) record usage rather than prescribe it. In addition, vocabulary and usage change with time; words are freely borrowed from other languages and other strains of English, and neologisms are frequent.*

[http://en.wikipedia.org/wiki/British\\_English#Standardisation](http://en.wikipedia.org/wiki/British_English#Standardisation)

## 2.6 Test dataset

### The Opinions Dataset

The suggested data set to use is the *Opinions Dataset*, originally used by [Ganesan *et al.*, 2010]. The data set consists of texts from actual user reviews on a total of 51 different topics. The topics are ranging over different objects, from consumer electronics (e.g. GPS navigation, music players, etc.) to hotels, restaurants and cars. For most of the objects, reviews are covered by multiple topics. For instance a specific car is covered by the topics *comfort*, *interior*, *mileage*, *performance*, and *seats*.

It has been hard to find any real alternatives for the *Opinions Dataset* for several reasons: Most collected reviews are commercial, and thus not free to use; furthermore the *Opinions Dataset* also contains summarized texts for each of its topics, which are constructed by manual, human interpretation. The latter allow a straight approach

for comparison of any results the proposed system will yield.

*The hotel buffet had fabulous food.* (2.4)

*Very friendly servers and nice selection of food at a reasonable price.* (2.5)

*Room service was extortionate though, very very expensive,  
so we didnt bother, as food outlets a few minutes walk away.* (2.6)

The texts (2.4) to (2.6) show actual extracts from the data set for a topic on food quality on the Swissôtel Restaurant. While (2.4) is a valid declarative sentence, (2.5) is not, since it lacks a subject (i.e. the restaurant). A coarse review of the text in the dataset reveals that missing subjects are a repeating issue. This might not seem that odd, since many people would implicitly imply the subject from the topic that they are reviewing. Thus text missing subjects can in many cases still be considered as valid sentences with minimal effort. The text (2.6) is on the other hand missing a transitive verb (presumably *are*) from the subordinate clause. In cases where such severe grammatically errors occurs it is suggested to ignore the clause, and try only to analyse the main clause. Furthermore the text (2.6) use repeated adverbs (e.g. *very very*) to express intensification, however it should not be any major concern that a verb or adjective are modified multiple times by the *same* adverb, but the intended intensification will probably not be included in the semantic analysis. Thus formalizing such a grammar is mostly a task of designing such lexicon.

As evident from these examples far from all texts in the dataset are valid sentences.

# Combinatory categorial grammar

---

In this chapter the formalism of Combinatory Categorical Grammar (CCG) is introduced, and based on this applied to the proposed sentiment analysis introduced in the previous chapter. For the purpose of explaining and demonstrating CCG a small fragment of English is used. This allow the usage of a “handwritten” lexicon initially. In section 3.4 the issues related to acquiring, and analysing with, a wide coverage lexicon are addressed. For the syntactic analysis a CCG lexicon is defined as follows:

**DEFINITION 3.1** A CCG lexicon,  $\mathcal{L}_{\text{CCG}}$ , is mapping from a lexical unit,  $w \in \Sigma^*$ , to a set of 2-tuples, each containing a lexical category and semantical expression that the unit can entail cf. (3.1), where  $\Gamma$  denotes the set of lexical and phrasal categories, and  $\Lambda$  denotes the set of semantical expressions.

$$\mathcal{L}_{\text{CCG}} : \Sigma^* \rightarrow \mathcal{P}(\Gamma \times \Lambda) \quad (3.1)$$

A *tagging* of a lexical unit  $w \in \Sigma^*$  is simply the selection of one of the pairs yielded by  $\mathcal{L}_{\text{CCG}}(w)$ . Thus given some ordered set of lexical units, which constitutes the text  $T \in \Sigma^*$  to analyse, there might exists many different taggings. This is simply due to the fact that a lexical unit can entail different lexical categories (e.g. “service” is both a noun and a verb), and different semantical expressions (e.g. the noun “service” can

well, it  
isn't really  
applied to  
the senti-  
ment anal-  
ysis

both refer to assistance and tableware). The number of taggings can thus be large, but is always finite.

