

Automatic discovery of constructions in children's speech
Masters thesis

Gideon Borensztajn

UvA, December 2006

Contents

1	Introduction	9
1.1	What are the primitive productive units of speech?	10
1.2	Outline of the project	10
1.2.1	The first phase - Bayesian Model Merging	11
1.2.2	The second phase - Push 'n Pull and comparative quantitative measures	12
2	Theoretical background	15
2.1	Introduction	15
2.2	Generative grammar	16
2.2.1	The assumptions of generative grammar	16
2.2.2	Arguments for the innateness of UG	17
2.2.3	A package deal	18
2.2.4	Language acquisition in generative grammar	18
2.2.5	Critique on generative grammar	19
2.2.6	Semantics and compositionality	20
2.3	Construction grammar	21
2.3.1	Definition of a construction	22
2.3.2	Constructions are pervasive - radical constructionism	22
2.3.3	Cognitive linguistics	23
2.3.4	Implications for a learning theory	24
2.4	Towards a formal constructionist theory of language learning	25
2.4.1	The learning paradox	25
2.4.2	An analysis of the learning paradox	26
2.4.3	Language as a dynamical system	27
2.4.4	The Talking Heads experiment - learning through categorization	27
2.4.5	Fluid Construction Grammar	29
2.4.6	Inducing constructions from CHILDES	30
2.5	Empirical work on language acquisition	31

2.5.1	The work of Ann Peters	31
2.5.2	Usage based grammar	33
2.5.3	Empirical studies aimed at identifying the minimal productive units of speech	34
2.6	Conclusions	35
3	Unsupervised Grammar Induction	37
3.1	Introduction	37
3.2	The Bayesian Approach	39
3.2.1	Probabilistic Context Free Grammars	39
3.2.2	Maximum Likelihood Estimation	40
3.2.3	The Expectation Maximization Algorithm	40
3.3	Bayesian Model Merging - The Stolcke algorithm	41
3.3.1	Initialization, Merging and Chunking	41
3.3.2	Bayesian Learning and MAP hypothesis	42
3.3.3	Priors	43
3.3.4	Structure Prior	43
3.3.5	Parameter Prior	44
3.3.6	Computation of Likelihood	44
3.3.7	Search Algorithm	46
3.3.8	Implementation	46
3.4	e-GRIDS	46
3.4.1	GDL in e-grids	47
3.4.2	DDL in e-grids	48
3.4.3	Contribution of the chunking operator to the Description Length	49
3.4.4	Contribution of the merging operator to GDL	49
3.4.5	Contribution of merging operator to DDL	50
3.4.6	Searching in e-Grids	51
3.5	Adaptations of e-Grids	51
3.5.1	Implementation of the Poisson distribution in e-Grids	52
3.5.2	Implementation of Dirichlet prior in e-Grids	52
3.5.3	Correction of ΔDDL from merging for non-uniform distribution	52
3.5.4	Additional improvements of the e-Grids algorithm	54
3.6	Other work on Unsupervised Grammar Induction	55
3.6.1	The CCM model	55
4	Experiments	57
4.1	Materials and evaluation	57
4.1.1	Evaluation	59
4.2	Results on Benchmark Tests	61

4.2.1	Dirichlet prior	62
4.2.2	Dropping the uniform distribution assumption	62
4.3	Results for WSJ-lexical	63
4.4	Results for CHILDES	64
4.5	Follow-up experiments	66
4.5.1	Intermediate evaluation - what goes wrong?	66
4.5.2	Evaluation of the effect of the priors on the parses	68
4.5.3	Testing the Viterbi approximation	69
4.5.4	Description lengths of treebank grammars	70
4.5.5	Running BMM on the treebank grammar	71
4.5.6	Testing the algorithm without the most unlikely sentences	71
4.5.7	Temporal dynamics of the grammar induction algorithm	74
5	Unsupervised labeling	77
5.1	The prosodic bootstrapping hypothesis	78
5.2	A case for prosody-based sentence segmentation for the discovery of constructions and categories	79
5.2.1	Some reflections on the nature of constituency	79
5.2.2	Prosodic bracketing for compatibility with Construction Grammar	80
5.2.3	Evidence for prosodic ‘constituents’ from contractions and from pro-drop languages	81
5.2.4	The two stage language acquisition hypothesis	82
5.2.5	A hypothesis about the function of constituents	82
5.3	The model merging unsupervised labeling algorithm (MMULA)	83
5.3.1	Implementation	84
5.3.2	Constraints on merging	85
5.3.3	Forecasting the GDL gain of a merge	86
5.4	Experimental results	87
5.4.1	Evaluation	87
5.4.2	Results on Benchmark Tests	87
5.4.3	MMULA Results on WSJ	90
5.4.4	Induction from trees with branching factor greater than 2	90
5.4.5	Results on CHILDES	91
5.4.6	Preliminary results with the push ‘n pull algorithm	92
6	General discussion and suggestions for future work	95
A	The philosophy of the Talking Heads experiment	105
A.1	Some concepts in philosophy of language	105
A.2	The philosophical assumptions underlying the Talking Heads experiment	111

B	Learnability and the Gold Theorem	119
C	Developmental stages in UBG	121
D	The Poisson distribution in e-Grids	125
E	Removal of STOP bit from SB3 in e-Grids	127

Acknowledgements

I want to thank my supervisor, Jelle Zuidema, for his fantastic guidance during this at some times not so easy project. Were it not for his confidence and enthusiasm, and the many valuable advises and brain storm sessions, this thesis could never have been written.

Yoav Seginer generously offered to use his unsupervised dependency parser, and was always there to answer my silly questions.

I would also like to express my gratitude to Remko Scha, for his comments on Chomsky, and to Reinhard Blutner for willing to be my co-assessor, despite the massive volume of the thesis, and for recommending Lakoff's book *Philosophy in the Flesh*.

And last but not least to my dear family, who would have rather seen me doing different things, talking with children instead of studying their talk. This thesis is dedicated to my lovely nephews and nieces, Abel, Sam, Ita, Eva, David and Levi. Please never grow up!

Chapter 1

Introduction

This research deals with the intriguing question how language is represented in the mind of the child. This question divides the linguistic community into two major camps.

According to generative grammar [7], which is the mainstream linguistic theory, the primitive building blocks of language are single words. Language comprehension and production follows a process of rule-application over abstract syntactic categories, such as *noun phrase*, *verb phrase*, etc. A similar system of rules and syntactic categories is believed to be represented in every individual's mind.

In contrast, in construction grammar [27], [19] a person's knowledge of language is believed to consist of a structured inventory of constructions, which are stored fragments of one or more words with optional variable slots. Sentences are produced and interpreted by combining constructions.

As for language acquisition, generative grammar predicts that children already have access to an adult grammar, represented by adult rules and syntactic categories, whereas construction grammar predicts that acquisition is gradual, and that the internal representation of constructions evolves during development towards a consensus, through interaction with other language users.

A large body of empirical research seems to be in support of the constructionist views. The research of Ann Peters [48] suggests that children do not initially identify word boundaries, but that they gradually learn how to segment larger fragments of sentences. Also in Tomasello's Usage Based Grammar (UBG) account of language [66] the empirical finding is that the primary units of speech of children in their first stage of language acquisition are not words but complete utterances. Tomasello observes that there are no system-wide categories, but that children's early language is organized entirely around individual verb constructions which are learned in isolation.

Chapter 2 will cover both linguistic theories and how they cope with the question of

acquisition extensively, as well as the empirical work of Tomasello and Peters. Towards the end of the chapter an integrated theory of language and concept learning, in line with construction grammar, will be presented.

1.1 What are the primitive productive units of speech?

It would be revealing if we could identify the smallest building blocks that are responsible for generating all of the child's utterances. If we could find such a minimal set of generative building blocks, the fact whether or not multi-word expressions occur in the set with a high likelihood could possibly shed light on the nature of the linguistic representation.

In fact, such an experiment has been done by Lieven and Tomasello [41]. In their study, target utterances were reconstructed from a set of previously used utterances. However, in their study the elementary, generative constructions were identified manually, which introduces a certain measure of arbitrariness. In the current project we aim at extracting the constructions automatically and in an objective manner.

The goal of the project is thus to develop an automated procedure for identifying the smallest productive units in children's language. This would enable us to evaluate the core assumptions of generative grammar and construction grammar: apart from the question whether or not some of the smallest units are multi-word constructions, this procedure should enable us to look closer into the development of the grammar of the child, by comparing the most frequent constructions found at different stages of the language acquisition process.

1.2 Outline of the project

Our analysis is based on the Adam corpus from CHILDES [43], which is a collection of corpora of recorded children's speech made freely available to researchers through the internet.

Of course, it is not a trivial matter to infer the status of the productive units - unitary or complex - from the child's language, because they are not observable on the surface of the child's recorded speech.

We assume that the child's underlying linguistic representation is in the form of a stochastic tree substitution grammar (STSG). The choice for an STSG is motivated because it can accommodate both the flat rewrite rules of generative grammar, and multi-word (or single-word) constructions. The generative components of an STSG are tree fragments of arbitrary depth, which can be (partly) lexicalized, or abstract. In the latter case the fragments contain variable slots for syntactic categories, making them suitable for representing constructions.

The specific formalism we propose to use is Data Oriented Parsing (DOP) [2], [54]. One of the reasons for working with the DOP-formalism is that there exist well-documented estimators for DOP, many of which were developed at our faculty [57], [72], [73].

Unfortunately, at present there exist no algorithms for direct induction of an STSG from plain text. Therefore the STSG grammar, including a list of the most frequent productive units (tree fragments) used by the child, is induced in two phases. First, a probabilistic context free grammar (PCFG) is induced with Bayesian Model Merging [63] (see the next section). Then, the corpus is parsed by the induced PCFG, enriching it with a syntactic annotation, which is needed for the second phase. In the second phase an STSG is estimated from the annotated corpus, using the push 'n pull algorithm [73]. Issues relating to the STSG induction from an annotated corpus have mostly been solved. Therefore, the current research focuses mainly on the first phase.

1.2.1 The first phase - Bayesian Model Merging

For the task of annotating the Adam corpus an unsupervised grammar induction algorithm must be used, because it is undesirable to assume a priori that the child has an adult grammar, and therefore we cannot use a parser trained on adult language. As to date, the best algorithms known to us that can label a natural language corpus of the size of the Adam corpus are the Stolcke [63] and e-Grids [47] algorithms. A few algorithms exist that are able to detect constituents (brackets) in natural language with reasonable success [37], [3], [56] but neither of them is able to discover syntactic categories, and neither has reported results on a children's speech corpus. We therefore undertook to re-implement and in several ways extend and modify the unsupervised grammar induction algorithms of [63] and [47]. These algorithms employ the framework of Bayesian Model Merging (BMM) [63] to induce a probabilistic context free grammar (PCFG). Hence, we will use the term BMM algorithm to refer to our own implementation.

Chapter 3 deals with the formalism of Bayesian learning, and extensively covers the Stolcke and e-Grids algorithms. By itself, the Bayesian framework can also be viewed as a statistical model for inductive language learning. In particular, an evaluation of different prior distributions may shed light on the kind of a priori knowledge that is to be assumed for learning to be successful.

Chapter 4 discusses the results of the BMM algorithm on benchmark tests, such as OVIS and the Wall Street Journal, and on the Adam corpus. Although the results by themselves are somewhat disappointing, several follow-up experiments give valuable insights into the limitations of the Bayesian Model Merging approach applied to natural language corpora.

Not discouraged by these results, we tried a different approach, where we specialize

the BMM algorithm for label induction alone, and give the phrase structure (the brackets) for free. This turns the algorithm into an unsupervised labeling algorithm, which we call MMULA (Model Merging Unsupervised Labeling Algorithm). The decision to do semi-supervised grammar induction, with given bracketings, was motivated by our understanding that phrase boundaries can be learnt by the child prior to labeling, from prosodic cues, such as pauses and stress patterns.

Chapter 5 first discusses some of the issues involved in prosodic bracketing versus syntactic bracketing, and exposes our idea, that unsupervised label induction with given prosodic bracketing can be taken as a model for language acquisition. Then, it explains in some detail the implementation of MMULA, and discusses experimental results on benchmark tests and on the Adam corpus. The labeling results we achieved with the Adam corpus are, although still not optimal, encouraging enough to continue with the second phase.

1.2.2 The second phase - Push 'n Pull and comparative quantitative measures

In the second phase, a stochastic tree substitution grammar (STSG) is estimated from the annotated corpus using the push 'n pull algorithm [73]. The push 'n pull algorithm estimates the probabilities of the elementary trees of the STSG based on usage frequency. It will therefore produce an ordered list of the most frequently used constructions (elementary trees). The ordered list also includes constructions with one or more open slots, and even completely unlexicalized constructions.

Assuming we have obtained a list of the most frequent constructions at different stages of Adam's development, we propose to evaluate some quantitative measures that allow a comparison between the grammars at the different stages, in order to test the predictions of construction grammar versus generative grammar. Following are some example measures:

- **Fragment size**
The Usage Based Grammar's account of language acquisition predicts that the size of the used constructions will decrease with age. This is due to the fact that the child starts by copying entire utterances from the parents, and only later learns how to break them apart. In contrast, generative grammar predicts that fragment size is constant.
- **Lexicalization**
Construction grammar predicts a gradual increase of abstract constructions with age to capture the generalisation process that occurs in the child's syntax. In contrast, according to generative grammar, abstract rules are in place from the very beginning of syntax.

A measure of ‘lexicality’ of the constructions should be expected to correlate with the ‘abstractness’ of the syntax. The proposed measure is the proportion of unlexicalized fragments in the STSG to (partially) lexicalized fragments.

- Splitting up of fragments

Ann Peters’ work and UBG predict that large constructions are broken apart into smaller constructions. Accordingly, one should be able to test whether fragments in an early grammar can be ‘explained away’ by smaller fragments in a later grammar or vice versa.

At the present stage we have only some preliminary results on the most frequent constructions in the Adam corpus with push ’n pull, and we have not yet been able to implement the quantitative tests. We hope to complete this work in the near future. An analysis of the difficulties that still need to be coped with, and many suggestions for future work will be given in the discussion.

Chapter 2

Theoretical background

2.1 Introduction

This chapter provides some background about the major ideas on how language is represented in the mind. We will present the positions of the two main camps in the discussion: generative grammar and construction grammar.

Founded by Noam Chomsky in the 50's [6], generative grammar is until this day the mainstream linguistic theory. In the past decades many versions of generative grammar have seen the light, the Standard Theory [7], the Government-Binding Theory [10] and the Minimalist Program [12], to name some of them, but for the sake of the present overview we will take the extended standard theory [7] as representative.

Construction grammar emerged as a reaction to the central position that generative grammar gives to syntax (or form) in understanding language. Construction grammar holds that many linguistic phenomena, such as multi-word idioms, cannot be explained by syntax alone, and that all levels of linguistic description must involve an interaction between syntax (form) and semantics (meaning) [26].

We will confront both theories by focusing on language acquisition: it is here that the central questions regarding the nature of the linguistic representation, such as innateness of a grammar, the status of the word and of the syntactic category, can possibly be resolved. A major part of this chapter centers around the question: how is learning of a language possible within each framework? We will discuss the philosophical issues involved in language and concept learning (the learning paradox), and based on a careful analysis of the paradox, propose to revise the traditional views on concepts and knowledge acquisition. Within this new philosophical framework, we will present an integrated theory of language and concept learning, which will be exemplified by the Talking Heads experiment of Luc Steels [58]. This is followed by a short account of one of the formal constructionist theories, Fluid Construction

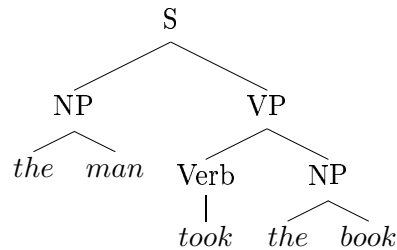


Figure 2.1: Example of a parse tree

Grammar [60]. Towards the end of the chapter, we discuss empirical work on language acquisition and related data-driven theories (e.g. usage-based grammar [66]).

The reader will notice that this overview is not and does not pretend to be completely unbiased, but it is rather supportive of the ideas of construction grammar.

2.2 Generative grammar

In the first place, generative grammar is a formalism that describes the syntactic structure of natural language. Within the framework of generative grammar, the treatment of natural language parallels that of formal logic: generative grammar postulates an autonomous syntax module, which is a computational device that generates well-formed sequences of words, as are the grammatically correct sentences (hence, generative grammar).

The syntax consists of a set of ‘rewrite rules’ (or ‘productions’), which are applied to abstract syntactic categories, such as Noun Phrase (NP), Verb Phrase (VP), etc. A grammatically correct sentence is any sentence that can be obtained by an ordered sequence of such productions, that starts with the START symbol (S), and terminates with lexical items. As such, all grammatical sentences can be generated from the syntax and the lexicon, based on form alone, without the interference of meaning (semantics). As a simple example, consider the sentence *The man took the book* (from [6]). This sentence might be represented by the tree in figure 2.2, where the following rewrite rules were applied, starting from the top of the tree: $S \rightarrow NP VP$, $NP \rightarrow the\ man$, $VP \rightarrow Verb\ NP$, $Verb \rightarrow took$ and $NP \rightarrow the\ book$.

2.2.1 The assumptions of generative grammar

Apart from its merits as a formal model for analyzing language, generative grammar has the ambition to be a theory of human cognition. The assumptions it makes are:

- **Mentalism.**
According to Chomsky, the syntax module has psychological reality, as a mental structure consisting of a system of rules and variables represented in the brain [9]. Chomsky thinks of the language faculty as a specialized mental organ, analogous to other specialized organs, such as the heart or the liver.
- **Modularity.**
The brain has a modular organization, in which each module does its task independently of other modules. In particular, the syntax module, which is responsible for generating grammatical sentences, operates autonomously, and independently of the conceptual system, and of the lexicon, which contains the word meanings.
- **An innate Universal Grammar (UG).**
The main enterprise of generative grammar is to find universal principles (UG), which are shared by all languages in the world. It is believed that the UG is realized in the cognitive structure of language in the human brain, at the core of the language faculty, as some kind of blueprint for the development of language. Quoting Chomsky [7]: ‘as a precondition for learning the child must process an initial prespecification that specifies the form of a grammar of a possible human language’. The UG is assumed to be genetically determined (innate), and unique for humans. The precise content of UG is subject to scientific investigation.

2.2.2 Arguments for the innateness of UG

The usual argument for the innateness of UG is the Poverty of Stimulus (POS) argument [7]: the environment alone does not provide the developing child with sufficient linguistic input to learn the correct grammar. There are some complex linguistic phenomena, such as some specific movement rules involved in forming questions from statements, for which it is difficult to imagine that the child has learned them from experience. Yet, all children succeed to master their language within a limited time span and on the basis of a finite input. To explain that they are successful language learners, it is supposed that the core of the syntax (the UG) must be innate. More so, the PoS argument is believed to be the principle tool for scientific inquiry into the question which components of language pertain to the UG [9].

The other argument for an innate UG is that it explains the existence of many linguistic universals shared across different people and different languages. If these universals are not genetically determined, then how could their universality come about? (The answer we propose will have to wait till section 2.4.4).

2.2.3 A package deal

It is of interest to contemplate on how the various axioms of generative grammar are related. First, since generative grammar treats natural language as a formal language, and since formal languages assume a strict separation between form (syntax) and meaning (semantics), the generative framework is likewise required to postulate a modular design of the language faculty, regardless of any empirical evidence to the contrary.

Second, assuming an autonomous syntax module, operating independently from the sensory and conceptual systems, the only way to explain how rules and primitive categories (like nouns and verbs) enter it, is by postulating them to be innate.

Moreover, using a rigid mathematical formalism to describe language implies that the rules and syntactic categories must be monotonic (their meaning must be fixed and not change over time). Thus, they must exhibit the adult form right from birth.

A final assumption, that linguistic concepts are objective entities, which exist externally to individual minds, as universals, follows necessarily if one denies the possibility that concepts can be induced from experience.

This collection of assumptions may be regarded as a package deal, that goes under the name of ‘rationalism’. Rationalism, usually associated with the French philosopher René Descartes, holds that mind and body belong to separate spheres. The mind is reserved uniquely for humans, while animals are at best sophisticated machines. As a consequence of the separation between body and mind rationalists believe that knowledge cannot be induced through the senses, but must come from universal principles, that are given to each human ‘a priori’ (or innate). The epistemological question (‘how can we know?’) is central in rationalist philosophy, and it will be dealt with extensively in a later section. Chomsky, who reserves a unique role for language in defining what is human, has defended a rationalist position for language in his ‘Cartesian Linguistics’ [8]. In the duality between mind and body, the role of the mind is played by human language, which he argues is creative, and stimulus free, as opposed to animal languages.

It the course of this chapter it will become clear that the opposition between construction grammar and generative grammar is not just about a single aspect of the linguistic theory, but that at the heart of the dispute is a collision between the philosophical schools of rationalism and empiricism.

2.2.4 Language acquisition in generative grammar

If the UG is innate, one may wonder why children don’t have full grammatical competence when they start to speak. Well, according to Chomsky they do. The reason that we don’t hear them produce adult-like grammatical sentences, for instance when

they are in the ‘telegraphic speech’ stage, is not because they lack the knowledge of the grammar, but rather because of other, peripheral restrictions, such as memory and attention limitations.

Quoting Chomsky, the same child in the ‘telegraphic speech’ stage has ‘fully internalized the requisite mental structure, but for some reason lacked the capacity to use it; perhaps he spoke through a filter that passed only content words, perhaps because of limits on memory’ [9], page 6.

According to Chomsky, one must distinguish between competence and performance (or knowledge and capacity). While the linguistic competence is the same for every individual, right from birth, her performance depends on external factors. Therefore, one cannot judge the individual’s knowledge of the grammar by the language she produces alone (a point of view that makes empirical research into knowledge of grammar quite cumbersome).

Although the blueprint of the grammar is prespecified in the UG, quite different grammars may grow from it, at least as many as there are languages in the world. Some versions of generative grammar (e.g. principles and parameters [10]) account for the diversity of grammars in the world’s languages by a process of parameter setting on an innate UG, which constrains the syntax to that of a specific language during child development. The environment (the linguistic experience) plays the role of a trigger that sets the correct parameters, comparable to configuring a software package.

In [9], Chomsky even compares language learning to growing arms - the organism doesn’t learn how to grow arms, but the growth is determined in a precise manner by genetic endowment. In fact, language development really ought to be called ‘language growth’ because the language organ grows like any other body organ. We now know of course that this is not true: the brain, as opposed to most other organs, does not develop according to a blueprint. Environmental factors determine to a large extent the brain’s structure.

The Gold theorem, a theorem about learnability of formal languages by Gold [25], is often invoked to support the innateness claim. Gold proved that the set of regular grammars is not identifiable in the limit by means of positive examples alone. Many linguists supportive of generative grammar infer from this that natural language grammars cannot be learned by experience alone. We’ll discuss the Gold theorem and our objections against its applicability to natural language in appendix B.

2.2.5 Critique on generative grammar

From a neurobiological perspective, many of the claims of generative grammar are quite implausible. The existence of an autonomous syntax module in the brain has never been confirmed. In general, there is no evidence for a strictly modular organization of the brain; on the contrary, all brain areas are densely interconnected.

It is claimed that the development of the language faculty is governed by special principles, specific to this domain. However, it is generally accepted that all parts of the cortex develop according to the same general principles, without domain-specificity. For instance, it has been demonstrated that early in development tissues from the visual cortex can be replaced by tissues from the auditory cortex, which subsequently take on the functions of the former [51].

Recall that the major argument in favor of innateness of a UG is the Poverty of Stimulus argument (PoS). However, the PoS claim has never actually been verified experimentally [50]. In contrast, studies are accumulating indicating that

1. Productivity is less than what should be expected if one assumes an innate grammar to be in place. The experimental facts seem to indicate that productivity revolves around specific items (item-based), and does not generalize to all members of the same category. Tomasello [66] observes that for young children (2-3 years) the scope of the syntactic ‘rules’ is not system-wide, but limited to specific verb constructions (so-called ‘verb islands’), and as a result their productivity is limited likewise. (These and similar issues will be discussed later in more detail.)
2. Many aspects of language acquisition can be explained on the basis of the observed input [50]. In a dense corpus study Lievens et al. [41] showed that 74 % of the target utterances produced in one day by a 2 year old child could be reduced to utterances produced in the previous 6 weeks by using a single combinatorial operation. Converging evidence comes from neural network simulations, which have demonstrated that what looks superficially as rule-like behavior can be explained on the basis of induction from examples alone, for example in the case of the past tense of English [45]

2.2.6 Semantics and compositionality

A last point of critique is that generative grammar doesn’t offer a theory of meaning. There is a meaning (semantics) associated with the entries in the lexicon, which is carried over to the syntax (so-called lexical insertion), but there is no prescription for how to construct the meaning of the sentence.

In formal languages, the semantics of a sentence is constructed in parallel with the syntax, according to the so-called principle of compositionality:

For every complex expression in a language, its meaning is determined completely by the meanings of the constituents of the expression and the syntactic structure.

Although one would expect that generative grammar, being a formal language, has no other option than to adopt the compositionality principle, in fact compositional

semantics never entered the theory of generative grammar. Mainstream generative grammar has always refrained from dealing with semantics, as it maintained that it is possible to study language solely on the basis of grammatical form, and independently of semantics.

However, many linguistic findings (some of which will be discussed in the next section) seem to support the conclusion that syntax is not the only generative component. Semantics seems to have its own primitives, that do not always correspond to the syntactic primitives. Therefore, influential linguists such as Jackendoff have come to see semantics ‘not as derived from syntax, but as an independent generative system correlated with syntax through an interface’ [33], p124.

In sum, the critics claim that a theory of language cannot be based on syntax alone, but must be integrated with a theory of meaning. And as a consequence, also language acquisition is integrated with concept acquisition.

This only scratches the surface of the critique on generative grammar. It was merely our intention to sketch the theoretic background on which construction grammar emerged, before we introduce this major alternative linguistic theory. We will see that construction grammar rejects most of the assumptions of generative grammar. In construction grammar

- There is no modular design, syntax and lexicon are integrated by means of constructions, which incorporate both rules and syntactic categories.
- Constructions (which replace rules and categories) are not fixed and innate, but dynamically changing and learned incrementally.
- Meaning (of words and concepts) is not objective, and does not exist independently of individuals, but it is constructed by the individuals, in parallel with language.

2.3 Construction grammar

The research agenda of construction grammar was motivated by the observation that many linguistic expressions, and their semantics, cannot be explained on the basis of syntax alone, suggesting that a linguistic theory is needed, which, unlike generative grammar, can associate a semantic (and pragmatic) interpretation with configurations larger than single words. A large body of empirical and theoretical linguistic research, united under the title ‘construction grammar’ [27], [19], [14] has lead to a reformulation of linguistic theory.

From a different point of interest, developmental psychologists such as Tomasello, studying language acquisition in children, were unhappy with the framework offered

by generative grammar, as they found that the description of child language in terms of adult units, rules and categories was inadequate. This led to empirically based models of language acquisition, such as ‘usage based grammar’ [66], and the model of Ann Peters [48]. These will be discussed in detail in sections 2.5.1 and 2.5.2.

2.3.1 Definition of a construction

In construction grammar the primitive productive units of language are constructions. Goldberg defines a construction as a form-meaning pair (an association between a semantic frame and a syntactic pattern), for which the meaning or form is not strictly predictable from its component parts, that is, non-compositional [26].

For example, the meaning of the construction *kick the bucket* (to die) cannot be predicted from the meanings of the constituent words. At the same time, the form of the construction shows idiosyncratic behavior: the passive form - *the bucket was kicked by John* is unacceptable.

Another example is the construction *the more you think about it, the less you understand*, used by [26] to illustrate that the conventional apparatus of rewrite rules is inadequate to describe many linguistic phenomena. In this case, not only the semantics, but also the form of the construction cannot be predicted from its components within the framework of generative grammar: First, phrases starting with ‘the’ are normally classified as noun phrases, but in this case they are neither noun phrases nor clauses. Second, there does not exist a phrase structure rule which juxtaposes two phrases without a conjunction in between. Hence, according to the above definition, this expression is a candidate for a construction.

Constructionists believe that syntax is not the only generative system, but that semantics and pragmatics associate with the expression at different levels independently of syntax. Fillmore et al. [20] demonstrated this in a classical case study of the expression *Let alone* (as in *John won't touch the food, let alone eat it*). Fillmore's study showed that syntax and semantics cannot be separated, because they interact at many levels. According to Fillmore, *Let alone* is paradigmatic for the majority of expressions in language, the meanings (and pragmatics) of which cannot be accounted for by the principles of compositional semantics.

2.3.2 Constructions are pervasive - radical constructionism

Mainstream generative grammar has argued that constructions exist only in the periphery of language (e.g. *kick the bucket*), and that therefore they need not be the focus of a linguistic or learning theory. But, in fact constructions of varying degree of complexity and abstractness are pervasive in language. On one side of the spectrum there are completely lexicalized constructions, e.g. *all of a sudden, by and large, so*

far so good. Other constructions, such as *the more you think about it, the less you understand*, have one or more open ‘slots’, making them productive. For all these the meaning cannot be produced on-line by combining the meanings of the parts, and thus the language learner must memorize and store the construction as a whole, together with its associated meaning.

Further on the scale are constructions that are composed entirely of free variables. An example (from [33]) is the resultative construction, as in *Clyde cooked the pot black* and *Hank drove his engine clean*. The resultative construction has an associated semantics that means *cause NP to become AP by V-ing with it*, where all the constituents are free variables. The construction must be stored as a whole in the lexicon, because its meaning is not predictable from the components: normally, one cannot cook a pot, but one uses a pot for cooking.

Thus, even completely abstract, unlexicalized constructions are stored in the lexicon. As a consequence, the lexicon contains many redundant entries for words on different levels of description [33]. In this manner the distinction between lexical items and what have traditionally been regarded as rewrite rules of grammar begins to blur: a phrase structure rule is just another item in the lexicon with variables in it.

In radical constructionism [14] it is believed that all syntactic patterns in language are in fact constructions, meaning that there are no syntactic patterns that are completely compositional. For example, the transitive expression *John sneezed the napkin off the desk* assigns a semantics to the sentence, which is not licensed by composing the meaning of *sneeze* with the meanings of the other parts. The part of the meaning which indicates movement is contributed by the transitive construction itself.

Goldberg [26] showed that even for a simple construction as *Mary sent the book to John* the semantics at the construction level are different than the semantics of *Mary sent John the book*, even though all the constituents are the same: In the latter example an inanimate indirect object is unacceptable, for example *Mary sent storage the book*, while in the former, e.g. *Mary sent the book to storage* it is acceptable. Also, in the passive construction (*John was bit by a dog*) a certain semantic perspective on the scene is associated with the whole construction and not with the components.

In sum, according to radical construction grammar our knowledge of language consists of a structured inventory of constructions with associated meanings (a *constructicon*). In construction grammar syntax and semantics are not separated, but unified in the lexicon through constructions.

2.3.3 Cognitive linguistics

Cognitive linguistics [39] aims at combining the ideas of construction grammar with the neurobiological facts and with cognitive science. According to the theory of cognitive linguistics the function of grammar is to associate mental concepts with lexical

structures in the brain. The grammar of a language is a neural system which, far from being an autonomous module, consists of neural connections linking the bodily-grounded conceptual and the expressive (phonological) systems of the brain [40]. This is the anti-thesis of the autonomous syntax module of generative grammar: there are no autonomous syntactic primitives at all. On the conceptual level, the links are represented by a collection of constructions, which are pairings between phonological categories and conceptual categories.

Constructions have a conceptual pole and a lexical pole. The former is a conceptual frame, or an image schema. These are not directly observable, but studied within the context of frame semantics [18], a discussion of which falls outside the scope of this thesis. The lexical or grammatical pole consists of lexical or syntactic patterns, which can be directly observed from the language. See figure 2.2 for an example.

Semantic:	TRANSFER-TARGET	<Agent>	<Patient>	<Target>
	Slide	<slide-1>	<slide-2>	<slide-3>
Syntactic:	Predicate	Subject	Dir-Object	'to' Target

Figure 2.2: A construction relates a syntactic pattern with a semantic frame

Both conceptual and syntactic categories are prototypical, in contrast to the classical definition of syntactic categories in generative grammar. For example the prototypical category of *nouns* is associated with a prototypical conceptual category of *things* centered around bounded physical objects. This ensures that the categories can be learned from exemplars in an incremental fashion (non-monotonic). Learning a grammar amounts to learning associations between forms and meanings, which are constructions.

2.3.4 Implications for a learning theory

Since by definition the meaning of a construction cannot be computed from its parts, constructions must be learned as a whole. It is therefore perfectly conceivable that the meaning of a part of a construction (e.g. a single word) is acquired after the meaning of the entire construction is already known. This exemplifies how meaning can be acquired by differentiation (the meaning of the parts is differentiated from the meaning of the whole), decompositionally, rather than compositionally.

If language learning consists of the acquisition of an inventory of constructions of varying degrees of complexity and abstractness, it is easier to conceive how children can get from 'here' to 'there'. In contrast to the all-or-nothing learning of generative grammar, construction grammar allows for incremental learning by enabling children to follow a route from simple constructions to complex and abstract constructions. The sections on empirical work on language acquisition (2.5) illustrate this approach.

2.4 Towards a formal constructionist theory of language learning

What should a theory of learning look like? Before we sketch a proposal for a learning theory along the lines of construction grammar, we will try to analyze the fundamental reason why, in our view, formal language theories of learning (such as proposed in generative grammar) are doomed to fail. The reason is the so-called ‘learning paradox’. The learning paradox sharply illustrates the inadequateness of invoking a theory borrowed from formal logic to explain cognitive phenomena such as learning. A solution of the learning paradox requires a complete revision of the way we think about language, concepts, and knowledge. Equipped with these new insights, we will be present a constructionist theory of learning in section 2.4.5.

2.4.1 The learning paradox

In our view, learning theories based on formal languages, such as generative grammar, suffer from a fundamental and in principle unsolvable logical problem, that is concerned with learning formal concepts: the so-called learning paradox.

We assume that learning a new concept consists of defining it in terms of existing concepts, because the new concept must be incorporated into the existing conceptual framework. In turn, the previously known concepts must be defined in terms of even more previously known concepts, and so on. Eventually we arrive at a point where the primitive concepts cannot be learned, because they are the elementary building blocks for all other concepts. Therefore those primitive concepts must be innate. In other words, the learning paradox says that, in formal languages, learning new meanings can only proceed in a compositional fashion, on top of previously known meanings. In Fodor’s formulation [21]:

Concept learning should be hypothesis testing and confirmation. The hypothesis must be formulated in terms of the existing concepts in the conceptual system.

We cannot formulate a hypothesis for a primitive concept, unless we use other concepts in the hypothesis. But the latter would be a circular definition.

Taking this to the extreme, Fodor concludes that, since all morphologically simple lexical concepts are monadic (have no parts), the meanings of all simple words must be innate [22].

The relevance of the learning paradox for language learning is in its application to the acquisition of syntactic categories and rules, as well as lexical concepts. When

Chomsky argues against the possibility of learning (the syntactic categories of) a grammar by induction, he formulates this in terms of the learning paradox [11]:

Induction cannot create abstractions since to recognize similarities among different exemplars, a child must already have the abstraction *a priori*, that is, innately.

2.4.2 An analysis of the learning paradox

The core of the paradox is in its use of the classical definition of a concept, which has its origin in the Fregean ‘sense’ [23]. In Frege’s (and Fodor’s) interpretation a concept is objective and universal, monotonic (its meaning is fixed and unchanging) and independent of individual minds.

However, this is not the same ‘psychological’ concept that cognitive scientists envision when they inquire into learning. To solve the learning paradox, one needs to give up on classical concepts. One must concede that concepts do not pre-exist in the external world independent of and external to individual minds, as is implied by the so-called ‘correspondence theory of meaning’. Instead, concepts are constructed by every individual autonomously. These ideas challenge the traditional epistemology (theory of knowledge) associated with the rationalist school: not only concepts, but in general knowledge of the world is constructed by the individual, rather than passively learned, and in particular knowledge of a language, or grammar. This view is commonly known as ‘conceptualism’.

Cognitive linguistics assumes prototypical concepts, constructions, which are non-monotonic, and grounded in the sensorimotor system (embodied) [40]. Prototypes are characterized by family resemblance, typicality effects, classification of new instances based on a distance function, and incremental learning of a concept from exemplars [53]. Prototypical concepts can thus be induced by experience. An interpretation of concepts as prototypes allows for incremental and non-monotonic learning (in which the concepts gradually change their meaning), and doesn’t suffer from the assumptions made in the learning paradox (how this works exactly will be explained in section 2.4.4). Obviously, linguists in the generativist tradition are unwilling to accept prototypical concepts, because the classical definition of concepts is fundamental to the formal logic framework of generative grammar.

In Chomsky’s earlier argument for the innateness of syntactic categories, the implicit assumption is that linguistic concepts are classical and monotonic, and syntactic categories are assumed to have an external and objective existence. Instead, if we take linguistic concepts (and the associated meanings) to be prototypical, the conclusion of innate syntactic categories is no longer justified.

With a prototypical view on concepts children can no longer be assumed to have the same syntactic categories as adults. Generative theories of language acquisition

are based on the so-called ‘continuity assumption’, which permits them to describe young children’s language with adult-like grammars. However, in view of the above philosophical considerations, and accumulating empirical evidence [65], we believe that the continuity assumption is not justified.

2.4.3 Language as a dynamical system

The issues sketched above, together with evolutionary considerations, suggest quite a different picture of natural language than the generative grammar programme. In generative grammar, language is seen as a static system. There is only one perfect and ideal language, which all competent speakers speak, so the language community is assumed to be homogeneous. All words have a fixed meaning; all syntactic categories and rules are fixed (monotonic). Learning a language amounts to learning the ideal, objective language, in a passive way, by identification. Theorems about language learning, such as Gold’s theorem, treat language as such a formal system.

An alternative perspective on language is gaining momentum, in particular among researchers modeling the evolution of language and artificial communication systems [58], [59], [35]. In this view, language is a dynamic and adaptive system, like an ecology. It lives, in the form of constructions, as a kind of parasite (meme) on a language community, independent of any single individual speaker. No single individual language user is in control of (nor has an overview of) the ideal language. In contrast, in order to study natural language, one must consider the entire population, because language is a phenomenon of a population, not of an individual. Language is in constant flux and evolution, word forms and meanings constantly change. Learning is seen as participating in the dynamic system.

These ideas on language were put into practice in the Talking Heads experiment. The theory of learning that is proposed in the Talking Heads experiment incorporates the ‘conceptualist’ epistemology, discussed above. Moreover, it shows how learning with non-monotonic, prototypical concepts is feasible, and how this solves the learning paradox. The interested reader may find an additional discussion of the philosophical background of the Talking Heads experiment in the appendix. The following section may be skipped without losing the main course of argument.

2.4.4 The Talking Heads experiment - learning through categorization

In the Talking Heads experiment of Luc Steels [58], [62] the above principles have been implemented to demonstrate the feasibility of bootstrapping a communication system among artificial agents situated in a real world situation. In the experiment, the agents, embodied in robots with moving camera’s, conceptualize the perceived

world, invent and learn a shared language, and succeed in communicating what they see to each other in that language. In the simulation the robot agents communicate by playing language games: one agent describes a shape on the whiteboard with a word, and the other agent must guess which shape was intended.