The set of lexical and phrasal categories,  $\Gamma$ , are of a somewhat advanced structure in the CCG presented, since it follows recent work by Baldridge and Kruijff [2003] to incorporate *modalities*. A category is either *primitive* or *compound*. The set of primitive categories,  $\Gamma_{\text{prim}} \subset \Gamma$ , is language dependent and, for the English desired to be covered by this thesis, it consists of  $S$  (sentence),  $NP$  (noun phrase),  $N$  (noun) and  $PP$  (prepositional phrase). Compound categories are recursively defined by the infix operators  $/_{\iota}$  (forward slash) and  $\backslash_{\iota}$  (backward slash), i.e. if  $\alpha$  and  $\beta$  are members of  $\Gamma$ , then so are  $\alpha /_{\iota} \beta$  and  $\alpha \backslash_{\iota} \beta$ . This allows the formation of all other lexical and phrasal categories needed. The operators are left associative, but to avoid confusion inner compound categories are always encapsulated in parentheses throughout this thesis.

The basic intuitive interpretation of  $\alpha /_{\iota} \beta$  and  $\alpha \backslash_{\iota} \beta$  is as a function that takes a category  $\beta$  as argument and yields a result of category  $\alpha$ . Thus the argument is always stated on the right side of the operators, and the result on the left. The operator determines the dictionality of the application, i.e. *where* the argument should appear relative to the function: the forward operator ( $/_{\iota}$ ) denotes that the argument must appear on the right of the function, whereas the backward operator ( $\backslash_{\iota}$ ) denotes that the argument must appear on the left. The subscript,  $\iota$ , denotes the *modality* of the operator, which is a member of a finite set of modalities  $\mathcal{M}$  and will be utilized to restrict acceptance in the next section.

The syntactic categories constitutes a type system for the semantic expressions, with a set of primitive types  $\{\tau_x \mid x \in \Gamma_{\text{prim}}\}$ . Thus, if a lexicon entry has category  $(S \backslash_{\iota} NP) /_{\iota} (S \backslash_{\iota} NP)$  then the associated semantic expression must honor this, and have type  $(\tau_{NP} \rightarrow \tau_S) \rightarrow \tau_{NP} \rightarrow \tau_S$  ( $\rightarrow$  is right assosiative). This is a result of the *Principle of Categorical Type Transparency* [?, Montague, 1974]. For now it is sufficient to describe the set of semantic expressions,  $\Lambda$ , as the set of *simply-typed*  $\lambda$ -expressions,  $\Lambda'$ , cf. definition 3.2. In chapter 4 this is extended to support the desired sentiment analysis.

**DEFINITION 3.2** The set of  $\lambda$ -expressions,  $\Lambda'$ , is defined recursively as the set of expressions  $e$ , where  $e$  is either a variable  $x$  from an infinite set of variables  $V = \{v_1, v_2, \dots\}$ , a function abstraction, or a functional application. TODO

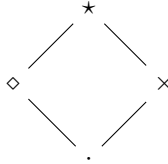
$$\begin{array}{lll}
 x \in V & \Rightarrow & x \in \Lambda' & \text{(Variable)} \\
 x \in V, M \in \Lambda' & \Rightarrow & \lambda x.M \in \Lambda' & \text{(Abstraction)} \\
 M \in \Lambda', N \in \Lambda' & \Rightarrow & (MN) \in \Lambda' & \text{(Application)}
 \end{array}$$

### 3.1 Combinatory rules

CCGs can be seen as a deductive proof system where the axioms are members of  $\Gamma \times \Lambda$ . A text  $T \in \Sigma^*$  is accepted as a sentence in the language, if there exists a deductive proof for  $S$ , for some tagging of  $T$ .

The inference rules of the proof system are known as *combinators*, since they take one or more function pairs, in the form of instances of  $\Gamma \times \Lambda$ , and produces new instances from the same set. The combinators determines the expressive power of the grammar. A deep presentation of which rules are *needed*, and thus the linguistic motivation behind this, is out of the scope of this thesis. In the following essential combinators covered by Steedman [2011, chap. 6] are succinctly described, which constitutes a *midely context-sensitive* class grammar. These are the development of the combinatory rules Steedman presented in [2000, chap. 3], however with significant changes with respect to coordinating conjunctions, due to the introduction of modalities on the infix operators.