If in the classical approach learning is compositional, that is, the meaning of the parts must be acquired before the meaning of the whole, the theory of learning advocated by cognitive linguistics [39], and implemented by Steels in the Talking Heads experiment, should be called decompositional.

Learning is conceived of as categorization. Categories are constructed by the individual learner by discretizing the continuous sensory input from the environment; the categories represent the individual's internal conceptualization of the world. Categories are fine-tuned to the environment if this leads to increased success in a task (for example successful communication). Initially, there are only a few very crude categories (or concepts); in the learning process more specialized categories are differentiated out of the crude categories. Thus, the agent conceptualizes the world in an incremental fashion, through a process of differentiation of categories.

Acknowledging that neither a nativist theory of learning (e.g. parameter setting), nor a purely empiricist (inductive) theory of learning can be maintained, Steels [58] endorses a biologically inspired, selectionist theory of learning categories. The principles of selectionism entail an interaction between the individual and the environment for controlling the growth of categories. Categories are neither innate nor induced, but grown from inside and are pruned by external factors. While the individual is responsible for growing new categories out of existing categories through random variation and invention, the environment exerts a 'selectionist force' that causes some categories (those that happen to be useful for characterizing the environmental input) to be strengthened and others to be pruned. In this way, the conceptual system of the individual conforms to the environmental input.

At the same time that the agents conceptualize the world, they lexicalize their concepts by giving the names: agents either invent new words or learn words from other agents. There is a bi-directional interaction between conceptualization and lexicalization: if an agent chooses a certain word to communicate a concept to another agent, and if communication is successful, then the internal association between the concept and the word is strengthened. Reversely, if the hearing agent interprets a word successfully, by inferring a concept from his own repertoire and applying it to the scene, this will strengthen the agent's current conceptualization of the scene. So we see that language and concepts interact. Language plays a role in determining the meanings of concepts, consistent with the Sapir-Whorff hypothesis [4]. Language is believed to co-evolve with conceptual development.

A coordination process between the agents guarantees that shared concepts and a shared lexicon emerge in the language community (for details see [58] and the appen-

dix). This is how universal categories are explained without assuming innateness.

Although the Talking Heads experiment doesn't deal with syntax, but only with the acquisition of a lexicon, its conclusions can be generalized for the acquisition of syntactic categories. This implies that syntactic categories are incrementally fine-tuned to the linguistic input received from other agents, until eventually they converge to 'universal' categories by a process of coordination between agents. Note the difference with the generative, 'compositional' account of learning. In the latter, fixed, objective and universal syntactic categories pre-exist in the world and have to be learned to perfection. Here, each individual autonomously constructs his own syntactic categories, and incrementally adjusts them to a consensus shared by the group.

Hopefully, the parallels between the Talking Heads experiment and construction grammar have become clear by now. As in construction grammar, the Talking Heads experiment shows that conceptual and linguistic development are immensely entangled. This implies that any language acquisition theory must incorporate both syntax and semantics. As in construction grammar, the Talking Heads experiment shows that it is essential for the success of the learning process that categories are prototypical, and that they can be learned by experience.

How the ideas of the Talking Heads can be formalized into a full-fledged constructivist linguistic theory, which includes syntax, is the subject of the next section. We chose to treat the formalism of Fluid Construction Grammar (FCG) [60], although there exist other formalisms for construction grammar, such as Embodied Construction Grammar (EBG) [1].

2.4.5 Fluid Construction Grammar

In [61] Luc Steels argues that the primary function of grammar, apart from contributing to the meaning of the sentence, is to reduce the possible interpretations of the sentence. Here, an interpretation is a unique assignment of each of the word meanings in the sentence to an object in the scene. If a sentence would be grammar-less then every word meaning in the sentence could in principle be mapped onto every object in the scene, resulting in an explosion of possible interpretations. According to Steels, the function of grammar is to group words together (for example, by means of grammatical markings and word order), resulting in the binding of variables on the semantic pole of the expression.

For example, in the sentence *the red ball bounces on the table* there are 2 objects in the scene, which can be bound to any of the words *red*, *ball*, and *table*. This means that there are 8 possible interpretations. By grouping the words *red* and *ball* into a noun phrases (NP) the grammar informs that *red* and *ball* refer to the same object, thereby reducing the number of variables to be resolved by one, and the number of interpretations by a factor 2.

These ideas are implemented in Fluid Construction Grammar (FCG) [60]. A construction is defined here, as in section 2.3.3, as an association or mapping between a grammatical pole (syntactic structure) and a conceptual pole (semantic structure). In its simplest form a construction maps a single word form to a single meaning. A more complex construction associates relations in the form domain with relations in the meaning domain. In FCG parsing and production are defined as constraint propagation. Linguistic competence or knowledge is formulated as a set of constraints (captured by the constructions) linking syntactic and semantic aspects of the sentence structure. Parsing (and production) is performed by successive application of appropriate constructions.

construction Throw-Transitive
constituents: t1 of type animate object t2 of type throw t3 of type inanimate object
grammatical constraints (on word order): $precedes(t1_f, t1_f)$ $precedes(t2_f, t3_f)$
semantic constraints (on argument structure): $t2_m(thrower) = t1_m$ $t2_m(throwee) = t3_m$

Figure 2.3: Throw-Transitive construction

The example of the *Throw-Transitive* construction (*Sue throws the ball*) in figure 2.3 (adapted from [5]) will make this clear. In the figure, the subscripts f and m denote the *form* (grammatical pole) respectively the *meaning* (semantic pole) of the construction. The construction in figure 2.3 translates grammatical constraints (here: word order) into semantic constraints (here: argument structure) and vice versa. Specifically, the precedence relation between $t1$, $t2$ and $t3$ is translated to θ -role bindings: $t1$ becomes the *agent* of the verb $t2$, and $t3$ becomes the *patient*. Parsing and producing a sentence thus amounts to constraint propagation, from the grammatical to the semantic domain and vice versa. This contrasts with the generative grammar view on parsing, where parsing occurs only in the syntactic domain (since the syntax is believed to be autonomous).

2.4.6 Inducing constructions from CHILDES

In construction grammar learning can be formulated in terms of learning the associations between forms and meanings (constructions), rather than in terms of abstract

rules of an autonomous grammar. Chang [5] proposes a model of language learning where constructions are induced from utterance-situation pairs, presented to the learning algorithm. It is assumed that a conceptual representation of the world (an ontology) is largely in place by the time children begin to speak the language. This prior knowledge is presented to the algorithm in the form of semantic descriptions based on frames [18]. Since the semantic frames are not available from CHILDES, semantic annotations are added manually to the sentences.

According to [5], the learning problem can be summarized as follows:

Given an ontology of previously learned concepts, an initial set of constructions C labeling some of those concepts, and a set of new utterance-situation training examples, find the best set of constructions C' to fit the seen data and generalize to new data.

The implementation makes use of a hill-climbing heuristic, and uses Bayesian learning and the Minimum Description Length principle to find the optimal grammar (set of constructions).

In our work, as in Chang's, we attempt to induce constructions from CHILDES within a Bayesian framework. However, in the present research we do not have semantic annotations to our disposition, and therefore we try to discover the constructions from the text alone. The semantic poles of constructions are ignored in our representation of a construction within the formalism of STSG.

2.5 Empirical work on language acquisition

A major source of motivation for construction grammar comes from the observation of language acquisition in children. Some developmental psychologists find that they cannot adequately describe linguistic development in terms of adult words and categories. The constructionist framework allows for the necessary flexibility in hypothesizing an incremental path from child to adult language, which is necessary for a theory of learning. Here we will sketch some of the empirical research that has been done on language acquisition.

2.5.1 The work of Ann Peters

Ann Peters [48] argues that linguists should not take for granted that first-language acquisition can be adequately described using the adult linguistic apparatus, which has the word as the basic unit of speech, and uses adult categories. From the child's point of view the raw data is a stream of speech sounds: out of this stream of unknown structure the child must attempt to capture some meaningful chunks, and preserve

them in the lexicon unanalyzed for later use. The child must also be able to segment the stored raw chunks into smaller units, that can be recombined into productive speech. The lexicon redundantly stores both whole constructions and segments of constructions, or words.

The experimenter must look at the child's production in order to identify the actual units from which the child constructs utterances. Peters gives some criteria that help determine what is a single unit rather than a multi-unit construction for the child. These are, among others:

- is the utterance an idiosyncratic chunk used repeatedly in exactly the same form?
- is the construction or utterance unrelated to any productive pattern in the child's current speech?
- is the use of the utterance inappropriate in the context? or does the usage always occur in a well-defined context?
- is the utterance phonologically coherent, spoken without hesitation?

She goes on developing a qualitative model for language acquisition based on constructions and a memory based account of learning: children typically first extract and remember whole utterances; if necessary they segment the stored constructions, match them to other constructions in order to find shared properties and categories, and fuse them into chunks.

She proposes some strategies that the child might use in the initial extraction problem, when it attempts to extract single units ("holophrases") from the continuous stream:

- the child may extract and store sound sequences that have a clear connection to context
- the child may extract sound sequences that are bounded by silence
- the child may extract sound sequences that are marked by suprasegmentally delimited stretches of speech, or by tone, or rhythm.

The subsequent segmentation strategies include segmenting off at syllable boundaries, or at intonationally salient places. Evaluation of segments is done by frequency and meaningfulness, and by trying out the segment in production. Subsequently, abstract categories and frames (constructions with a variable slot) are formed through a matching and fusing process, similar to the one that will be described later in the context of Bayesian grammar induction.

There is a clear parallel between this construction-based account of learning and the categorization process described in the context of the Talking Heads experiment (see section 2.4.4). In both cases knowledge of the linguistic concepts is acquired by segmenting a continuous environment (in Peters' work: a stream of sounds) into discrete categories. In both cases the child actively constructs the linguistic concepts, and incrementally fine-tunes them as more data comes in.

2.5.2 Usage based grammar

A similar theory of early linguistic and syntactic development has been proposed by Tomasello, based on observational studies, for instance diary studies of his daughter [64], and laboratory experiments. According to Tomasello [66], the children's complete utterances rather than individual words should be taken as their primary units of speech in the first stage of language acquisition. An utterance (or construction) is any meaningful, linguistic act that can vary in complexity and has some communicative intention. The question whether the primary units of speech are complete utterances or single words must be addressed empirically, on the basis of actual language usage, by distributional analysis over corpora of children's speech. This leads to a so-called usage based account of grammar [66].

Linguistic development is preceded by the child's ability for joint attention (at 9-12 months), and his understanding that an utterance is intended as a full communicative act. Tomasello distinguishes four main developmental stages in the child's use of constructions, which become more and more complex over time [68]. These are the 'holophrastic' stage, at around 12 months, where children attempt to reproduce entire utterances by imitation; the 'word combinations' stage, at 18 months, in which children combine words without using any syntax; the 'verb island' stage, at 24 months, in which children learn constructions with one or more open slots, usually built up around verbs; and the 'adult constructions' stage from 3 years old, where children arrive at adult-like categories by abstracting across verb islands, in a process which Tomasello calls 'grammaticalization'. Refer to appendix C for a detailed overview of the developmental stages of UBG.

Verb Islands

The best illustration of the item-based nature of children's early language is given by the verb-islands. Tomasello observed that children's early language is organized entirely around individual verb constructions which are learned in isolation. Each verb is used with its own unique set of inflections and with its own specific constructions irrespective of other verbs. For example, it was found that children know how to use different morphological markings on different verbs, but for any single verb the same

form was used consistently. Tomasello concludes that rules are not initially generalized over verbs, but they are local. This means that the categories with which children are working are not such verb-general things as ‘agent’ and ‘patient’, but rather verb-specific categories as ‘hitter’ and ‘hittee’, ‘breaker’ and ‘something broken’. This of course suggests that syntactic categories and syntactical rules are not innate.

In a corpus study based on the CHILDES corpus, Hodges et al. [31] analyzed the acquisition of the complex construction ‘I Verb (NP) to VP-INF’, as in *I want (you) to play*. They found that learning is asymmetrical and centers around a single verb, *want*, even though other candidate verbs, such as *have*, *got*, *like* and *need* are at the same time known by the child. This, and similar experiments [42] constitute evidence that at this early stage (2.5-4 years) the child’s language is item-based in nature, with virtually no evidence of any system-wide syntactic categories, or rules.

Converging evidence for the item-based nature of verb constructions, and for a limited productivity comes from laboratory experiments with nonce verbs. Tomasello and Brooks [67] conducted an experiment, where 2-3 year old children were trained on novel nonce verbs, which they heard only in an intransitive construction. When encouraged to use the verb transitively, very few children would do so, even though the accompanying picture showed a highly transitive action. In contrast, a control group of 5 year old children readily generalized to the transitive construction. This indicates that 2-3 year olds are not ready to generalize across constructions.

2.5.3 Empirical studies aimed at identifying the minimal productive units of speech

The question of identifying the actual productive units of speech has been addressed in a series of studies which involved distributional analysis on children’s corpora [41], [42]. The general method is to trace back the sources of creativity of the child’s speech. The experimenter tries to relate a target utterance to what the child has said before in order to assess the possibility that the utterance is rote learned or (partially) constructed from smaller lexical items. If most of the utterances produced by the child can be reconstructed from earlier constructions larger than a single word this would favor the hypothesis that the smallest psycholinguistic units are multi-word constructions.

Lieven et al. [41] recorded the speech output and input of a 2 year old child, during 6 weeks, for five days a week, and one hour a day (a so-called dense corpus study). The utterances of the last day were set as target utterances. For each target utterance they searched to find the closest matching utterance produced by the child in the preceding weeks. The main criterion used for defining an utterance as the closest match was the number of consecutive morphemes that it had in common with the target utterance.

Having found the closest match, the ways in which the novel utterance differed from it were analyzed. In particular, the number of operations needed to arrive from

the closest match to the target utterance was determined (operations were ‘substitute’, ‘add’, ‘drop’, ‘insert’ and ‘rearrange’). In some cases more than one prior utterance matched the target utterance in the same way, with variation in the same position. This was then identified as a construction with a slot. It was found that only 37 % of the target utterances had not been said before in their entirety. Of these novel target utterances 74 % could be composed from previous utterances with a single combinatorial operation.

These results show that

1. productivity is fairly limited, and less than should be expected if one assumes the existence of abstract, system-wide rules as in generative grammar
2. the vast majority of the produced utterances can be reconstructed from multi-word constructions, providing evidence for the psychological reality of multi-word constructions.

2.6 Conclusions

We have surveyed some of the major ideas about language acquisition. We have tried to convince you that construction grammar currently offers the best explanation for language acquisition. Yet, tools are lacking to systematically inquire into the acquisition of constructions. Formal models for construction grammar are being developed [60], [5] but no implementation exists yet that can be evaluated on realistic linguistic data. There is an overwhelming amount of empirical evidence from corpora and laboratory studies in favor of construction grammar, and in favor of a usage-based account of language acquisition. Nevertheless, no systematic studies into the construction and its evolution over time have been undertaken, due to the lack of an adequate technique for automatic identification of constructions. The development of such a technique could enable us to study the acquisition and use of constructions in a more systematic manner.

Chapter 3

Unsupervised Grammar Induction

3.1 Introduction

The field of unsupervised grammar induction is concerned with understanding how grammatical structure can be inferred from plain text, and with developing computational techniques for doing so. It is believed that children who acquire a language do not store every sentence they ever hear (although there are exemplar-based theories where this assumption is made), but rather develop a compact internal representation of the language, which is cognitively more efficient to store. Opinions differ about the nature of this representation; for the current research we assume a tree substitution grammar, which is a bank of tree fragments that can be reused and recombined to interpret and to form new sentences.

Of course, we realize that apart from plain text, there are many other cues that aid children to infer the correct grammatical structure and categories. To name some: joint attention, visual information, prosody (intonation and stress patterns), and prior knowledge of the semantic categories. A constructionist theory of language learning should ideally take semantics into account, and in fact we have seen constructionist acquisition models [5] where it is assumed that a complete ontology is in place prior to grammar acquisition. When we induce grammar from plain text, we use distributional methods to abstract over this.

There is a long tradition of unsupervised grammar induction from text, starting perhaps with Wolff [70]. He introduced two learning operators, ‘chunk’ (or SYN) and ‘merge’ (or PAR).

- The chunk operator concatenates or chunks repeating patterns (sequences), so that the pattern may be stored just once in its entirety, and then accessed via pointers.

- The merge operator creates generalizations by forming disjunctive groups (categories) of patterns that occur in the same contexts. This enables one to store a single abstract rule consisting of the context plus the category, instead of storing separate sentences for every member of the category.

Learning is seen by Wolff as optimization of cognitive structures, in order to minimize storage demands. When compressing data for efficient storage or transmission, usually there is a trade-off between minimizing the size of the theory and minimizing the size of the data that is not explained by the theory. At one extreme, the theory can be compact but too general, so that not compress the data much; at the other extreme the theory can be overly specific by having a specific rule for every sentence in the data, but then it grows very large.

This can be formulated in an information theoretic framework as the principle of Minimum Description Length (MDL): For an optimal encoding one should take care that the encoding length of both the theory and the unexplained data is minimized. Stolcke [63] took Wolff's ideas of structure search through a grammar space by means of merging and chunking operators, and casted it in a probabilistic framework, so that it can be applied to the search for an optimal probabilistic context free grammar (PCFG). He calls this Bayesian Model Merging (BMM), since models, or grammars, are merged according to the Bayesian criterium of maximum a posteriori probability, which is just another formulation of MDL. As our own research builds on the work of Stolcke, much more is going to be said about BMM in the rest of the chapter.

There are also different approaches to unsupervised grammar induction. To name some: ADIOS [71], U-DOP [3], Seginer's unsupervised dependency parser [56], and the Constituent-Context model (CCM) [36]. It is useful to distinguish between models that perform a structure search through the grammar space, such as BMM, and models that do a parameter search, with fixed structure. The CMM model is an example of a parametrized model, and for comparison with BMM it will be discussed at the end of the chapter.

The first part of the chapter will introduce the formal framework of Bayesian learning and Bayesian Model Merging as an approach to unsupervised grammar induction. We will extensively deal with the algorithms for Bayesian Model Merging developed by Stolcke [63] - in short, the Stolcke algorithm - and by Petasis et al. [47] - which they call the e-Grids algorithm. A further section will discuss some of our own adaptations of Stolcke and e-Grids. From now on, we'll refer to our implementation in short as the BMM algorithm.

3.2 The Bayesian Approach

3.2.1 Probabilistic Context Free Grammars

A rewriting grammar is a tuple $\langle V_N, V_T, S, \mathcal{R} \rangle$ with V_N a set of nonterminals, V_T a set of terminals, S the start nonterminal, and \mathcal{R} a set of rewrite rules. In a *context free grammar* (CFG) sentences are assumed to be produced by repeated application of *context free rewrite rules*, which are of the form $A \rightarrow \alpha$, where A is a nonterminal ($A \in V_N$) and α is any sequence of terminals and nonterminals ($\alpha \in (V_N \cup V_T)^+$). The derivation of a sentence is a sequence of context free rules that generates the sentence, starting with the symbol S , and followed by repeated replacement of the left-most non-terminal by applying the next rule in the sequence.