The set of modalities used,  $\mathcal{M}$ , follows [Baldrige and Kruijff, 2003] and [Steedman, 2011], where  $\mathcal{M} = \{\star, \diamond, \times, \cdot\}$ . The set is partially ordered cf. lattice (3.2).



(3.2)

The basic concept of annotating the infix operators with  $\iota \in \mathcal{M}$ , is to restrict the application of inference rules during deduction in order ensure the soundness of the system. The  $\star$  is the most restrictive, allowing only basic rules,  $\diamond$  allows rules which perserves the word order,  $\times$  allows rules which permutate the word order, and finally  $\cdot$  allows any rule without restrictions. The partial ordering allows the most restrictive categories to also be included in the less restrictive, e.g. any rule that assumes  $\alpha/\diamond\beta$  will also be valid for  $\alpha/\star\beta$ . Since  $\cdot$  permits any rule it is convenient to simply write  $/$  and  $\backslash$  instead of respectively  $/\cdot$  and  $\backslash\cdot$ , i.e. the dot is omitted from these operators.

The simplest combinator is the *functional application*, which simply allows the instances to be used as functions and arguments, as already described. The forward and backward functional application combinator can be formulated as respectively  $(>)$  and  $(<)$ , where  $X$  and  $Y$  are variables ranging over lexical and phrasal categories, and  $f$  and  $a$  are variables ranging over semantical expressions. Since the operators are annotated with  $\star$ , the rules can apply to even the most restrictive categories. For





Besides functional application, CCG also has a set of more restrictive rules, including *functional composition*, defined by the forward and backward functional composition combinators, respectively ( $>_{\mathbf{B}}$ ) and ( $<_{\mathbf{B}}$ ), where  $Z$  likewise is variable ranging over  $\Gamma$ , and  $g$  over  $\Lambda$ .

$$\begin{aligned} X /_{\diamond} Y : f \quad Y /_{\diamond} Z : g &\Rightarrow X /_{\diamond} Z : \lambda a. f(g a) & (>_{\mathbf{B}}) \\ Y \backslash_{\diamond} Z : g \quad X \backslash_{\diamond} Y : f &\Rightarrow X \backslash_{\diamond} Z : \lambda a. f(g a) & (<_{\mathbf{B}}) \end{aligned}$$

Notice that the semantical expression yielded by ( $>_{\mathbf{B}}$ ) and ( $<_{\mathbf{B}}$ ) is equivalent to regular functional composition ( $\circ$ ) of  $f$  and  $g$ , but since  $f \circ g \notin \Lambda$  they need to be written as  $\lambda$ -expressions.

Functional composition is often used in connection with another rule, namely *type-raising*, defined by the forward and backward type-raising combinators, respectively ( $>_{\mathbf{T}}$ ) and ( $<_{\mathbf{T}}$ ), where  $T$  is a variable ranging over categories.

$$\begin{aligned} X : a &\Rightarrow T /_{\iota} (T \backslash_{\iota} X) : \lambda f. f a & (>_{\mathbf{T}}) \\ X : a &\Rightarrow T \backslash_{\iota} (T /_{\iota} X) : \lambda f. f a & (<_{\mathbf{T}}) \end{aligned}$$

Type-raising allows a, often primitive category, to ...  $\iota$  is ... (unpredictable?) and is thus often suppressed. ...

Finish

Re-write: The introduction of function composition into a categorial grammar leads to many kinds of derivational ambiguity that are vacuous in the sense that they do not correspond to semantic ambiguities.

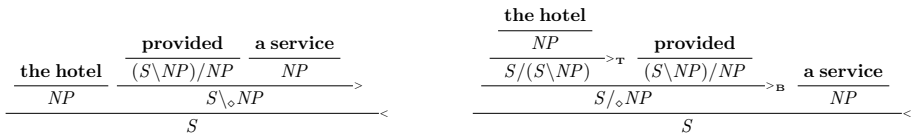


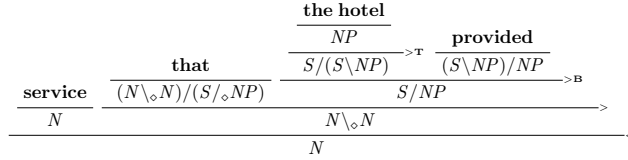
Figure 3.3: Multiple deductions.

A system with these rules demonstrates what is arguably CCG's most unique advantage, namely the ability to handle unbounded dependencies without any additional lexicon entries. For instance a transitive verb, with the *same* category as shown in figure 3.1, can participate in relative clauses as shown in Example 3.1.

$$\mathbf{that} \models (N \backslash_{\diamond} N) / (S /_{\diamond} NP) : \lambda f : (\tau_{NP} \rightarrow \tau_S). \lambda x : \tau_{NP}. (\dots) \quad (\text{Relative pronouns})$$

Figure 3.4: A fragment of a tiny handwritten lexicon TODO.

**EXAMPLE 3.1** Figure 3.5 shows an example of both type-raising and functional composition. The transitive verb (*provided*) is requiring an object in the form of a noun phrase to its right. However, since it participate in a relative clause, its object is given by the noun *that* the clause modifies. Type raising allows the subject of the relative clause to raise into a category that can compose with the verb, and thus allows the relative pronoun (*that*) to bind the relative clause to the noun.



**Figure 3.5:** Deduction of noun phrase with relative clause.

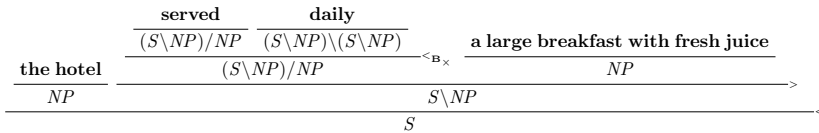
The last set of rules presented here is the *crossed functional composition*, defined by the forward and backward crossed functional composition combinators, respectively ( $>_{B \times}$ ) and ( $<_{B \times}$ ).

$$\begin{array}{ll}
 X /_{\times} Y : f \quad Y \backslash_{\times} Z : g \quad \Rightarrow \quad X \backslash_{\times} Z : \lambda a. f(g a) & (>_{B \times}) \\
 Y /_{\times} Z : g \quad X \backslash_{\times} Y : f \quad \Rightarrow \quad X /_{\times} Z : \lambda a. f(g a) & (<_{B \times})
 \end{array}$$

Crossed functional composition allows *permutation* of the word order. This is useful to allow adverbs in sentences with shifting of heavy noun phrases as shown in Example 3.2.

**EXAMPLE 3.2** Normally an adverb is put after the object of the verb it modifies in English, e.g. “the hotel served breakfast daily”. However if the object of the verb becomes “heavy” it may sometimes be moved to the end of the sentence, e.g. “the hotel served daily a large breakfast with fresh juice”.

In such cases the adverb needs to compose with the verb, before the verb combines with its object. The crossed functional composition allows exactly such structures as shown in figure 3.6.



**Figure 3.6:** Deduction of “heavy” noun phrase shifting.

Steedman [2000; 2011] introduces a few additional combinators to capture even more “exotic” linguistic phenomena. Recollect that the rules are language independent, and indeed some of the additional phenomena covered by Steedman is either considered infrequent (e.g. *parasitic gaps*), or even absent (e.g. *cross-serial dependencies*), from the English language desired to cover by this sentiment analysis. It will later be shown (section ??) that the rules already presented indeed cover a substantial part of English.