In a CFG a derivation corresponds to exactly one parse tree. A CFG can be extended to a probabilistic CFG (PCFG) by assigning a probability to every rewrite rule:

$$P : \mathcal{R} \rightarrow (0, 1] \quad (3.1)$$

The context free assumption imposes the following constraint on the probabilities:

$$\forall A \in V_N : \sum_{\alpha: A \rightarrow \alpha \in \mathcal{R}} P(\alpha|A) = 1 \quad (3.2)$$

Using the context free assumption and the chain rule, it can be derived [44] that the probability of a derivation is the product of the probabilities of the rules r_i in the derivation sequence, given their LHS.

$$P(\text{der}|S) = \prod_i P(r_i|\text{LHS}(r_i)) \quad (3.3)$$

In a CFG a single sentence can have a large number of parses, and this number can grow exponentially with the length of the sentence. The main reason for employing PCFGs is that they allow to disambiguate between alternative parses by assigning a probability to every parse. Given an utterance U , and a PCFG G , then the most probable parse T is given by

$$\text{argmax}_{T \in G(U)} P(T|G) \quad (3.4)$$

where $G(U)$ is the set of parses of U licensed by G . The most probable parse is sometimes referred to as the Viterbi parse, after the Viterbi algorithm, which is a well-known algorithm to dynamically compute the most probable parse in a chart-style parser. The probabilistic parser can be evaluated by comparing the Viterbi parse to the manually annotated parse.

3.2.2 Maximum Likelihood Estimation

The rules and their probabilities can be estimated from a *treebank*, which is a corpus annotated with parse-trees, by relative frequency estimation.

Let $\mathcal{F}_{\mathcal{R}}$ denote the frequencies of the rewrite rules in \mathcal{R} , then its relative frequency in the treebank is given by

$$rf(A \rightarrow \alpha) = \frac{\mathcal{F}_{\mathcal{R}}(A \rightarrow \alpha)}{\sum_{\beta: A \rightarrow \beta \in \mathcal{R}} \mathcal{F}_{\mathcal{R}}(A \rightarrow \beta)} \quad (3.5)$$

The denominator denotes the total count of rules with LHS equal to A . In the above case the relative frequency estimation is equivalent to maximum likelihood estimation. Generally, a maximum likelihood (ML) estimator finds the parameters of a model M that maximize the likelihood of the data X , generated by M .

$$M_{ML} = \operatorname{argmax}_M P(X|M) \quad (3.6)$$

So far, we have considered how one can estimate the parameters of the PCFG from an annotated corpus (a treebank). Since our intention is to understand how children acquire language, we should however attempt to model language acquisition from a corpus without taking annotated parse trees for granted. In the following we will first deal with the case of how to estimate the parameters of the grammar given an unlabeled corpus and the rules of the grammar. This can be achieved by an instantiation of the so-called *Expectation Maximization* (EM) algorithm.

3.2.3 The Expectation Maximization Algorithm

Sometimes the variables of the model that one wants to estimate cannot be observed directly. For example, the productions that generate sentences cannot be observed unless a parse tree is provided; we can think of them as hidden variables. Still, it is often possible to estimate production probabilities even though the derivations are hidden. In case the structure of the model is known, we can resort to a technique called *Expectation Maximization* (EM). For instance for the case of a PCFG, EM can be used when all non-zero production rules are known beforehand.

The idea behind the EM algorithm is that the parameters of the model are initialized to arbitrary values, then the expected values of the hidden variables are calculated based on the initial model parameters. Then the maximum likelihood hypothesis given these values can be computed, by replacing the hidden variables by their expected values, resulting in adjustment of the model parameters. This process is then iterated, such that eventually the model parameters converge to a local maximum.

- In the E-step the expected values of hidden variables is calculated, given the approximated parameters.

- In the M-step, the model parameters are set to values that maximize the data likelihood, given the expected values of the hidden variables.

The instantiation of EM for estimating the parameters of a PCFG is called the Inside-Outside algorithm. Here the parameters are the rule probabilities, and the hidden variables are the derivations (the corpus is unlabeled).

In the the E-step of the Inside-Outside algorithm, the expected counts of the rules are computed from the derivations, given the samples and the current rule probabilities.

In the M-step the relative frequency of the rules is computed by equation 3.5.

In practice, a Viterbi-approximation is often used in the E-step of the EM algorithm. The expected values are approximated by assuming that the most probable parse (the Viterbi-parse) concentrates almost all probability mass of the sentence [63].

$$P(x|M) = \sum_{der} P(x|der, M) \cdot P(der, M) \approx \max_{der} P(x|der, M) \cdot P(der, M) \quad (3.7)$$

In the present work, based on [63], we intend to tackle the harder problem, where the structure of the grammar is not known beforehand, that is, the rules of the grammar are unknown. We will introduce two operators that change existing rules and induce new rules, ‘merge’ and ‘chunk’, and aided by these operators we can implement a (hill climbing) structure search through the space of grammars.

3.3 Bayesian Model Merging - The Stolcke algorithm

3.3.1 Initialization, Merging and Chunking

In the Bayesian Model Merging approach to grammar induction [63], the initial rules are set to incorporate all samples as follows: for each sentence $a_1a_2\dots a_l$, new nonterminals X_1, X_2, \dots, X_l are created and the following productions:

$$\begin{aligned} S &\rightarrow X_1, X_2 \dots X_l & (1) \\ X_1 &\rightarrow a_1 & (1) \\ X_2 &\rightarrow a_2 & (1) \\ &\vdots & \\ X_l &\rightarrow a_l & (1) \end{aligned} \quad (3.8)$$

Also, the count of every production is kept track of. After initialization the grammar is optimized through a greedy hill climbing search involving merging and chunking operations.

- The *merging* operator replaces two existing nonterminals X_1 and X_2 with a single new nonterminal Y . In the process two or more existing rules may merge into a single rule, if they become identical. In that case the count of the new rule is updated to the sum of the counts of the old rules. The rules which had X_1 or X_2 in their LHS are combined in the new nonterminal Y .
- The *chunking* operator takes a sequence of two nonterminals X_1 and X_2 and creates a new nonterminal Y that expands to X_1X_2 . We substitute all occurrences of the sequence X_1X_2 in the RHS of the productions by Y . The count of the newly created production is the sum of the counts of the productions where substitution took place.

At every step of the search, all candidate merges and chunks are considered, and a single one is selected that scores best on an evaluation function. It can be shown that any target CFG can be obtained from the merging and chunking operations alone, provided the training data was generated by a set of derivations that used all rules from the target grammar [63].

If we want to deal with a dynamically changing model with no fixed structure, the Inside Outside algorithm is no longer suited as an optimization algorithm. It is a known result that, if there are no constraints on the hypothesis space, a maximum likelihood model will evolve towards a trivial CFG, which doesn't generalize over the examples. Unlike in the fixed structure problem, model merging affects the structure of the grammar and not just the weights. We will need an evaluation function that incorporates our preferences for the model structure, to help us direct the search through the hypothesis space. In the EM algorithm, therefore, the maximization of likelihood is replaced by maximization of the a posteriori probability, which takes into account the probability distribution of the models prior to data.

3.3.2 Bayesian Learning and MAP hypothesis

Suppose we have a prior probability distribution over a hypothesis space, and we want to search for the hypothesis that maximizes the probability after some data X is given, the so-called posterior probability. This so-called Maximum a Posteriori (MAP) hypothesis, M_{MAP} , can be expressed in terms of the likelihood and the prior probability by applying Bayes Law:

$$\begin{aligned} M_{MAP} \equiv \operatorname{argmax}_M P(M|X) &= \operatorname{argmax}_M \frac{P(X|M) \cdot P(M)}{P(X)} = \\ &= \operatorname{argmax}_M P(X|M) \cdot P(M) \end{aligned} \quad (3.9)$$

Since the data X is given, $P(X)$ is constant and can be ignored in the maximization. The MAP hypothesis takes into account an a priori hypothesis, $P(M)$, called the

‘prior’, which is a best guess for the model before any data has been observed. As more data comes in, the influence of the prior diminishes. We may construct the prior such that it expresses the designer’s a priori preferences for the model: this is a probabilistic form of bias. Often the prior is designed such that it biases the model towards simplicity, as a way to implement Occam’s Razor, the idea that simpler models are preferred over complex models. If the prior $P(M)$ is uniform, it can be left out of the equation, and in that case the MAP hypothesis is equal to the ML hypothesis.

The maximization of $P(X|M) \cdot P(M)$ is equivalent to minimizing

$$-\log P(M) - \log P(X|M) \quad (3.10)$$

where \log is the logarithm with base 2. This equation is interpreted in information theory as the total description length (DL): The Grammar Description Length $GDL = -\log P(M)$ is the length needed to encode the model (rounded to an integer number of bits) and the Data Description Length $DDL = -\log P(X|M)$ are the bits that are needed to describe the data given the model (assuming an optimal, shared code). Finding the MAP hypothesis is therefore equivalent to estimation by *Minimum Description Length* (MDL).

In practice, we will employ MDL as the heuristic for evaluating candidate grammars during the search. We can then compute the likelihood and the priors separately, and as we will see later, we can even compute them for every nonterminal separately.

3.3.3 Priors

The model M can be decomposed into a structure part and a parameter part $M = (M_S, \Theta_M)$. Although the division is somewhat arbitrary, typically the structure enumerates the rules that have non-zero probability in the model, and the parameters represent the (non-zero) probabilities of existing rules. Hence, the prior is decomposed into a structure prior $P(M_S)$ and a prior of continuous parameters given the structure, $P(\Theta_M|M_S)$.

$$P(M) = P(M_S) \cdot P(\Theta_M|M_S) \quad (3.11)$$

For the a posteriori probability we therefore have

$$\operatorname{argmax}_M P(M|X) = \operatorname{argmax}_M P(X|M_S, \Theta_M) \cdot P(M_S) \cdot P(\Theta_M|M_S) \quad (3.12)$$

3.3.4 Structure Prior

The *structure prior* $P(M_S)$ should incorporate the preference for a simple model. It should therefore be a function of the code length of the model. In one proposal, the code length of the model is given by

$$DL(M_S) = NT \cdot (\log|\mathcal{N}| + 1) + T \cdot (\log|\Sigma|) \quad (3.13)$$

where $|\mathcal{N}|$ is the number of distinct nonterminals in the grammar, $|\Sigma|$ is the number of distinct terminals in the grammar, NT and T are the numbers of nonterminals respectively terminals in the RHS of the rules. The first term encodes the non-lexical productions. Each non-terminal symbol requires $(\log|\mathcal{N}|+1)$ bits to encode, where the 1 represents the end marker. The second term encodes the lexical production, which have a single terminal in the RHS, requiring $(\log|\Sigma|)$ bits to encode (no end marker needs to be encoded here). The distribution of the structure prior is then given by $\log P(M_S) = -DL(M_{PS})$. This prior gives rise to a geometric distribution which has its maximum for a grammar with productions of zero length.

The alternative proposal is to have the production lengths drawn from a Poisson distribution with mean μ . In this case the code length of a production of length k would be $\log(p(k-1; \mu)) + k \log|\mathcal{N}|$ bits (p is the Poisson distribution). The first term replaces the end marker (1 bit in the geometric distribution). $k-1$ is used because the minimum production length is 1. The total description length of the grammar DL is computed like before .

3.3.5 Parameter Prior

A common choice of priors for the continuous parameters are Dirichlet distributions, given by

$$P(\Theta) = \frac{1}{B(\alpha_1, \dots, \alpha_n)} \prod_{i=1}^n \Theta_i^{\alpha_i-1} \quad (3.14)$$

where we take the prior weights $\alpha_1 = \dots = \alpha_n$ and $\sum_i \alpha_i = 1$. $B(\alpha_1, \dots, \alpha_n)$ is the Beta function:

$$B(\alpha_1, \dots, \alpha_n) \equiv \frac{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1) + \cdots + \Gamma(\alpha_n)} \quad (3.15)$$

and $\Gamma(\alpha) = (\alpha-1)!$. Here the Θ_i 's are given by the probabilities of the i rules that share the same LHS, and there is a different Dirichlet distribution for every nonterminal. The Dirichlet prior is maximized for equal Θ_i , thus it biases towards a grammar where all rules have equal probability.

3.3.6 Computation of Likelihood

We still have to deal with the toughest part, which is the calculation of the likelihood $P(X|M_S, \Theta_M)$ over the entire corpus X , which must be evaluated for every single candidate merge. (The chunk operation does not affect the likelihood, since it does not change rule probabilities, because no rules are deleted).

For a given structure M , the likelihood of a single sentence is given by the sum of the probabilities of all the parses of the sentence. The likelihood of the corpus

is the product of the likelihood of the sentences, since sentences are assumed to be independent.

$$P(X|M) = \prod_{x \in X} \sum_{der: yield(der)=x} P(der) \quad (3.16)$$

Although the search algorithm searches only in structure space, to find the best model it requires a joint maximization over both the model structure M_S and its (hidden) parameters Θ_M . This would imply that evaluation of every single candidate structure involves the application of the Inside-Outside algorithm (with ML replaced by MAP).

Two approximations make the calculation feasible. First, it is assumed that most of the probability mass of the sentence is concentrated in the Viterbi parse, so the contribution of all non-Viterbi parses to the sentence probability and hence to the likelihood are ignored. Secondly, it is assumed that the merging operation preserves the Viterbi parse. This means that after a merge operation the Viterbi parse of the sentence is generated by exactly the same sequence of rewrite rules as before, except for the rules affected by the merge. This allows us to consider only the contributions or changes to the likelihood of those rules that are involved in the merge.

Trivially, the Viterbi parse at initialization is given by the initialization rules (3.8) with a probability of 1. At initialization the rules of the grammar are identical to the sentences in the corpus, so computing the likelihood over the grammar is equivalent to computing the likelihood over the corpus under the assumption mentioned above. The same remains true even after one alters the grammar by merging or chunking. This can be seen by rearranging the equation of the likelihood, regrouping the rules used in all the samples according to their left hand side. (This uses the earlier assumption that after a merge the Viterbi parse is affected only locally).

$$\begin{aligned} P(X|M) &= \prod_{x \in X} \sum_{yield(der)=x} P(der) \\ &\approx \prod_{x \in X} P(der(X)_V) = \\ &= \prod_{x \in X} \prod_{r_i \in der_V} P(r_i)^{C_i} = \prod_{A \in V_N} \prod_{r_i \in A} P(r_i)^{CC_i} \end{aligned} \quad (3.17)$$

where der is a derivation, der_V is the Viterbi parse, $r_i \in \mathcal{R}$ is a rewrite rule, $A \in V_N$ a nonterminal, C_i the count of rules occurring within single sentence and CC_i the count of rules occurring in the Viterbi parses of the entire corpus.

Therefore, if we keep count of the number of times that every rule is used in the entire corpus (implemented by the merging and chunking algorithms) we may equally well compute the likelihood over the nonterminals of the grammar as over the corpus.

For example, given the following rules for the nonterminals

$$S \rightarrow A B \quad (10)$$

$$S \rightarrow A A B B \quad (5)$$

$$S \rightarrow A A A B B B \quad (3)$$

$$A \rightarrow a \quad (30)$$

$$B \rightarrow b \quad (30)$$

the log likelihood would be $\log \left(\left(\frac{10}{18} \right)^{10} \cdot \left(\frac{5}{18} \right)^5 \cdot \left(\frac{3}{18} \right)^3 \cdot 1 \cdot 1 \right)$.

3.3.7 Search Algorithm

The search algorithm implemented in the present application is a multi-level best search with look-ahead. The idea behind look-ahead is that a candidate grammar is not discarded right away if it doesn't improve the a posteriori probability (AP): the algorithm continues the search with the best candidate, and only if after a specified number of additional merges the AP is still not improved, all the changes are discarded. Otherwise, all the changes to the grammar are accepted. The look-ahead looks only ahead for merges (since adding chunks hardly ever improves the AP). Multi-level means that the search is carried out in two alternating phases: in the first phase only merges are considered until a state is reached where no single merge improves the AP anymore. In the second phase the same is done for chunks. Phase 1 and phase 2 alternate until neither merges nor chunks improve the AP anymore, and in that case the algorithm stops.

3.3.8 Implementation

This concludes our description of the Stolcke algorithm. We have re-implemented the algorithm in Java, using Java runtime environment 1.5. In section 4 we will present the experimental evaluation of the Stolcke algorithm on benchmark tests and on CHILDES.

3.4 e-GRIDS

The egrids algorithm [47] was designed with the aim of making grammar induction of large natural language databases feasible by reducing the complexity of the computations. To achieve this aim, the authors of [47] derive mathematical formulas that describe the change of the likelihood and structure prior due to the application of a merge or a chunk operation. The formulas obviate the need to construct a grammar for every merge in the search, since the DL gain can be calculated beforehand.

In the terminology of e-Grids, assuming that the prior distribution over grammars $P(M_S)$ decreases exponentially with the description length of the grammar, the grammar description length (GDL) is given by $GDL = -\log(P(M_S))$. Likewise, the Data Description Length (DDL) is the negative logarithm of the likelihood.

3.4.1 GDL in e-grids

In the e-grids algorithm [47] the encoding of the grammar is optimized by splitting the rule set into three separate parts:

- the start symbol subset (SB1) contains all rules headed by the start symbol
- the terminal subset (SB2) contains all lexical rules (rules which have only a single terminal in their body)
- the non-terminal subset (SB3) contains all non-lexical rules headed by a non-terminal other than the start symbol

The reduction in encoding length follows from the fact that within the start symbol subset the LHS of the rule need not be encoded, because it is always the same; within the terminal subset no STOP symbol needs to be encoded, because all rule bodies have length one ¹. The transmitter and receiver must of course agree on the separation in three subsets. Every single non-terminal in the body requires $\log(A_{UNT} + 1)$ bits to encode, A_{UNT} is the number of unique non-terminals in the rule bodies (so excluding the START symbol), and the extra 1 is for the STOP symbol. A single terminal requires $\log(T)$ bits to encode, where T is the number of unique terminals. For any rule in $SB1$ there are $NT + 1$ symbols that need to be encoded: NT is the number of non-terminals in the rule body, and 1 represents the STOP symbol. For a rule in $SB2$ one needs to encode one non-terminal in the LHS and a single terminal in the rule body. For a rule of $SB3$ $NT + 2$ symbols need to be encoded, because in addition to the STOP symbol the LHS needs to be encoded. Altogether the Grammar Description Length (GDL) is given by

$$\begin{array}{l|l}
 GDL = \text{bits required to encode rules of} & SB1 \cup SB2 \cup SB3 = \\
 = \sum_{R \in SB1} (|NT_R + 1| \cdot \log(A_{UNT} + 1)) + & \text{Body } NT + \text{STOP symbol} \\
 + \sum_{R \in SB2} (\log(A_{UNT} + 1) + \log(T)) + & \text{one LHS } NT + \text{one Body Terminal} \\
 + \sum_{R \in SB3} (|NT_R + 2| \cdot \log(A_{UNT} + 1)) + & \text{LHS } NT + \text{Body } NT + \text{STOP symbol} \\
 + 2 \cdot \log(A_{UNT} + 1) + & \text{Subset Separators}
 \end{array} \tag{3.18}$$

¹The authors seem to overlook the fact that the rules of $SB3$ also have a fixed length of two. This is dealt with in appendix E

This equation can be rearranged in a more compact form:

$$GDL = \left(\sum_{SB_1} (|NT_R + 1|) + \sum_{SB_3} (|NT_R + 2|) + T + 2 \right) \cdot \log(A_{UNT} + 1) + T \cdot \log(T) \quad (3.19)$$