## 3.2 Coordination

Coordination by appearance of a coordinating conjunction, such as *and*, *or*, *but*, etc., can be modeled simply by the intuition that such should bind two constituents of same syntactic category, but with different semantical expressions, and yield a result also of that category. Some examples of the *and* coordinating conjunction is shown in figure 3.7.

$$\begin{aligned}
 \text{and} &\models (S \backslash_{\star} S) /_{\star} S : \lambda x : \tau_S. \lambda y : \tau_S (\dots) && (\text{Conjunctions}) \\
 \text{and} &\models (N \backslash_{\star} N) /_{\star} N : \lambda x : \tau_N. \lambda y : \tau_N (\dots) \\
 \text{and} &\models (NP \backslash_{\star} NP) /_{\star} NP : \lambda x : \tau_{NP}. \lambda y : \tau_{NP} (\dots) \\
 &\dots
 \end{aligned}$$

**Figure 3.7:** A fragment of a tiny handwritten lexicon TODO.

It now becomes evident, why the modalities are needed, since application of the crossed composition combinators without any restrictions could allow scrambled sentences to be deducted falsely, e.g. figure 3.8.

$$\begin{array}{c}
 \begin{array}{c}
 \text{I} \\
 \hline
 NP
 \end{array}
 \begin{array}{c}
 \text{the service} \\
 \hline
 NP
 \end{array}
 \begin{array}{c}
 \text{enjoyed} \\
 \hline
 (S \backslash NP) / NP
 \end{array}
 \begin{array}{c}
 \text{and} \\
 \hline
 (NP \backslash NP) / NP
 \end{array}
 \begin{array}{c}
 \text{the view} \\
 \hline
 NP
 \end{array}
 \begin{array}{c}
 \xrightarrow{\text{B}_\times} \\
 \hline
 (S \backslash NP) \backslash NP
 \end{array}
 \begin{array}{c}
 \xleftarrow{\text{B}_\times} \\
 \hline
 S \backslash NP
 \end{array}
 \end{array}
 \begin{array}{c}
 \xrightarrow{\text{B}_\times} \\
 \hline
 S
 \end{array}$$

**Figure 3.8:** Without the modalities .

Similar pit-falls are possible if unrestricted application of ( $\text{B}_\times$ ) and ( $\text{B}_\times$ ) was allowed, as shown by Baldridge [2002, chap. 4] for the Turkish language. This justifies the requirement for the modalities Baldridge originally proposed in [2002, chap. 5] and Baldridge and Kruijff presented in a refined version in [2003].

### 3.3 Features and agreement

In order to ensure that the parsed phrases indeed follows correct English grammar, it is not enough to only consider the phrase structure with respect to the word classes. It is also necessary to

As stated it is essential to categorize phrases, however it is important to notice, that each category can have numerous arguments i.a. denoting features that apply. For instance Bob Carpenter [1995] states features for person (e.g. 1st, 2nd, or 3rd), number (e.g. singular or plural), and case (e.g. subject or object). The set of features that may apply is language dependent, for instance most indo-european languages<sup>1</sup> has gender features (English being an exception<sup>2</sup>), but while Danish use common and neuter classes, other languages like German use masculine, feminine, and neuter classes. The important is thus solely that the set of features is always finite – limited by the specific language.

As stated earlier (see section ??), each base category is allowed to carry a finite set of features attached as arguments.

(For example, functional application corresponds to the familiar classical rule of Modus Ponens under this view)

---

[van Eijck and Unger, 2010]

## 3.4 Lexicon acquisition

As mentioned earlier, acquiring a suitable lexicon is crucial in the development of a lexicalized grammar. For small language fragments the lexicon can simply be “handwritten”, as demonstrated in the previous section. However this is obviously not an sane option when the grammar is to accept a large vocabulary and a wide range of sentence structures.

There exists some wide coverage CCG lexicons, most notable *CCGbank*, compiled by Hockenmaier and Steedman [2007] by translating almost the entire Penn Treebank [Marcus *et al.*, 1993], which contains over 4.5 million words. The result is a highly covering lexicon, with some entries having assigned over 100 different lexical categories. Unfortunately these lexicons are often not free to use, and it has not been possible to fund a license.

We expect such an algorithm to calculate a match score, that is a weighted average over several metrics. Given below are methods for calculating scores for some evident metrics.