3.4.2 DDL in e-grids

In Stolcke [63], we have for log likelihood (see 3.17)

$$DDL = - \sum_{X \in V_{NT}} \sum_{Rules \ j \in X} F_{j_x} \cdot \left(\log \left(\frac{F_{j_x}}{F_{tot_x}} \right) \right) \quad (3.20)$$

where V_{NT} are the non-terminals of the grammar, F_{j_x} = rule frequency of rule j_x with head X and F_{tot_x} = frequency of all rules with head X (the index X indicates that the head of the rule is X). With the approximation, that all rules with the same head have equal probabilities (the ‘uniform distribution’) this becomes:

$$DDL = \sum_{X \in V_{NT}} \sum_{j_x} F_{j_x} \cdot (\log H_X) = \sum_{X \in V_{NT}} F_{tot_x} \cdot (\log H_X) \quad (3.21)$$

where H_X is the number of distinct rules with the LHS X

From the perspective taken in e-Grids, the frequency F_{tot_x} of rules with LHS $X \neq S$ can be obtained as the sum of the frequencies of the rules for which non-terminal X occurs in their body:

$$F_{tot_x} = \sum_{\substack{R : R \in SB_1 \cup SB_3 \\ \wedge X \in rhs(R)}} N_R^X \cdot F_R$$

where F_R is the frequency of rule R , and N_R^X is the number of occurrences of non-terminal X in the right hand side of rule R (note that this factor was erroneously omitted in the original paper). The contribution to the DDL of rules with LHS equal to S is accounted for separately. Thus, if we split F_{tot_x} as in the above equation, we obtain equation (2) of e-grids [47] for the log likelihood (DDL) (with the additional correction factor N_R^X)

$$DDL = \sum_{R \in SB_1} \log(H_S) + \sum_{R \in SB_1} \left(\sum_{X \in R-body} N_R^X \cdot \log(H_X) \right) F_R + \sum_{R \in SB_3} \left(\sum_{X \in R-body} N_R^X \cdot \log(H_X) \right) F_R \quad (3.22)$$

3.4.3 Contribution of the chunking operator to the Description Length

Chunking does not affect the DDL, but only the GDL. After a chunk which replaces the sequence $X Y$ by Z , a rule containing 4 additional symbols is added to SB3 (e.g. $Z \rightarrow X Y STOP$), and the rules of SB1 contain $BF(X, Y)$ non-terminals less (BF=bigram frequency). The set of non-terminals is augmented by 1, so every symbol requires $\log(A_{UNT} + 2)$ bits to encode. Thus, (ignoring the constant term $T \cdot \log(T)$), the GDL after $Chunk(X, Y) \rightarrow Z$ is

$$GDL_{After\ chunk} = \left(\sum_{SB1}(|NT_R + 1|) + \sum_{SB3}(|NT_R + 2|) + T + 2 \right. \\ \left. + (4 - BF(X, Y)) \right) \cdot \log(A_{UNT} + 2) \quad (3.23)$$

The GDL before the chunk was given by equation 3.19:

$$GDL_{Before\ chunk} = \left(\sum_{SB1}(|NT_R + 1|) + \sum_{SB3}(|NT_R + 2|) + T + 2 \right) \\ \cdot \log(A_{UNT} + 1)$$

and their difference is

$$\Delta DL_{CHNK(X,Y)} = \left(\sum_{SB1}(|NT_R + 1|) + \sum_{SB3}(|NT_R + 2|) + T + 2 \right) \cdot \frac{\log(A_{UNT} + 2)}{\log(A_{UNT} + 1)} \\ + (4 - BF(XY)) \cdot \log(A_{UNT} + 2) \quad (3.24)$$

which is equivalent to equation (4) of e-grids [47]:

$$\Delta DL_{CHNK(X,Y)} = (A_{NT} + A_R - A_S + 2) \cdot \log\left(\frac{A_{UNT} + 2}{A_{UNT} + 1}\right) \\ + (4 - BF(X, Y)) \cdot \log(A_{UNT} + 2) \quad (3.25)$$

The first term corresponds to the change in the number of bits needed to encode a single non-terminal in the grammar: $\log\left(\frac{A_{UNT} + 2}{A_{UNT} + 1}\right)$. This number is multiplied by the A_{NT} occurrences of non-terminals in the data, plus A_R STOP symbols, minus A_S non-terminals in the LHS of SB1, which need not be encoded, plus 2 separation symbols.

Since the first term is independent of any particular chunk, it plays no role in the search for the best chunk. The best chunk will always be the one with the highest bigram frequency (BF), and whether the chunk is applied depends on whether BF exceeds a certain threshold. The complexity of finding the best chunk is linear in the size of the grammar.

3.4.4 Contribution of the merging operator to GDL

The GDL after the application of a merge is

$$GDL_{After} = \left(\sum_{SB1}(|NT_R + 1|) - \sum_{\Omega_1}(|NT_R + 1|) + \sum_{SB3}(|NT_R + 2|) \right. \\ \left. - \sum_{\Omega_3}(|NT_R + 1|) + T + 2 \right) \cdot \log(A_{UNT}) \quad (3.26)$$

Since the number of non-terminals is decreased by 1, all symbols in the rule bodies are encoded by $\log(A_{UNT})$ rather than $\log(A_{UNT} + 1)$ bits. The grammar is furthermore made smaller through elimination of duplicate rules: Ω_1 is the set of rules from SB1 that are eliminated, and Ω_3 is the set of rules from SB3 that are eliminated.

If we take again equation 3.19 for the GDL before the merge, then the change of the GDL as a result of the merge is

$$\begin{aligned} \Delta GDL_{MRG} &= GDL_{After} - GDL_{Before} \\ &= (\sum_{SB1} (|NT_R + 1|) + \sum_{SB3} (|NT_R + 2|) + T + 2) \cdot \frac{\log(A_{UNT})}{\log(A_{UNT}+1)} \\ &\quad - (\sum_{\Omega_1} (|NT_R + 1|) + \sum_{\Omega_3} (|NT_R + 1|)) \cdot \log(A_{UNT}) \end{aligned} \quad (3.27)$$

The term including Ω depends on the specific merge, but the other term is a constant term for all merges.

3.4.5 Contribution of merging operator to DDL

One can think of the merging of the non-terminals X and Y as consisting of two subprocesses, executed subsequently. First, the rule sets with LHS X and Y are joined and receive the same LHS, which results in a different conditional probability of those rules and thus a change in DDL. Second, duplicate rules that may occur as a result of the merge are eliminated in the entire grammar.

Change in likelihood from merging the sets of rules of X and Y As for the first process, for the non-terminals X and Y, after merging them into the non-terminal Z their number of rules is increased by a factor $\frac{H_X+H_Y}{H_X}$ and $\frac{H_X+H_Y}{H_Y}$ respectively (before elimination of duplicate rules). Here again, H_X and H_Y are the numbers of rules with LHS X and Y respectively. From equation 3.22 we then find that the contribution to the change of DDL from the non-terminals X and Y is

$$\begin{aligned} \Delta DDL_{from\ merging\ LHS} &= \\ &= \sum_{R \in SB1 \cup SB3} \left(\sum_{X \in R-body} \log \left(\frac{H_X+H_Y}{H_X} \right) + \sum_{Y \in R-body} \log \left(\frac{H_X+H_Y}{H_Y} \right) \right) \\ &= \sum_{R \in G_{Fin}} \left(N_R^X \log \left(1 + \frac{H_Y}{H_X} \right) + N_R^Y \log \left(1 + \frac{H_X}{H_Y} \right) \right) \cdot F_R \end{aligned} \quad (3.28)$$

where N_R^X equals the number of occurrences of the non-terminal X in the RHS of rule R.

Change in likelihood from elimination of rules After the application of a merge, through elimination of rules, H_X , the number of rules with LHS X, may change. Assume that, as a consequence of a merge, two rules with the same LHS X become

duplicates, and therefore H_X decreases by one. Then, for every occurrence of X in the body of a rule, its contribution to the likelihood is increased by an amount $\log\left(\frac{H_X-1}{H_X}\right)$.

More generally, if we denote by θ the set of non-terminals from which duplicate rules were eliminated, and if we denote $M_X = \log\left(\frac{H_{X, after}}{H_{X, before}}\right)$, then we have for the change in DDL caused by elimination of rules:

$$\begin{aligned}
\Delta DDL_{from\ rule-elim} &= \\
&= \sum_{R \in SB_1 \cup SB_3} \sum_{X \in \theta, R-body} \log\left(\frac{H_{X, after}}{H_{X, before}}\right) \cdot F_R \\
&= \sum_{X \in \theta} \sum_{X \in R-body, R \in G_{Fin}} (M_X \cdot F_R) + \sum_{R \in S_1} M_S \cdot F_R \\
&= \sum_{X \in \theta} M_X \cdot \sum_{X \in R-body, R \in G_{Fin}} F_R + \sum_{R \in S_1} M_S \cdot F_R
\end{aligned} \tag{3.29}$$

Together, equations 3.27, 3.28 and 3.29 make up equation (6) of e-grids for the DL gain after a merge, plus an additional term $\sum_{R \in S_1} M_S \cdot F_R$ (with $M_S = \log\left(\frac{H_{S, after\ merge}}{H_{S, before\ merge}}\right)$). This additional term was omitted in the original egrids article, but our derivation shows that it should not be neglected.

In practice we can usually replace G_{Fin} by G_{In} equation 3.29 in order to simplify the computation, unless the rules that become duplicates differ in their LHS. In that case, one must look for the merged terminal in the final grammar G_{Fin} . Note, that equation 3.29 holds only by virtue of the approximation, mentioned earlier, that the rule probabilities conditioned on the LHS are distributed uniformly.

3.4.6 Searching in e-Grids

As opposed to the original e-Grids implementation, where a multi-level hill-climbing search is improved by a beam search, we implemented a multi-level best first search with lookahead, as in [63]. Multi-level means that the search for a grammar alternates between the two stages of merging and chunking. See section 3.3.7 for a detailed explanation. Unless indicated otherwise, we set the lookahead parameter to 5.

In order to improve the chunk search we added a one level deep beam search for the chunks: Thus, before every chunking operation we tested 15 chunks and selected the best one with lookahead.

3.5 Adaptations of e-Grids

E-grids was originally designed for a non-probabilistic CFG. We adapted the e-Grids algorithm, so it could accommodate the priors proposed by Stolcke [63]. We also relaxed the assumption of the uniform distribution of the rules.

3.5.1 Implementation of the Poisson distribution in e-Grids

The Poisson distribution takes into account our prior knowledge that the length of the rules has a Poisson distribution, which is centered about a mean length μ . (See section ?? for a more detailed explanation). It is incorporated in the structure prior, for rules from *SB1*, where it replaces the STOP bit, but not for rules from *SB2* and *SB3*, since their length is known beforehand. We recomputed the change in GDL for the Poisson distribution as a result of a merge or a chunk. Details can be found in appendix D.

3.5.2 Implementation of Dirichlet prior in e-Grids

The change of the Dirichlet priors caused by a merge follows from equation 3.14. Chunks have no effect on the Dirichlet priors of the candidate merges. For any candidate merge the change $\Delta GDL_{Dirichlet}$ contributed by the Dirichlet prior is the sum of the $\Delta GDL_{Dirichlet_X}$ for every non-terminal X that has duplicate rules as a result of the candidate merge. The update can be implemented quite efficiently if the rule frequencies are indexed together with the duplicate rules, which are stored for each of the candidate merges. The details will not be covered here.

3.5.3 Correction of ΔDDL from merging for non-uniform distribution

E-grids was designed for non-probabilistic context-free grammars, and therefore when we embed e-Grids in a probabilistic context, all rules with the same LHS are given the same probability. We termed this the ‘uniform distribution assumption’. In Stolcke’s derivation [63] however, the probability of the rules is not uniform, but proportional to the relative frequency with which they are used in the most probable derivations of the training sentences. The e-Grids equations for the likelihood (or DDL) gain can be modified to cover this case.

We will return to the encoding scheme of Stolcke [63], without the separation in 3 sets of rules of different types, because it simplifies the mathematics, and from our experience, the difference between the results obtained by the e-grids optimized encoding scheme and Stolcke’s encoding scheme are minimal.

As before, the contribution of the merging operator to the DDL can be split into two processes: first, joining the rules with LHS X and with LHS Y into a single set of rules. Second, removal of duplicate rules from the entire grammar.

1. Merging sets of rules

According to [63], the contribution of a non-terminal to the DDL is given by (cf. equation 3.20)

$$DDL_X = - \sum_{j: lhs(j)=X} F_j \cdot \left(\log \left(\frac{F_j}{F_{Tot_X}} \right) \right)$$

where F_j is the frequency of a rule j with LHS X , and F_{Tot_X} is the sum of the frequencies of all rules with LHS X . When the rules with LHS X are joined with the rules with LHS Y , the change in DDL contributed by rules that originally had X as their LHS is

$$\begin{aligned} \Delta DDL_X &= DDL_{X, \text{ After Merge}} - DDL_{X, \text{ Before Merge}} \\ &= - \sum_{j: lhs(j)=X} (F_j \cdot \log(\frac{F_j}{F_{Tot_{X+Y}}})) - \sum_{j: lhs(j)=X} (F_j \cdot \log(\frac{F_j}{F_{Tot_X}})) \\ &= - \sum_{j: lhs(j)=X} (F_j \cdot \log(\frac{F_{Tot_X}}{F_{Tot_{X+Y}}})) \end{aligned} \quad (3.30)$$

with $F_{Tot_{X+Y}} = \sum_{R: LHS(R)=X} F_R + \sum_{R': LHS(R')=Y} F_{R'}$.

The same holds for the contribution of rules with LHS Y , so altogether we have

$$\begin{aligned} \Delta DDL_{\text{from merging LHS}} &= \\ &= - \sum_{R_j \in X} \left(F_j \cdot \log(\frac{F_{Tot_X}}{F_{Tot_{X+Y}}}) \right) - \sum_{R_j \in Y} \left(F_j \cdot \log(\frac{F_{Tot_Y}}{F_{Tot_{X+Y}}}) \right) \end{aligned} \quad (3.31)$$

2. Elimination of duplicate rules

Consider a set ω of rules with the same LHS W , that become duplicates by merging the non-terminals X and Y . Before elimination of the duplicate rules, the probability of a particular rule R_j with LHS W being selected for expansion is $\frac{F_j}{\sum_{R_k: LHS(R_k)=W} F_k}$ where F_j is the usage frequency of rule R_j . As a consequence of the merge, the frequencies of the duplicate rules are summed up to $\sum_{R_k \in \omega} F_k$. For every non-terminal W in the LHS, there can be multiple sets of duplicate rules ω_1, ω_2 , etc. For example $\omega_1 = \{W \rightarrow X R Y, W \rightarrow X R X, W \rightarrow Y R X\}$ and $\omega_2 = \{W \rightarrow X P, W \rightarrow Y P\}$. Let us denote by Ω_W the set of sets of duplicate rules with LHS W . Summing over all LHS non-terminals W that have duplicate rules, and over all sets of duplicate rules, we have

$$\begin{aligned} \Delta DDL_{\text{from rule-elim}} &= \\ &= - \sum_{W \in \theta} \sum_{\omega \in \Omega_W} \left[\left(\sum_{R_j \in \omega} F_j \right) \cdot \log \left(\frac{\sum_{R_k \in \omega} F_k}{\sum_{R_k: LHS(R_k)=W} F_k} \right) \right. \\ &\quad \left. - \sum_{R_j \in \omega} \left(F_j \cdot \log \left(\frac{F_j}{\sum_{R_k: LHS(R_k)=W} F_k} \right) \right) \right] \end{aligned} \quad (3.32)$$

where θ is the set of LHS nonTerminals of the duplicate rules resulting from merging X and Y . Sets of rules with unequal LHS's that become duplicates are

treated as though they had equal LHS's (since it is assumed that sets of rules with different LHS's are merged in the previous operation).

Computational complexity of e-Grids operators

The improved efficiency of the merge and chunk operators in e-Grids results from the fact that forecasting the ΔDL from the formulas obviates the need to generate all child grammars after a merge or chunk operator. For a derivation of the computational complexity involved in merging and chunking refer to [47]. We will only reproduce the results here:

The complexity of applying the chunk operator is reduced from $O(N^2)$ to $O(N)$ if you use equation 3.25, where N is the number of non-terminals.

The complexity of the merge operator is reduced from $O(N^3)$ to $O(N^2)$ by use of equations 3.28 and 3.29.

3.5.4 Additional improvements of the e-Grids algorithm

Removal of STOP bit from SB3 As mentioned before, not only in SB2, but also in SB3 the STOP bit to indicate the end of the rule is redundant, since all the rules of SB3 have the same number of non-terminals (2) in their body. We worked out the e-Grids formulas leaving out the STOP bit in S3. Our expectation was, that this would delay termination of the chunking process, since the price to encode a chunk would be only 3 symbols rather than 4. The derivations can be found in Appendix E.

Missing term in formula for ΔDDL as result of rule elimination As mentioned before, a term $\sum_{R \in S_1} M_S \cdot F_R$, representing the reduction in choice of rules from SB1 (with the START symbol as their LHS) was omitted in equation (6) of [47]. After noticing that the effect of this term cannot be neglected, we reintroduced the term into 3.29.

Efficient implementation of the merge operator The process of finding the merge which decreases the DL most can be separated into two sub-processes. First, find all possible candidate merges, and store every candidate merge together with the set of duplicate rules that it yields. Second, enumerate the candidate merges, determine ΔDL for each of them, and choose the one that most decreases the DL.

The advantage is that thus only at initialization the sets of duplicate rules need to be exhaustively searched. After the application of a merge, one can look for incremental changes in the sets of duplicate rules (Ω_1 and Ω_3) of the other merges by inspecting only those rules where a substitution occurred as a result of the merge. Thus, rather than determining the sets of duplicate rules Ω anew after every merge,

one can find $\Delta\Omega^+$ by comparing only those rules where a substitution took place, and $\Delta\Omega^-$ by inspecting the duplicate rules, of a particular merge, that become identical as a result of the applied merge.

3.6 Other work on Unsupervised Grammar Induction

3.6.1 The CCM model

The Constituent-Context model (CCM) [36] is an algorithm for the unsupervised induction of constituent boundaries (brackets).

A bracketing of a sentence is a boolean matrix $B_{ij} = 0, 1$, $j \geq i$ for which

$$B_{ij} = \begin{cases} 0 & \text{span } \langle i, j \rangle \text{ distituent} \\ 1 & \text{span } \langle i, j \rangle \text{ constituent} \end{cases} \quad (3.33)$$

The elements of the bracketing matrix represent all the possible spans $\langle i, j \rangle$ of a sentence with their constituency information. In the CCM model, the marginal probability of the sentence S is given by conditioning its probability on the bracketings B and then summing over all possible bracketings of S

$$P(S) = \sum_B P(B) \cdot P(S|B) \quad (3.34)$$

where $P(B)$ is the selected prior distribution for the bracketing and $P(S|B)$ is the likelihood of the observed data given a bracketing B .

Every span defines a *yield* Y_{ij} , which is the sequence of words from the sentence that corresponds to the span $\langle i, j \rangle$ and a *context* C_{ij} , which is the pair of words surrounding the yield. With the naïve Bayesian classifier assumption of independency between features, the sentence probability can be partitioned into the joint (conditional) probabilities of all yields and contexts. It follows that for a given bracketing B (consisting of the constituencies B_{ij} of all sentence spans), the sentence probability equals the joint probability of the yields $P(Y_{ij}|B_{ij})$ and contexts $P(C_{ij}|B_{ij})$, conditioned on the constituency of the spans. Thus we have

$$P(S|B) = \prod_{\langle i, j \rangle \in \text{spans}(S)} P(Y_{ij}|B_{ij}) \cdot P(C_{ij}|B_{ij}) \quad (3.35)$$

This means that if one would know the parameters $P(Y_{ij}|B_{ij})$ and $P(C_{ij}|B_{ij})$ for all possible yields and contexts, and the priors $P(B)$, one could compute the most probable bracketing for every sentence ($\text{argmax}_B P(S|B)$).

For example, for all of the spans of the sentence *The cat walks on the roof* one would like to estimate the probabilities of the yield and context being a constituent and being a distituent, so one would estimate $P(Y_{1,3} = \text{cat walks}|DS_{1,3})$, $P(Y_{1,3} =$

$cat\ walks|CS_{1,3}$), $P(C_{1,3} = The_on|DS_{1,3})$, $P(C_{1,3} = The_on|CS_{1,3})$, etc, for all spans. (DS=distituent, CS=constituent).