- Symbolic similarity – at its most basic form we can consider a sample string (i.e. a word from an input text) against the system’s vocabulary using approximate string matching algorithms such as the *Levenshtein distance* as described by [Wagner and Fischer, 1974].
- Pronunciation similarity – it is an valid assumption that many misspellings still share a majority of the pronunciation with the intended word, i.e. they are approximately homophone. Thus comparing the phonetic properties of an sample string with possible matches can in cases correct misspellings. The *Soundex algorithm* by Robert C. Russell and Margaret K. Odell, as described by [Knuth, 1998, p. 391–92], is a simple, yet power full approach for this purpose.

## 3.5 Part of speech tagging

and and is a non-trivial task

### 3.6 Annotating the lexicon

$$\begin{array}{c}
 \text{Service} \\
 \text{NNP} \\
 \hline
 N : \text{Service}_0 \\
 \hline
 NP : \text{Service}_0 \\
 \hline
 S_X / (S_X \backslash NP) : \lambda f.(f \text{ Service}_0) \xrightarrow{>\mathbf{T}}
 \end{array}
 \begin{array}{c}
 \text{was} \\
 \text{VBD} \\
 \hline
 (S_{dcl} \backslash NP) / (S_{adj} \backslash NP) : \lambda x.x \\
 \hline
 S_{dcl} \backslash NP : \lambda x.(x \sim_{14}) \\
 \hline
 S_{dcl} : \text{Service}_{14}
 \end{array}
 \begin{array}{c}
 \text{excellent} \\
 \text{JJ} \\
 \hline
 S_{adj} \backslash NP : \lambda x.(x \sim_{14}) \\
 \hline
 S_{dcl} \backslash NP : \lambda x.(x \sim_{14}) \\
 \hline
 \xrightarrow{>}
 \end{array}$$

## CHAPTER 4

# Semantic networks

---

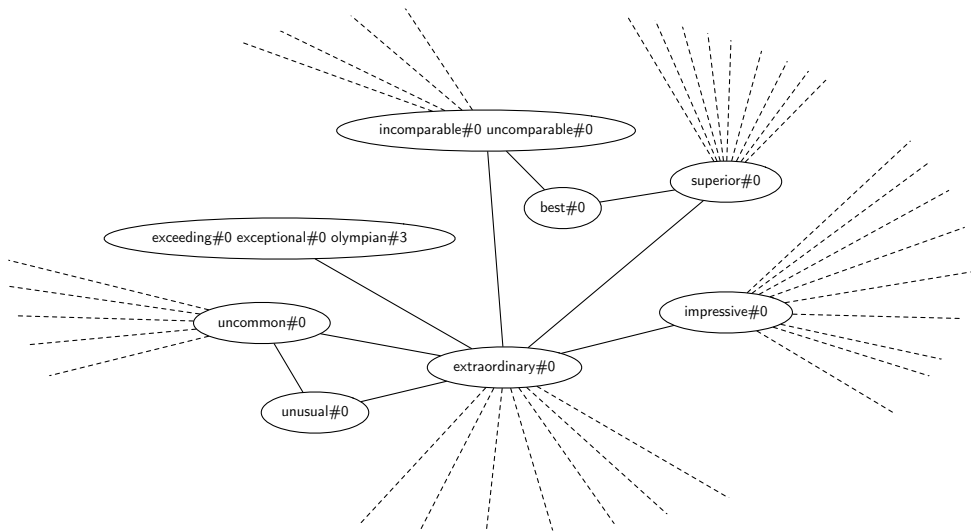
In order to reason about the polarity of the selected subject in the review text, a understanding of the domain of the review is needed. For this purpose the concept of *semantic networks* is introduced. Formally it is defined follows:

**DEFINITION 4.1** a semantic network is a quadruple  $(L, S, R, M)$  where:

- $L$  is the set of lexical units recognized by the network.
- $S$  is the set of *semantic concepts* in the network.
- $R$  is a set of binary relations on  $S$  where the relation  $r \in R$  describes links between semantic concepts, i.e.  $r \subset S \times S$ .
- $M$  is a mapping from lexical units to a set of semantic entities that the lexical unit can entail, i.e.  $M : L \rightarrow \mathcal{P}(S)$ .

Notice that  $S$  and  $R$  constitutes a set of graphs, i.e. for each relation  $r \in R$  the graph  $(S, r)$ . The graph is undirected if  $r$  is *symmetric*, and directed if  $r$  is *asymmetric*.

[Russell and Norvig, 2009, p. 453-456, 468, 471]



An illustrative example of such a graph, denoting a relation in a tiny semantic network of adjective concepts is given in figure X. The relation depicted . transitivity...

We have no way of detecting different semantic ambiguity, since we have no knowledge bade ... therefore we shall restrict our-self to



## CHAPTER 5

# Implementation

---

It was chosen to use the functional programming language *Haskell* for implementing the *proof of concept* program. In the following sections key aspects of the implementation will be presented. For the complete source code for the implementation please see appendix ??.

The reason Haskell, specifically the *Glasgow Haskell Compiler*, was chosen as programming language and platform, was the ability to ...

## 5.1 WordNet interface

The interface used to lookup data in WordNet is ... yada yada

However the interface is not complete ... the function for calculating the closure of synsets  $S$  for some relation  $R$  does not terminate if  $R$  is symmetric. Since several semantic relations are symmetric (e.g. synonyms, antonyms)

## CHAPTER 6

# Test results

---



## CHAPTER 7

# Conclusion

---

### 7.1 Discussion

Even though the structure of the sentences is inspected deeply through the syntactic analysis the semantic expression assigned to adjectives and adverbs is still atomic, and based on the key concepts the context specific positive and negative lists. Even though the lists are context specific, a concept can have ... løses dette evt. af subject-focus?

Since adjectives and adverbs are always reduced to only contribute to the polarity, they cannot be used to identify subjects. E.g. what do you think about the white iPhone vs. the black? (need better ex.!)



## APPENDIX A

# Appendix A

---

## A.1 A naive attempt for lexicon acquisition

### A.1.1 The Brown Corpus

The Brown Corpus was compiled by Francis and Kucera [1979] by collecting written works printed in United States during the year 1961. The corpus consists of just over one million words taken from 500 American English sample texts, with the intension of covering a highly representative variety of writing styles and sentence structures.

Notable drawbacks of the Brown Corpus include its age, i.e. there are evidently review topics where essential and recurring words used in present day writing was not coined yet or rarely used back 50 years ago. For instance does the Brown Corpus not recognize the words *internet*, *hotspot*

sentences will containing words has found it's way into comon that

Other corpora has been considered

## A.2 Tokenizer and tagger

The tokenizer has a very simple task, namely to convert an input string to a list of tokens (lower case words) that represent the symbols of the language. An example of the transformation is shown in (A.1).

“Put the pyramid onto the table.”  $\rightarrow$  [put, the, pyramid, onto, the, table] (A.1)

### A.2.1 Shift-reduce parser

## A.3 Find a good title

The initial attempt is simply to construct a parser that

There are three basic ways to build a shift-reduce parser. Full LR(1) (the ‘L’ is the direction in which the input is scanned, the ‘R’ is the way in which the parse is built, and the ‘1’ is the number of tokens of lookahead) generates a parser with many states, and is therefore large and slow. SLR(1) (simple LR(1)) is a cut-down version of LR(1) which generates parsers with roughly one-tenth as many states, but lacks the power to parse many grammars (it finds conflicts in grammars which have none under LR(1)).

LALR(1) (look-ahead LR(1)), the method used by Happy and yacc, is tradeoff between the two. An LALR(1) parser has the same number of states as an SLR(1) parser, but it uses a more complex method to calculate the lookahead tokens that are valid at each point, and resolves many of the conflicts that SLR(1) finds. However, there may still be conflicts in an LALR(1) parser that wouldn’t be there with full LR(1).

The state  $S_\tau$  ...