The set of probabilistic ‘production rules’ from span constituencies to yields and contexts should be interpreted as constituting the rules set of the ‘generative model’ of the CCM. The constituents and distituents generate yields and contexts, which in turn generate the sentence.

The idea is now to estimate the hidden parameters $P(Y_{ij}|B_{ij})$ and $P(C_{ij}|B_{ij})$ using the EM algorithm.

- First, one initializes the parameters by some random values.
- In the M-step one finds the most probable bracketings for every sentence in the corpus by a maximum likelihood estimation, according to the current parameter values: $argmax_{\{B\}} \sum_S \sum_B P(B) \cdot P(S|B)$, where $P(S|B)$ is given by 3.35.
- In the E-step, one estimates new values for the parameters by counting the relative frequencies of $Y_{ij}|B_{ij}$ and $C_{ij}|B_{ij}$ from the bracketings that were obtained in the M-step.

The authors choose a prior $P_{bin}(B)$ that puts (a uniform) mass only on tree bracketings and has zero probability for all other bracketings. A tree bracketing corresponds to a binary tree; it is obtained if one sets the diagonal elements $B_{ii} = 0$, the next diagonal (corresponding to spans of the preterminals) $B_{i,i+1} = 1$, and the top right element $B_{0,length} = 1$, and furthermore allows only for binary branches. This bias is needed if one wants to stimulate the model to induce valid trees; without it the model ‘degenerates’ to a soft clustering algorithm that is not able to distinguish between distituents and constituents.

To summarize the main differences with the BMM algorithm:

- The CCM induces bracketings, that is parses without labels, as opposed to the Stolcke algorithm, which induces bracketings as well as labels.
- The generative grammar underlying the CCM is not a PCFG, but a grammar of bracketings generating yields and contexts
- There is no ‘model merging’ or structure search; it is a parametrized model with a fixed structure, and the parameters are found by the EM algorithm.
- The hidden ‘bracketings’ (or chunks) are produced directly from both the yield and the context information. In the Stolcke algorithm, only the yields (bigrams) are used directly to choose chunks; the contextual information is exploited only through the intermediation of the merges: thus, successful chunking depends crucially on successful merging and vice versa.

Chapter 4

Experiments

4.1 Materials and evaluation

Experiments were carried out with 3 corpora: OVIS, Wall Street Journal (WSJ), and CHILDES. The first two are manually annotated corpora (so-called treebanks), which can be used for evaluating the correctness of the induced parses. The original aim of the research was to induce a grammar for the CHILDES corpus, which is not annotated, and the annotated corpora served to test the induction algorithm before moving on to CHILDES. From the 3 corpora 4 training sets were extracted:

	OVIS	WSJ10- postags	WSJ10- lexical	CHILDES
Sentences	10040	7422	7422	47170
Sentences>1word	6892	7263	7262	35191
Words	31697	52089	52072	142880
Av. sentence length	4.60	7.17	7.17	4.06
Vocabulary	946	35	11122	3566
Av. token frequency	33.5	1488	4.68	40.1
Types with freq<10	658	0	5266	2478

Figure 4.1: Characterization of corpora used in experiments

- OVIS [55] is a specialized, homogeneous corpus containing annotated Dutch sentences from a public transport information system. It contains 10040 sentences, and has a vocabulary of 946 words. See figure 4.1 for details.
- WSJ10-POSTAGS is the portion of 7422 sentences of length ≤ 10 extracted from the Penn treebank Wall Street Journal section (WSJ) [17]. Because WSJ

is too large for most of our experiments, we only looked at this subset. The Part-Of-Speech (POSTAG) sequences are used as input for the induction algorithm. This results in a vocabulary of 35 POSTAGs. Following [38], punctuation and null-elements were removed before sentences of length ≤ 10 were selected.

For this special case we incorporated the prior linguistic knowledge that the POSTAGs are separate categories by not allowing POSTAGs to merge with each other directly. So, a POSTAG was only allowed to merge with a non-terminal that resulted from a chunk operation, or with one resulting from another merge operation.

- WSJ10-lexical is the same set of utterances as WSJ10-POSTAGS, but rather than POSTAGs lexical input is used to induce the grammar. This is a very heterogeneous set, with a vocabulary of 11122 distinct words.
- CHILDES is a collection of corpora of spoken child language [43]. We selected the ADAM corpus, based on Brown 1973, because it is one of the biggest corpora in CHILDES, and it offers an opportunity to do a longitudinal study on grammar acquisition. Adam’s age ranges from 2;3 to 4;10. The total number of child utterances in the corpus is 46,722.

We removed the parental speech and any annotation or comments. We also removed from the child’s speech repetitions, incomplete and interrupted sentences, ununderstandable speech, and paralingual events. We accepted word completions and corrections. This yields a corpus of 47170 sentences with a vocabulary of 3566 words.

	WSJ10-lexical filtered	CHILDES filtered
Sentences	4715	28904
Sentences > 1word	4715	28904
Words	34689	115264
Av. sentence length	7.36	3.99
Vocabulary	5848	1087
Av. token frequency	5.93	106
Types with freq < 10	5266	85

Figure 4.2: Characterization of filtered corpora

In order to cope with the ‘sparse data’ problem in CHILDES and in the WSJ-10-lexical set we further considered training sets where infrequent words are filtered out.

From CHILDES we removed all sentences for which less than 85% of the words had frequency <10 . This led to a reduction to 28904 sentences and a vocabulary of 1087 words.

For WSJ10-lexical this was not possible, because it would lead to an almost complete reduction of the corpus. Therefore we filtered out those sentences for which less than 70% of the words had frequency 4 or more. This reduced the corpus to 4715 sentences with a vocabulary of 5848 words.

The filter on CHILDES proved to be very effective: in the filtered corpus only 85 words are left with frequency <10 . Thus it seems that the sparse data problem is solved. This is essential for successful merging: the results in section 4.4 show that almost all words of CHILDES-filtered could be categorized.

In contrast, we were not able to effectively filter out unfrequent words from WSJ10. After application of the filter, still 5266 out of 5848 words have frequency <10 . There remains a serious sparse data problem. Merging is therefore not very successful, as can be concluded from the results in section 4.3.

As a last adaptation, we removed single word sentences from all corpora, because those will cause an artifact of spurious merges (single word sentences will always merge with each other).

4.1.1 Evaluation

To evaluate our results and be able to compare them with other induction algorithms the *PARSEVAL* metric is used as a standard ([44], page 432). The constituents of the computed parse tree are compared one by one to the constituents of a manually annotated parse tree, a so-called gold standard.

A constituent labeled X which spans from position i to j in the sentences is written as the triple $\langle i, X, j \rangle$. Constituents with the S label and constituents representing preterminal nodes are omitted from the sets. See figure 4.3 for an example.

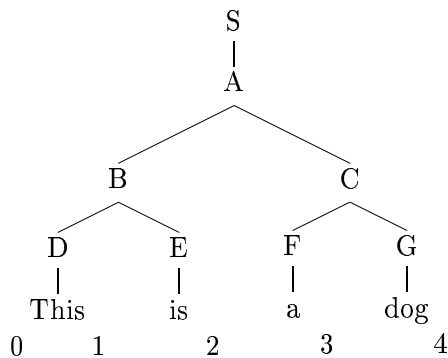
Let $\mathcal{C}(T_C^i) = \{T_C^1 \dots T_C^n\}$ be the constituents of the correct parse and $\mathcal{C}(T_G^i) = \{T_G^1 \dots T_G^m\}$ the constituents generated by the parser. Two measures are generally used to capture the success of the parse:

- Labeled Recall (LR) is the number of matching constituents in the parse relative to the total number of constituents in the correct parse (the proportion of correct constituents that are recognized by the parser)

$$\text{Labeled Recall } LR = \frac{\sum_i |\mathcal{C}(T_C^i) \cap \mathcal{C}(T_G^i)|}{\sum_i |\mathcal{C}(T_C^i)|}$$

- Labeled Precision (LP) is the proportion of correct constituents relative to the total number of constituents in the produced parse.

$$\text{Labeled Precision } LP = \frac{\sum_i |\mathcal{C}(T_C^i) \cap \mathcal{C}(T_G^i)|}{\sum_i |\mathcal{C}(T_G^i)|}$$



$\langle 0, A, 4 \rangle, \langle 0, B, 2 \rangle, \langle 2, C, 4 \rangle$

Figure 4.3: Example of PARSEVAL constituents

For the evaluation of unsupervised learning algorithms unlabeled precision and recall are used. This means that only the spans are considered for a match, and the label names become irrelevant. We adopted a version of the UP and UR measures introduced by [38], which differs from the standard PARSEVAL metrics in the following ways: the sentence-level (S) bracket is counted, brackets of span one are ignored, multiplicity of identical brackets is ignored, and averaging is done over all the brackets in the treebank (rather than per sentence). The F-score is defined as the harmonic mean of UP and UR: $F = \frac{2*UP*UR}{UP+UR}$.

For the annotated corpora lower and upper bounds on reasonable performance can be computed. The lower bound is commonly taken to be the Right Branching (RB) score, obtained by bracketing from right to left, e.g. [*The [cat [sits [on [the [tree]]]]]]]. Note, that right-branching is a significant linguistic fact of English, so the RB scores for English sentences are generally better than random scores. The upper bound is the score of the parses of the grammar extracted from the treebank. For OVIS and WSJ-10 the bounds are given in the table below.*

	UP-RB	UR-RB	UP-Treebank	UR-Treebank
OVIS	68.91%	66.30%	n.a.	n.a.
WSJ-10	70,0%	55.12%	90.10%	88.84%

Figure 4.4: Lower and upper bounds

4.2 Results on Benchmark Tests

Here we report the first results of the algorithms of Stolcke [63] and Petasis [47] on the benchmark tests OVIS and WSJ10-POSTAGS, the latter being the prime benchmark used in recent grammar induction research ([37], [3], [56]). Table 4.2 summarizes results from experiments with the algorithm, where we varied the Poisson parameter μ , and compared it to the geometric prior (see section 3.3.4). EXM refers to ‘exact match’, the proportion of parses where the induced brackets match the treebank brackets exactly. We assumed a uniform distribution of the rules with the same LHS, as in [47]. As will become clear, these results are very disappointing. In section 4.5 we report on a series of additional experiments aimed at clarifying the causes of the failure of the algorithm.

	OVIS				WSJ			
	UP	UR	F	EXM	UP	UR	F	EXM
Geometric	72.08	65.41	68.58	55.48	59.26	37.84	46.19	7.13
Poisson								
$\mu = 1.5$	71.62	66.64	69.04	55.92	52.50	44.40	48.11	7.66
$\mu = 2.0$	71.60	66.64	69.03	55.87	55.57	43.21	48.62	7.56
$\mu = 2.5$	71.70	66.56	69.03	55.72	56.57	43.00	48.86	7.59
$\mu = 3.0$	71.96	66.25	68.99	55.64	57.57	42.65	49.00	7.59
$\mu = 3.5$	71.88	66.15	68.90	55.64	57.45	42.00	48.52	7.53
$\mu = 4.0$	71.86	66.11	68.87	55.64	57.80	42.00	48.65	7.57

Figure 4.5: PARSEVAL scores for OVIS and WSJ10-POSTAGS

For OVIS, the UP score with the geometric prior is slightly better than with the Poisson distribution. UP is just above the lower bound (RB-UP=68.91%). The best UR score is for the Poisson prior with the lower values of μ , and this score is in the region of the lower bound (RB-UR=66.30%). The F-scores are very similar, whether the Poisson or the geometric distribution is used, and the effect of the parameter μ seems to be negligible.

To test whether despite the similar scores there is a qualitative difference in the parses, we compared the parses of the geometric distribution to those of the the Poisson distribution. The results are discussed in section 4.5.2.

For WSJ10-POSTAGS both the UP and the UR scores are far below the lower bound. The trends are the same as for OVIS but a bit more pronounced. The best F-score is for the Poisson distribution with $\mu = 3.0$ but for other values of μ the scores are very near.

When we compare the scores on WSJ10-POSTAGS to those published by previous

work on unsupervised grammar induction, they do not stand out well. Table 4.6 compares our results to the results of the Constituent-Context Model (CCM) of Klein & Manning [36], the combined dependency learning and CCM model (DMV+CCM) [37], and the U-DOP model of Bod [3].

	UP	UR	F
CCM	64.2	81.6	71.9
DMV+CCM	69.3	88.0	77.6
U-DOP	70.8	88.2	78.5
BMM	57.57	42.65	49.00

Figure 4.6: Results of BMM compared to previous work on the same data

4.2.1 Dirichlet prior

As mentioned in section 3.5, when we extend e-Grids to a probabilistic model (PCFG), we can impose a parameter prior on the rule probabilities. This is the Dirichlet prior, which was discussed in section 3.3.5. The Dirichlet prior biases the grammar towards categories whose members are equally frequent, which results in the formation more homogeneous categories. Thus, we expect that categories such as $\{this, that, these\}$ will not attract less frequent words, and remain relatively free of noise. We tested the effect of the Dirichlet prior (D) on OVIS and WSJ10-POSTAGS with the geometric distribution (G), and with the Poisson distribution (P), where we took $\mu = 2.5$. However, as can be seen in the result table, there is no indication that the Dirichlet priors give any improvement on the PARSEVAL scores.

	WSJ			OVIS		
	UP	UR	F	UP	UR	F
¬D G	59.26	37.84	46.19			
¬D P	56.57	43.00	48.86			
D G	54.96	39.45	45.93			
D P	52.03	43.73	47.52			

Figure 4.7: Scores with and without Dirichlet

4.2.2 Dropping the uniform distribution assumption

The table below compares the PARSEVAL scores of OVIS and WSJ-10-POSTAGS with and without the uniform distribution assumption (U). This time we show the

Poisson condition (P) with $\mu = 3.0$ because this gave the highest score in both conditions¹. (G) stands for geometric distribution (no Poisson), and the Dirichlet prior is turned off.

	WSJ10-POSTAGS					
	UP	UR	F	DL	GDL	DDL
U G	59.26	37.84	46.19	317189	202604	114584
U P	57.57	42.65	49.00	294094	160862	133231
¬U G	60.44	39.22	47.57	314177	197105	117071
¬U P	57.63	43.10	49.32	289051	165131	123919
	OVIS					
U G	72.08	65.41	68.58	n.a.	n.a.	n.a.
U P	71.96	66.25	68.99	n.a.	n.a.	n.a.
¬U G	73.04	69.20	71.07	n.a.	n.a.	n.a.
¬U P	71.75	68.17	69.91	n.a.	n.a.	n.a.

Figure 4.8: Scores with and without Uniform distribution assumption

As the table shows, when the uniform distribution assumption is dropped, the F-score improves slightly, but not significantly. The effect on the description length of dropping the uniformity assumption in the induction algorithm is minor, although it has a major effect on the description length of the treebank grammar.

4.3 Results for WSJ-lexical

The PARSEVAL scores on WSJ-lexical are very low: with Poisson distribution set to $\mu = 2.5$, no Dirichlet prior and uniform distribution assumption, scores of $UP = 25.05$, $UR = 56.23$, and $F = 34.66$ were obtained. It is already obvious from inspecting the induced categories and constituents that the algorithm failed to induce a full-fledged grammar: only 663 out of 5848 of the terminals are categorized, which amounts to 11.3%. The majority of the words thus remain unclassified. The vast majority of the categories are number categories, and there are a few that are typical of the content of WSJ: $\{acquisition, transaction, agreement, accord\}$, $\{dropped, gained, rallied, chnk\#were|up\}$, $\{Lyonnais, Suisse\}$.

There are 584 chunks, the majority of which is between two individual words, rather than between categories. In only 9 cases the chunks are between categories. (The run was with the following settings: Poisson distribution, $\mu = 2.5$, Dirichlet prior and non-uniform distribution of rule probabilities.

¹Except for the condition $\neg U - P$, where $\mu = 3.5$ gave a slightly better F-score: $F = 70.40$

Obviously, the cause of the relatively small number of merges is the sparse data problem. We saw already in section 4.1 that 5266 out of the vocabulary of 5848 words occur with frequency < 10.

4.4 Results for CHILDES

The parses of the CHILDES corpus cannot be evaluated against a gold standard; we therefore discuss only the qualitative observations. The listed results were obtained for the filtered CHILDES corpus, consisting of 28904 sentences. Various parameter settings were tried with comparable results. Here we discuss the results of a setting with Geometric distribution of the structure prior, Dirichlet distribution of the parameter prior, and without the uniformity assumption. During the merging phase, initially some very plausible linguistic categories are discovered. We give the most interesting ones, with the words listed in the order that they were added (note, that most of these categories eventually cluster, so they don't survive until the end):

- A category of indicatives: $\{dis, it, dat, this, that\}$. Let us refer to this category as the 'dis' category. Subsequently, the category is extended with some personal pronouns, namely $\{him, dem, them, dese, dose, these, dat's, that's\}$, then it absorbs an entire category of nouns, after which the 'dis' category becomes a mixture of nouns, adjectives, adverbs and function words, eventually absorbing almost every other category.
- Starting out as a category of personal pronouns with occasional proper nouns: $\{I, we, you, he, she, Paul, dey, Robin, they\}$, after the inclusion of some 'noise' words $\{it's, let's, cha, I'm\}$ this category is absorbed in the 'dis' category.
- $\{dere, here, there\}$
- A category of Wh-words or prerogatives: $\{what, what's, who, where, where's, how, why, which, who's, what're\}$. Later, some unrelated words are inserted: $\{ok, everybody, sure, whose, aren't, drawing, harpsichord, fishing, etc.\}$. This category survives till the end.
- A category of modal verbs: $\{can, could, will, can't, going, may, better, gonna\}$: it is absorbed as a whole in the 'dis' category.
- A category of frequent transitive verbs: $\{need, want, got, found, have\}$. This is absorbed in a more general verb category, described below, which is in turn absorbed in the 'dis' category. also, $\{are, is\}$ initially have a category of their own.

- A transitive verb category: {*do, open, blow, close, say, mix, make, fix, push, use, cut, break*, etc.} This category remains a pure verb category, and assembles 62 transitive and intransitive verbs until it is eventually absorbed in the ‘*dis*’ category.
- A category of nouns, initially mainly drinks, usually encountered in the context of ‘*I want*’, or ‘*give me*’: {*more, sugar, milk, scissors, money, tea, water, coffee, juice*, etc.}. This is then absorbed in the ‘*dis*’ category.
- A category of determiners: {*a, another, de, the*}. It merges with the category {*my, your*}, which is then absorbed in the ‘*dis*’ category.
- A category of prepositions: {*in, on, apart, out, off, down, together, through, over, up, back, away*}, merges with a category consisting of {*fun, right*} after which it is absorbed into the ‘*dis*’ category.
- A category of exclamations: {*oink, ow, ho, gobble, yum, quack, pow, beep, ding, santa claus*}, and another category consisting of {*hmm, huh, mom*}

Eventually every other category is absorbed into the ‘*dis*’ category, except for the exclamation category, the question word category, and one other minor category.

In runs with different parameter settings, the results were similar: the major categories, such as an indicative category, a verb category, a food category, which turns into a noun category, a question-word category, personal pronouns, exclamations, modal verbs, propositions, are persistent in all runs. There is generally one big category that acts as a magnet absorbing all other categories; in some cases this starts off as a noun category, or the verb category.

Yet other categories are discovered. With the Geometric distribution, no Dirichlet, and uniformity assumption, for instance the algorithm finds in addition to the former categories, categories for colors and adjectives, animals, count words, and body parts:

- Colors and other adjectives: {*red, yellow, green, grey, blue, black, brown, silly, nice, raining, waiting, bucket, dancing, dining, each, fourteen*}. At this point the category clusters with a category of count words {*eight, ten, six, three, remember, ursula’s, nineteen, working*}, and then it further clusters with some frequent adjectives categories: {*hard, heavy, strong, happy, tight, having, moving, parking, long*}, {*good, old, wait, cool, bad, tape, gone, already, shiny, wonder, fat*}, and {*big, other, little, own, boy, baby*}. This would make for an adjective category if not soon after it was absorbed by the noun category, which eventually becomes a category of everything.

- A category of animals, formed by small animal clusters: {*seal, snake, lobster, kitten, whale, nut, lions, mixer, dinosaur, saucer*} clusters with {*duck, drum, wings, lumber*}, {*lion, tiger, pail, elephant, glass, diaper, tires, fence*} and with {*cat, door, balloon, button, kitchen, spider, dirt*} before all of them are absorbed in a general noun category.
- A category of body parts: {*mouth, nose, eyes, ear, end, coat, trucks*}
- A category of proper names: {*Cromer, Ursula, Paul, mommy, Robin*}

To conclude, the merging process looks promising initially, but through the lack of a good stopping criterion, eventually all categories are clustered into one or a few supercategories. Initially categories are distinguished not only on the basis of syntactic function, but also on a semantic basis. Many adult part of speech categories are identified ‘correctly’. However, no ‘adult’ syntactic categories are detected higher up in the hierarchy. The good news is that there is no sparse data problem (using the filtered corpus): eventually almost every word is categorized. For instance, in the first setting there were only 2 unclassified words: *prudential* and *tower*.

In the current setting of the search algorithm, chunking sets in very late in the search process, after all possible merges have already been exploited. At this stage there are however only a few supercategories left, and the chunks make chains of these supercategories: for instance, in the first setting, the algorithm finds only 2 chunks before it stops. Those are: $CHNK(dis, dis)$ which chunks together two instances of the ‘*dis*’ category, and $CHNK(dis, dis, dis, dis)$, which chunks together twice the former chunk.

Obviously, with the kind of constituents discovered by the algorithm parsing makes no sense. We have therefore not attempted to evaluate the parses, or use this version of the BMM algorithm for providing input to phase of the construction discovery procedure.

4.5 Follow-up experiments

4.5.1 Intermediate evaluation - what goes wrong?

For evaluating where the algorithm goes wrong, the F-scores are not very informative. When we inspecting the merging and chunking process directly, we can observe 4 major problems :

1. Initially, the categories formed by the merges do make sense, but after a while they attract more and more unrelated words (noise). It seems that this effect is self-reinforcing.

2. Categories that initially make linguistic sense are often absorbed in bigger categories, especially if the bigger category contains a lot of noise. The tendency is for all categories to merge into one big category.
3. There are many ungrammatical chunks, which are formed by cutting across constituent boundaries, e.g. *put the, on the, and a, up and*. This can be explained by the fact that the prime criterium for selecting a chunk is the bigram frequency.
4. Higher up in the hierarchy there are very few chunks that make linguistic sense. This is probably due to the fact that by the time the algorithm reaches the chunking stage, the categories have already become too noisy.

Following are a number of suggestions for the cause of the problems:

1. We exploited the principle of minimal description length (MDL) which assumes that the optimal grammar is the one that minimizes an objective function. Formulated in probabilistic terms, we try to find the grammar G that maximizes the posterior probability, as given in 3.12 and repeated here

$$\operatorname{argmax}_G P(G|D) = \operatorname{argmax}_G P(D|G_S, \Theta_G) \cdot P(G_S) \cdot P(\Theta_G|G_S) \quad (4.1)$$

(D is data, G_S is the structure prior; $P(\Theta_G|G_S)$ is the parameter prior). We may not have found relevant priors that adequately express the a priori knowledge. The heuristics used in the algorithm to make the calculation feasible, for instance the ‘Viterbi approximation’ (section 3.3.6), may not approach the real objective function in the end. Alternatively, it might be possible that the MDL principle is not applicable to find an optimal grammar. After all, the construction grammar approach, which we adhere to, advocates a redundant storage of lexical-grammatical items [26].

2. The search algorithm

The algorithm employs a greedy best first hill-climbing heuristic search with look-ahead.

- Greediness implies that the search doesn’t look for long distance profits, so it can get stuck in local minima. The look-ahead is activated only when no grammar is found which minimizes the description length in which case it can help lift the search over a small valley. We didn’t implement a beam search because it is computationally too costly.
- The variations of the grammars that are considered in the search are obtained by across the board substitution of non-terminals in the entire grammar. This may make the gold standard grammar unreachable.

- The alternation between a merging phase and a chunking phase may be undesirable.
 - Possibly not all operations are available that are needed to reach the optimal grammar from the initial grammar. Petasis proposes in [?] a series of additional operators besides merging and chunking. We have not investigated this option.
3. We have used a PCFG as the representational model in our induction algorithm, since such a PCFG-based algorithm was well-documented in the existing literature. However, if we assume the theory of construction grammar then a tree substitution grammar formalism such as used in DOP [2] would be better suited to serve as the underlying representation. In section 6 we will elaborate on this option.
 4. With the CHILDES corpus we encounter the additional problem, that the child's grammar is continuously changing. An option would be to incrementally incorporate new sentences in chronological order and update the grammar accordingly.

4.5.2 Evaluation of the effect of the priors on the parses

In the light of the negative results with different settings for the priors we performed a series of tests investigating the effect of the Poisson and Dirichlet priors and of the uniform distribution assumption on the parses. To measure the divergence of the parses with different settings we matched the parses against each other using the familiar PARSEVAL scores. Thus, we tested the effect of the Poisson prior by comparing parses using the Poisson distribution at different intervals with parses using the Geometric distribution. Likewise, we tested the effect of the Dirichlet prior by comparing parses with and without and, and the effect of the uniform distribution assumption. Following are the results of the test on OVIS, whereby the intervals are chosen after 20, 30, 50 and all chunks:

It thus seems that the effect on the parses of the priors and of assuming a uniform distribution of the rule probabilities (conditioned on the LHS) is minor, with F-scores remaining above 90% and exact match varying between 68% and 80%. This result corroborates the finding that the F-scores for different parameter settings for the priors matched against the treebank grammar are all very similar. The fact that scores in some cases go up again at 50 chunks suggests that similar chunks and merges occur for all parameter settings, albeit in a different order. We may therefore conclude that the reason that grammar induction is not so successful does not depend too much on the chosen parameters.

	UP	UR	F	EXM
P vs \negP (\negU, \negD)				
After 20 chunks	97.66	95.48	96.56	85.71
After 30 chunks	95.89	93.59	94.73	79.50
After 50 chunks	96.99	94.4	95.68	85.15
Final	93.35	93.82	93.58	79.50
U vs \negU (\negP, \negD)				
After 20 chunks	96.32	87.15	91.51	69.51
After 30 chunks	94.38	83.98	88.88	61.56
After 50 chunks	94.06	84.46	89.00	63.65
Final	91.33	86.85	89.03	67.88
D vs \negD (\negU, \negP)				
After 20 chunks	94.15	95.76	94.95	76.39
After 30 chunks	94.47	95.25	94.86	79.97
After 50 chunks	93.43	94.64	94.03	78.48
Final	91.85	92.43	92.14	75.20

Figure 4.9: Effect of the priors on the parses

4.5.3 Testing the Viterbi approximation

In the Stolcke algorithm the assumption is made that the Viterbi parse (the most probable parse) is approximately preserved after every change in the grammar (see section 3.3.6). This can be tested if we keep track of the parsed sentences during the process of model merging. The ‘real’ Viterbi parse can be found by reparsing the sentences with the induced grammar. We can then match the parses obtained without reparsing with the parses obtained after reparsing, and compute the PARSEVAL values.

	UP	UR	EXM
OVIS	95.94	95.94	91.9%
WSJ10-POSTAGS	95.10	94.21	84.4%

Figure 4.10: Test of Viterbi approximation

The results of table 4.10 imply that in the vast majority of the cases the ‘tracked’ parses are identical to the Viterbi-parses, and we can cautiously conclude that the Viterbi approximation is indeed valid.

4.5.4 Description lengths of treebank grammars

Our next test was to check whether indeed, as the minimum description length (MDL) principle presupposes, the treebank grammar is the optimal grammar, that is, a global minimum of the constructed objective function.

To our surprise this was not the case.

	DL	GDL	DDL
WSJ10 Initial	357421	266355	91066
WSJ10 Final	316055	209436	106618
WSJ10 Treebank	501220	53623	447597
OVIS Initial	364565	281371	83194
OVIS Final	241523	110507	131016
OVIS Treebank	261692	48042	213650

Figure 4.11: Description lengths with uniformity assumption

As can be seen from figure 4.11 both for OVIS and for WSJ10-POSTAGS the (negative logarithm of the) structure prior of the treebank grammar is well below that of the induced grammar, which means that if we look only at the structure prior the treebank grammar has a higher probability. However, because the Data Description Length for the treebank grammar is so much higher than that of the the induced grammar, altogether the treebank grammar has a higher model description length².

We also calculated the initial and final model description lengths using the definition of likelihood from Stolcke (section 3.3.6), which assumes non-uniform distributions, while for the search algorithm we used the uniform distribution assumption as before.

	DL	GDL	DDL
WSJ10 Initial	354199	266355	87845
WSJ10 Final	312162	209436	102726
WSJ10 Treebank	280075	53623	226453
OVIS Initial	358291	281371	76921
OVIS Final	219902	110507	109395
OVIS Treebank	258985	48042	210943

Figure 4.12: Description lengths without uniform distribution assumption

²The tests were performed for geometric distribution and no Dirichlet prior

Strikingly, when calculated without the uniform distribution assumption the model description length of the treebank grammar is below that of the induced grammar. It thus seemed reasonable to drop the uniform distribution assumption and test the algorithm with the likelihood as defined in Stolcke [63]. The modified equations are worked out in section 3.5.3. For the results refer to section 4.2.2 and figure 4.8.

The fact that the DDL of the treebank grammar is higher than the DDL of the induced grammar suggests that in the treebank grammar much probability mass is reserved for sentences that are not in the training set, so that there are generalizations made in the annotation of the treebank that are not found by the BMM algorithm, which overfits the data. Possibly the annotators based themselves on linguistic knowledge that is not found explicitly in the training set.

Another possible explanation for the phenomenon that the DDL of the treebank grammar is very high is that only a few very unlikely sentences contribute the mass of the DDL. This possibility will be tested in section 4.5.6 by removing the most unlikely sentences from the training and running the induction algorithm without these sentences.

4.5.5 Running BMM on the treebank grammar

The question arises whether the objective function in its current version is indeed chosen appropriately. We therefore tested whether the treebank grammar is if not an optimum, than at least a local optimum of the objective function. For this purpose we loaded the treebank grammar into the BMM algorithm as the initial grammar, from which the induction process started. If the treebank grammar is the optimal grammar with respect to the objective function, we would expect the algorithm to stop soon after initialization. This was tested on WSJ10-POSTAGS in 8 conditions, Poisson (P) ($\mu = 2.5$), or Geometric (G), Dirichlet (D) or not, uniform distribution assumption (U) or not.

Table 4.13 shows that, without exception, BMM can still optimize the description length when initialized with the treebank grammar (labeled TARGET in the table), thus the latter is not even a local minimum of the objective function. For any of the conditions induction doesn't stop when the optimal grammar is reached, but continues at the cost of a worse F-score. The results seem to point to the fact that it is not a failure of the search algorithm which prevents the algorithm from reaching the optimal grammar, but rather an inadequate choice of the objective function.

4.5.6 Testing the algorithm without the most unlikely sentences

We want to exclude the explanation that the BMM algorithm fails because it cannot deal with the idiosyncracies of natural language. The algorithm always opts for the

		DL	GDL	DDL	UP	UR	EXM
P D U	Target	483636	66067	417569	90.10	88.64	
	Final	452406	42379	410027	64.50	74.73	16.24
P D ¬U	Target	292515	66067	226448	90.10	88.64	
	Final	275807	40557	235250	64.31	74.78	17.29
P ¬D U	Target	464178	46609	417569	90.10	88.64	
	Final	438923	29866	409057	61.62	71.68	12.20
P ¬D ¬U	Target	273057	46609	226448	90.10	88.64	
	Final	261572	27761	233811	64.82	74.94	17.00
G D U	Target	491074	73505	417569	90.10	88.64	
	Final	457445	46491	410954	64.51	74.56	16.39
G D ¬U	Target	299953	73505	226448	90.10	88.64	
	Final	280127	45374	234752	64.95	75.33	17.62
G ¬D U	Target	471616	54046	417569	90.10	88.64	
	Final	443784	33993	409791	62.15	72.11	12.05
G ¬D ¬U	Target	280494	54046	226448	90.10	88.64	
	Final	266311	32052	234259	61.30	71.19	11.98

Figure 4.13: Induction initialized with treebank grammar

most probable merges and chunks, while replacing these in the entire grammar, and thereby it ‘flattens out’ the grammar. If one believes radical construction grammar to be correct, one would expect there to be many idiosyncratic constructions that cannot easily be caught in general context free rules. We tested this hypothesis by leaving out the most unlikely sentences from the training set, in the expectancy that that would reduce the number of idiosyncratic expressions, and improve the performance of the algorithm. The sentence probabilities were computed from the treebank grammar, and normalized for sentence length by dividing by the average probability for a certain sentence length, to prevent the training set from being biased for short sentences. Here are 10 of the most unlikely sentences in OVIS:

*ik wil niet op zesentwintig december reizen want zesentwintig december is al lang
verstreken we zitten nu in februari*
*ik wil niet naar rotterdam centraal station reizen ik wil naar van den haag naar roer-
mond reizen*
*ik wil morgen om zes uur dertig vanuit vlissingen naar utrecht centraal station vertrekken
dat is dat klopt*
*ik wil op maandag morgen niet om zes uur dertig maar om negen uur reizen vanaf
centraal station utrecht*

ik zou graag willen reizen op zaterdag tien februari van rijssen naar papendrecht na acht uur smorgens
ik wil niet van den haag centraal naar deventer ik wil van den haag centraal naar bodegraven
ja ik wil nog een andere verbinding weten van van amsterdam cs en dan een uur later naar driebergenzeist
ja ik wil vertrekken van almere stad om circa tien uur naar amersfoort op dinsdag vierentwintig januari
ik wil niet met de eerste trein de eerste doorgaande trein vanuit groningen op maandagmiddag na vier uur
ik wil niet op donderdag negenentwintig ik wil op zondag vier februari voor twaalf uur in winsum zijn vanuit

We reduced WSJ10-POSTAGS and OVIS to 90% of their original sizes, by removing the most unprobable sentences according to the treebank grammar. For WSJ10-POSTAGS the reduced training set consisted of 6682 sentences (of 7422), and for OVIS of 9036 (of 10040). We then tested the algorithm with the settings $\neg P$, $\neg D$, U (Geometric, no Dirichlet, and Uniform distribution assumption).

Without the sentences that contribute most to the DDL, the DDL of the treebank of WSJ10-POSTAGS is 381005, compared to 447597 for the complete training set, which is a relative reduction of the DDL by 5.5% (after correcting for the smaller number of sentences). The DDL of the initial grammar of the induction algorithm (which incorporates all the sentences as rules) is still much lower: 71966 compared to 81959 (=90% of 91066).

The DDL of the OVIS Treebank is reduced from 214650 to 176770 after removal of 10% of the most unlikely sentences. This corresponds to a relative reduction of 8.5 % in DDL.

WSJ	UP	UR	F
90% most likely	59.24	41.53	48.83
Complete	59.26	37.84	46.19
OVIS			
90% most likely	72.21	66.84	69.42
Complete	72.08	65.41	68.58

Figure 4.14: Results with most likely sentences

From figure 4.14 it can be seen that there is a slight, though not dramatical improvement of the F-scores compared to a run with the complete training set. Moreover, this improvement is entirely due to a better recall. This means that the algorithm

discovers more correct chunks than before, a fact that is substantiated when inspecting the merges (4 in total) and chunks: {DT+JJ+NN, JJ+NNS, DT+NNS, NN+NNS} are merged together, as well as {IN+DT+NN, PRP+VBZ, IN+NN, TO+VB, PRP+VBD, PRP+VBP, IN+JJ+NNS, JJ+NNS+VBP, DT+NN+VBZ, PRP+VBZ+RB, DT+NN+VBD, NNP+NNP+VBZ, PRP+VBZ+PRP+VBZ} , {CD+CD, MD+VB, RB+VB} and {NN+VBD, NN+VBZ, IN+NNP, NNS+VBP, DT+NNP, IN+CD, NNS+VBD, PRP+VBZ+VBN, NN+VBD+RB, JJ+NNS+VBD}

4.5.7 Temporal dynamics of the grammar induction algorithm

Following the merges and chunks of OVIS and WSJ over time revealed that initially they are mostly linguistically relevant, but after a while they become distorted. The motivation for this experiment was to see whether this behavior is reflected in the PARSEVAL scores and whether it is possible to trace down a point where the induction process goes astray.

Rather than parsing the corpus again after every chunk, we kept track of the ‘Viterbi’ parses throughout the merging and chunking process, by performing the merges and chunks directly on the sentences every time the grammar was changed. The experiments were performed for WSJ10-POSTAGS and OVIS with the following settings: Poisson distribution, $\mu = 2.5$, Dirichlet prior, and non-uniform distribution.

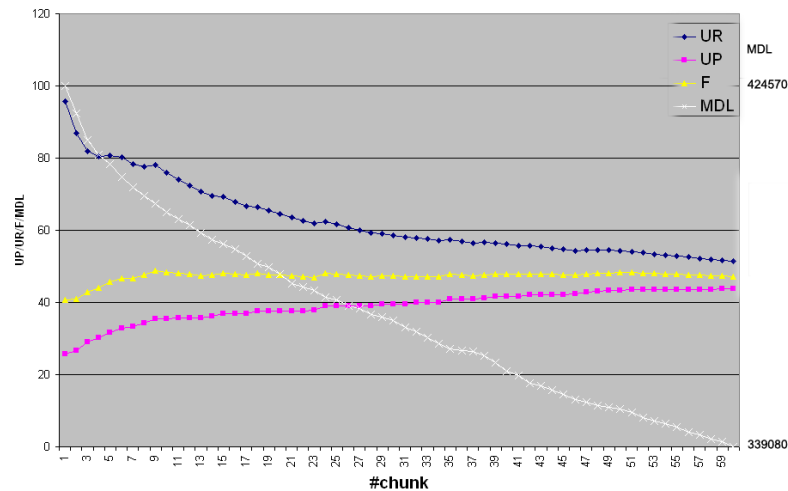


Figure 4.15: PARSEVAL scores for WSJ10-POSTAGS as function of chunk number

The graph confirms what was already apparent from watching the merges and

chunks over time: after no more than 10 out of a total of 60 chunks the F-score doesn't improve anymore, UP reaches an asymptote around chunk #20, and UR reaches an asymptote at a somewhat slower rate. Most chunks after chunk #10 thus do not improve the F-score anymore, which is also evident from inspection of the chunks. At the same time though, the DL decreases monotonically, all the way up to the last chunk.

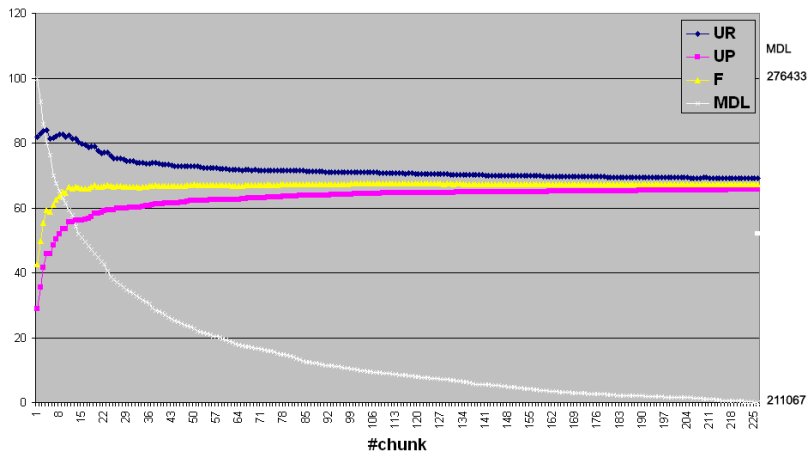


Figure 4.16: PARSEVAL scores for OVIS as function of chunk number

For OVIS we see the same trend: UP, UR and F quickly reach an asymptote. F reaches a maximum at around chunk #15 of a total of 225 chunks.