Formally a rule  $\mathcal{R}_\tau$ , for the state type  $\tau$ , is a transformation from a state  $s \in \mathcal{S}_\tau$  onto a new set of states  $\mathcal{S}'_\tau \subset \mathcal{S}_\tau$  cf. A.2.

$$\mathcal{R}_\tau : \mathcal{S}_\tau \rightarrow \mathcal{P}(\mathcal{S}_\tau) \quad (\text{A.2})$$

The state type for analysing CCGs is a 2-tuple, where  $P$  is a totally ordered set of ...,



$$\mathcal{S}_{\text{CCG}} : \mathcal{P}(T) \times \mathcal{P}(\mathcal{P}(T))$$

$$\mathcal{R}_{\text{CCG}}^{\text{shift}} \tag{A.3}$$

If all rules in the set is monotone, then the parsing will terminate



# Bibliography

---

- [Baccianella *et al.*, 2010] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), 2010.
- [Baldrige and Kruijff, 2003] Jason Baldrige and Geert-Jan M. Kruijff. Multi-modal combinatory categorial grammar. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 211–218, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [Baldrige, 2002] Jason Baldrige. *Lexically Specified Derivational Control in Combinatory Categorical Grammar*. PhD thesis, University of Edinburgh, 2002.
- [Bresnan *et al.*, 1982] J. Bresnan, R. M. Kaplan, S. Peters, and A. Zaenen. Cross-Serial Dependencies in Dutch. *Linguistic Inquiry*, 13(fall):613–635+, 1982.
- [Esuli and Sebastiani, 2006] Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422. European Language Resources Association (ELRA), 2006.
- [Francis and Kucera, 1979] W. Nelson Francis and Henry Kucera. Brown Corpus Manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.
- [Ganesan *et al.*, 2010] Kavita Ganesan, Cheng Xiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opin-

- ions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*, 2010.
- [Hockenmaier and Steedman, 2007] Julia Hockenmaier and Mark Steedman. CCG-bank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.
- [Hockenmaier *et al.*, 2004] Julia Hockenmaier, Gann Bierner, and Jason Baldridge. Extending the coverage of a ccg system. *Journal of Language and Computation*, 2:165–208, 2004.
- [Joshi *et al.*, 1975] Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163, 1975.
- [Joshi *et al.*, 1990] Aravind K. Joshi, K. Vijay Shanker, and David Weir. *The Convergence Of Mildly Context-Sensitive Grammar Formalisms*, 1990.
- [Knuth, 1998] Donald E. Knuth. *The Art of Computer Programming, vol. 3: Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2nd edition, 1998.
- [Likert, 1932] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.
- [Marcus *et al.*, 1993] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [Padró *et al.*, 2010] Lluís Padró, Samuel Reese, Eneko Agirre, and Aitor Soroa. Semantic Services in FreeLing 2.1: WordNet and UKB. In Pushpak Bhattacharyya, Christiane Fellbaum, and Piek Vossen, editors, *Principles, Construction, and Application of Multilingual Wordnets*, pages 99–105, Mumbai, India, February 2010. Global Wordnet Conference 2010, Narosa Publishing House.
- [Paul and Baker, 1992] Douglas B. Paul and Janet M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.
- [Pollard, 1984] Carl Pollard. *Generalized Context-Free Grammars, Head Grammars and Natural Language*. PhD thesis, Stanford University, 1984.
- [Russell and Norvig, 2009] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.
- [Shieber, 1985] Stuart M. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343, 1985.

- [Steedman, 1998] Mark Steedman. *Categorical Grammar*, 1998.
- [Steedman, 2000] Mark Steedman. *The Syntactic Process*. The MIT Press, 2000.
- [Steedman, 2011] Mark Steedman. *Taking Scope: The Natural Semantics of Quantifiers*. The MIT Press, 2011.
- [van Eijck and Unger, 2010] Jan van Eijck and Christina Unger. *Computational Semantics with Functional Programming*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
- [Vijay-Shanker and Weir, 1994] K. Vijay-Shanker and David J. Weir. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, 27:27–511, 1994.
- [Wagner and Fischer, 1974] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, January 1974.
- [Webster and Kit, 1992] Jonathan J. Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *Proceedings of the 14th conference on Computational linguistics - Volume 4*, COLING '92, pages 1106–1110, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.