Chapter 5

Unsupervised labeling

So far we have followed an integrated approach to grammar induction in which both constituent brackets and labels were bootstrapped. Although this approach proved not to be as successful as other approaches, such as [36], [3] and [56], the latter approaches didn't cope with the labeling issue. One of the difficulties that were hard to overcome in the integrated approach was that errors from the labeling phase were carried over as input to the bracketing phase - and vice versa, causing a snow ball effect. This was due to the alternation between label induction and constituent induction, and the interdependence between those phases.

We will now try a different approach where constituent brackets are given a priori but the constituents are unlabeled. The BMM algorithm is adapted to specialize for label induction alone, and is thus turned into a unsupervised labeling algorithm, called MMULA. From here we have two options:

- We may pursue our original goal of inducing a completely unsupervised grammar by using the output from a specialized 'third party' unsupervised bracketing algorithm as input for the unsupervised labeling algorithm .
- We can settle for semi-supervised induction by supplying the algorithm with manually annotated brackets. If we assume that the constituent boundaries are known to the child prior to classification and meaning extraction, then the question whether it is possible to induce correct labels given a bracketing is still psychologically relevant.

The latter assumption is not so far-fetched - an extensive literature exists on the hypothesis that children might use prosodic cues, such as pauses, stress patterns, and intonation groups in order to detect syntactic phrase boundaries. Thus, prosody is exploited to help bootstrapping the syntax; this is usually referred to as the 'prosodic bootstrapping hypothesis'.

5.1 The prosodic bootstrapping hypothesis

It has been shown that both adults and infants from 9 months old are able to perceive prosodic phrase boundaries independently and irrespective of lexical information.

For instance, in an experiment involving Dutch adult subjects, de Pijper et al. [16] showed that subjects perceived phonological phrase boundaries, even if the lexical contents of the (Dutch) utterances were made unrecognizable, thus blocking access to lexical, syntactic, and semantic information. The adults were asked to rate the prosodic boundary strength (PBS) between all contiguous words of the sentences, and similar ratings were found whether subjects could understand the sentences or not. The fact that subjects could perceive relative strength of prosodic cues further suggests that there exists hierarchical structure within prosody.

Infants from the age of 9 months also perceive phonological phrase boundaries: Jusczyk et al. [34] tested whether infants display any sensitivity to the markers of major phrasal boundaries in English, in particular the NP VP boundary. Using the head turn paradigm, they tested whether infants preferred to listen to speech that was (artificially) interrupted at the boundary between the NP and VP syntactic phrases over speech that was artificially interrupted at any other location. They found that 9 month old infants preferred sentences that were segmented on the syntactic phrase boundary to sentences in which the phrase boundary was disrupted, but 6 month old infants did not.

Infants are even able to resolve conflicts between prosodic structures at multiple hierarchical levels, in which case the lower level prevails: Gout et al. [29] presented 10- and 13-month-old American infants with sentences that either contained familiarized bisyllabic words within a phonological phrase or contained both syllables of the same word separated by a phonological phrase boundary. When trained to turn their heads for the target word ‘paper’, infants responded to sentences which did contain the word ‘paper’, but not to sentences which contained both syllables of this word, separated by a phonological phrase boundary, as in *the butler with the least pay # performed the best*.

Despite all this, the correlation between prosodic markers and syntactic boundaries is usually far from perfect. Often, prosody marks some lower-level syntactic units (such as relative clauses) without marking the higher-level units which contain them (such as subjects and VPs) [24]. Even more common, in sentences with a pronoun subject the prosodic marker (a pause) falls after the verb, contrary to syntactic phrase structure, as in the following example:

1. Joe / kissed the dog.
2. He kissed (/) the dog.

where the slash / indicates the natural pause in the sentence.

Within the framework of prosodic phonology the difference between the sentences is explained by the fact that a weakly stressed pronoun subject is phonologically joined (cliticized) to the following strongly stressed verb.

Gerken [24] tested whether infants were able to infer correct syntactic constituents in cases where prosody is inconsistent with syntax. For this purpose, one group of infants was presented with sentences with a lexical NP subject, as in the first example, in which prosodic structure is consistent with syntactic structure. The other group heard sentences with a pronoun subject, in which prosodic structure does not reflect syntactic structure. In both groups, sentences were presented with pauses inserted either between the NP and VP or directly after the verb. In a preferential listening paradigm, infants in the first condition listened longer to materials containing pauses between the NP and VP, but not infants in the second condition (where the subject was a pronoun). Gerken concludes, that only the infants of the lexical subject group, where prosody and syntax are on a par, were able to infer syntactic structure.

5.2 A case for prosody-based sentence segmentation for the discovery of constructions and categories

Is it a problem that prosody is at odds with the syntactic constituents? Do children really need to learn conventional syntactic categories in order to acquire a language, or can they discover the categories and word meanings on the basis of prosodic constituents alone? We hope we can address this question by supplying the unsupervised labeling algorithm with a corpus of children's speech annotated with prosodic bracketing alone.

By prosodic annotation (or prosodic bracketing) we mean that we place the brackets according to suprasegmental cues, such as intonation groups, pauses and stress patterns, in the sentence. While WSJ-style bracketing, where the brackets are placed around conventional syntactic constituents, would produce a largely right branched annotation for English, the choice for prosodic bracketing would result in a largely left branched bracketing, as will be illustrated later.

5.2.1 Some reflections on the nature of constituency

By assuming prosodic bracketing we risk to enter a philosophic discussion about the nature of the relation between (syntactic) categories and the way the hierarchical structure of speech is perceived: if the perceived structure doesn't follow conventional syntactic constituent boundaries, then what is exactly the role of constituents? Conventionally, syntactic units are assumed to fulfill the substitutability requirement.

However, if the language learner employs prosodic units, is this requirement still fulfilled, and if it is not, is it still possible, and does it make sense to attach labels to prosodic units? At present, we do not know the answers to these questions.

There is however one argument in favor of prosodic bracketing that is not usually made by the proponents of the ‘prosodic bootstrapping hypothesis’: prosodic bracketing appears much better correlated with the kind of constructions we discussed in section 2.3 than syntactic bracketing, and therefore it is in our view better suited as a starting point for a construction grammar account of language acquisition. In the following sections we will explore this argument further.

Please note, that the ideas presented below are still in an explorative stage, and thus have only an informal status. Much literature research still has to be done, as the observations made here are mostly based on intuition.

5.2.2 Prosodic bracketing for compatibility with Construction Grammar

Why use prosodic bracketing? As mentioned before, the purpose of this work is to develop a technique for automatic discovery of constructions from annotated text. While syntactic constituents are often incompatible with the constructions from the construction grammar literature, prosody in most cases correlates with the boundaries of constructions.

For instance Many constructions have their fixed (and characteristic) part on the left side and the variable part on the right side. While prosody tends to capture this phenomenon, WSJ-style bracketing, being mostly right-branched, ignores it. Here are some examples from the Adam corpus (and some invented):

The *there is X* construction:

(there is) the dog

(there is) my daddy

WSJ-style bracketing would give : *there (is (my daddy))*, although the construction seems to be *there is*.

The *I don't know X* construction:

(I don't know) (his name)

(I don't know) how to do this

In WSJ *I don't know* would not be a constituent, but rather (*know (his name)*), but intuition says that the child segments the sentence using *I don't know* as a unit. The same goes for the *I'm gonna* construction:

(I'm gonna) cry

(I'm gonna) have another cookie

(I gonna be) a man today

In the last example, prosody follows the intuition that (*X gonna be*) is a single con-

struction, but WSJ-style bracketing would be something like *I (gon (na (be (a man) today)))* as is illustrated by the following excerpt from the WSJ treebank:

*(VP told (NP him) (SBAR that (S (NP-SBJ-1 we) (VP were n't (VP gon (S (NP-SBJ *-1) (VP na (VP let (S (NP-SBJ this guy) (VP beat (NP us))))))))))))))*

In all the above cases, and many more, prosodic (left-branched) bracketing coincides with the constructions. Our guess is, that the reason is, that the same process of entrenchment underlies both formation of constructions and of prosodic units: in the same way that frequent expressions are chunked together into a construction, also in prosody a sequence of words that co-occurs frequently tends to be pronounced fluently as a single unit.

In contrast, WSJ-style bracketing in most cases goes against the constructions. If one's intention is to discover psycho-linguistically motivated constructions in a corpus of spoken language, such as the Adam corpus, it might therefore be a good idea to start with prosodic bracketing.

A case in point are the verb islands from usage-based grammar. These are characterized by the use of specific forms for distinct verb islands, usually a distinct use of pronouns (in the subject position). For example, the child might have constructed one subcategory of verbs that always goes with the form *I Verb1* and uses no other pronouns, while another category always uses the form *do you Verb2?*. Then, if one wants to distinguish between these verb constructions, the bracketing must include both the subject and the verb. With conventional (WSJ-style) bracketing however the subject is separated from the verb phrase, making discovery of verb islands much harder.

5.2.3 Evidence for prosodic ‘constituents’ from contractions and from pro-drop languages

Some constructions are so frequent that their contracted form is incorporated in the spelling. Such constructions are obvious candidates for constituents, and we propose that they are bracketed accordingly. For example, the existence of the contracted form *wanna* seems to indicate that in *X (want to) Y (want to)* is a constituent. Conventional bracketing however separates *to* from *want*: *I (want (to X))*. The same holds for *gonna*, being (in our opinion) quite convincing evidence that *(going to)* is a constituent in *(I'm (going to)) play*. The fact that forms like *I'm*, *you're*, *he's* and *it's* are contracted argues in favour of taking them as constituents. In general, one can make a case for forming a constituent of the subject together with the verb, whenever the subject is short, e.g. a pronoun. Again, these contractions go against conventional syntactic bracketing, which has it that the verb forms a constituent together with the object, as in *I ('m Paul)*.

We believe that pro-drop languages, such as Italian, Spanish and Hebrew, where

the pronoun subject is incorporated in the verb form, constitute converging evidence for the view that subject and verb can form a unit. From the child's perspective it makes no difference whether subject and verb are written as two words, or as one. If they always appear as a unit, this makes them substitutable, and thus a good candidate for a constituent.

5.2.4 The two stage language acquisition hypothesis

There is also an argument for using prosodic bracketing that follows from our working hypothesis about language acquisition: We would like to view the unsupervised labeling algorithm as the second stage in a model for child language acquisition. We hypothesize that in the first stage sentences are segmented on the basis of prosodic cues. For the inference of a hierarchical bracketing from prosody we rely on relative prosodic strength, although this is not a trivial issue.

For prosodic bracketing, knowledge of the labels and of meaning is not required, making the process relatively independent of the labeling process. Conceived this way, the prosodic bracketing yields the prior knowledge that guides the child further with the discovery of categories in the labeling stage. The same argument doesn't hold for syntactic bracketing though, since syntactic brackets depend on the labels.

An interesting research question would thus be whether the presence of prosodic bracketing helps bootstrapping categories. This question can be assessed if we feed the unsupervised labeling algorithm with prosodically bracketed sentences.

We are also interested in comparing the behaviour of the unsupervised labeling algorithm to the case where it is provided with syntactic bracketing or completely left-branched or right-branched bracketing.

One possible objection against supplying the algorithm with prosodic bracketing is, that it gives away constructions for free, since many constructions are already present in the bracketing. But that could hardly be taken as an objection against using prosody, it only means that the child can infer constructions from prosodic structure alone, and we assume that the child is capable of learning the prosodic structure independently of the labeling process. Also, the focus of the research will be on constructions with variable slots, which are not explicit in the bracketing.

5.2.5 A hypothesis about the function of constituents

The approach we have taken with the unsupervised labeling algorithm suggests a certain hypothesis about the function of constituents. This hypothesis states that the main function of constituents is that, by subdividing the sentence into functional parts, they facilitate categorization (labeling), and with it meaning discovery: they enable the listener to ignore the part of the sentence outside the constituent, and regard

only the context of the constituent for extraction of the meaning and categories of the component words. Thus, constituents facilitate label extraction by shrinking the contexts. In fact, we have already encountered an argument that goes along the same lines in Steels' explanation of the function of grammar [61].

Constituents may be regarded as autonomous microscopic sentences. In principle, every constituent must have a clear meaning by itself (in the sense that it must assess something), independent of the rest of the sentence. This constraint is placed so that the constituent can fulfill its function as a context for classification of the functional parts.

In line with this definition, the child's first sentences are single-constituent sentences. Language development proceeds through an expansion process, whereby the child expands microscopic sentences into more complex sentences by combining constituents. We hypothesize that adult sentences are built up in the same way, and that this prescribes how one should bracket adult sentences as well as child sentences, roughly along the lines of prosody and constructions.

Altogether, it appears that constituents and constructions play remarkably similar roles. Both are defined as the substitutable and reusable units, that make up the language, and define its structure. The difference is perhaps only a matter of the perspective that one takes: a formal model-theoretical perspective, or a more empirically inclined, cognitive perspective.

If a comprehensive cognitive linguistic theory will ever come off the ground, the differences may well disappear. We expect, that language can then be analyzed exclusively in terms of constructions. This is also the view of radical constructionism, which disposes of (global) constituents altogether in favor of constructions [14].

5.3 The model merging unsupervised labeling algorithm (MMULA)

Let us assume that the correct bracketings of the sentences are known, either as the outcome of an alternative unsupervised bracket induction algorithm that yields sentence constituency, or from a treebank, or from manual bracketing. We will refer to these as the 'target bracketings'. How can we amend the BMM algorithm such that it exploits this prior knowledge? Since we must respect the target bracketings, we can no longer apply chunking 'across the board', as before, but we must consider application of the chunk on a sentence by sentence basis. For example, the constituent (*saw this*), as in the first example below must not be applied in the bracketing of the second example

$$\begin{array}{l}
 I \text{ never } [saw \text{ this }] \text{ before} \\
 I \text{ never } [saw [this \text{ jet}]]
 \end{array}
 \tag{5.1}$$

As for labeling, since we are ignorant about the labels and equivalence classes, and do not want to impose unwanted merges, we must initially attach a unique label to every constituent. Ideally, we could just read off all rules from the bracketed sentences, including the rules involving unique non-terminals throughout, add them to the grammar, and start the merging process. However, this leads to some major problems:

1. Due to the vast number of distinct non-terminals, one for every unique constituent, the number of merge pairs that has to be considered grows explosively.
2. Since a merge requires that two rules differ by at most one symbol, and all symbols higher up in the tree hierarchy are unique, initially the only rules that have a chance to be merged are the rules in the bottom of the tree, whose right hand sides correspond to the preterminal nodes. Therefore, merging of non-terminals higher up in the hierarchy will not get off the ground, and one might be better off by not loading them at all in the initial grammar.

The solution we opted for is to let the algorithm search for merges and chunks bottom-up as before, starting from a grammar consisting of flat rules, but constrain the search in several ways, as described below.

5.3.1 Implementation

In order to have the information of the target bracketings available (as an oracle) we keep a reference to the original sentences, and we parse them along with the grammar induction process, so we have a record of the ‘partial parse’ (PP) of every sentence. From looking at the partial parses the oracle can tell which part of the associated target parses has not been parsed yet by the algorithm at any stage in the induction process. Let us define the ‘front’ of the PP as the sentence that is obtained by partially processing of the original sentence.

Before a chunk is applied to a rule of the grammar, the oracle is consulted, and all sentences are looked up, whose associated PP fronts have evolved to the rule under consideration (e.g. by merging and chunking them in parallel with the grammar). The chunk is then applied discriminatively to those partial parses whose associated target parses license it.

If the application of the chunk for a specific rule is not across the board, because the partial parses that are represented by that rule have distinct target bracketings, it will cause to split the rule into multiple rules. This scenario may unfold when one

target bracketing licenses the chunk and another does not, as in example 5.1, or if the rule contains more than a single occurrence of the chunk, and some occurrences are licensed by some target sentences and other occurrences by others.

Only in the case that all partial parses represented by the rule license (the same occurrences of) the chunk, application of the chunk will result in a smaller grammar (due to a shorter rule). In the other cases the grammar will increase in size, due to the fact that both the new rule and the original rule must be retained (and perhaps some other variations) if the chunk is not applied across the board.

This has implications for forecasting the expected description length gain of a chunk $\Delta DL_{CHNK(X,Y)}$:

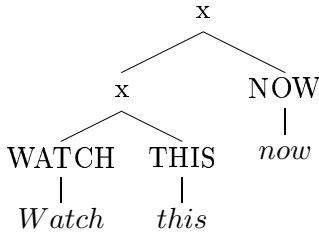
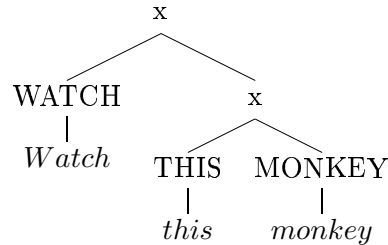
$$\Delta DL_{CHNK(X,Y)} = (A_{NT} + A_R - A_S + 2) \cdot \log\left(\frac{A_{UNT}+2}{A_{UNT}+1}\right) + (4 - BF(X, Y)) \cdot \log(A_{UNT} + 2) \quad (5.2)$$

For the computation of the bigram frequency (BF) it means that the BF is increased by 1 only in case the chunk is licensed by the oracle for all partial parses represented by the rule. If the chunk cannot be applied across the board for a particular rule (and therefore multiple variations of the rule are obtained) the BF is reduced by the number of non-terminals in the bodies of all the additional rules.

Because the bracketing is given, MMULA lets chunking go on until there are no target chunks left. The first term in equation 5.2, which was relevant in the full BMM algorithm for the stopping criterion, needs not be considered anymore. The algorithm therefore always reaches the target bracketing, with UP=UR=1.

5.3.2 Constrains on merging

How can we exploit the prior knowledge of the bracketings (the oracle) to improve the labels? Consider the two sentences



With our knowledge of the target bracketing we should be able to prevent the merge of *monkey* and *now*. We do so by allowing the bracketing to disambiguate the cases, partitioning the partial parses corresponding to a rule into distinct sets, by means of the associated target parse.

For this purpose we introduce the concept of ‘constituent front’. The ‘constituent front’ (CF) of a partial parse is defined as the set of possible chunks of neighboring pairs in the ‘front’ of the PP, that are licensed by the target parse. In other words, the CF is the collection of the immediate chunks that the target parse permits with respect to the partially parsed sentence. Formally, if the current front of the PP is given by $S_F = X_1 \dots X_i \dots X_n$, then the constituent front is

$$CF(PP, S_F) = \{ \langle i, i + 1 \rangle \mid Con(X_i, X_{i+1}) = 1, 1 \leq i < n \} \quad (5.3)$$

where $Con(X_i, X_j) = 1$ if $\langle X_i, X_j \rangle$ forms a constituent (chunk) in the target parse, and 0 otherwise.

In practice, at every stage in the induction process the original sentences are indexed by the CFs of their partial parses, which are updated after every chunk. The CF in turn partitions the rule into ‘CF Groups’ of original sentences that are bracketed one way or another.

Finally, if after consulting the oracle it is found that the sentences *Watch this monkey* and *Watch this now* belong to distinct CF Groups, they are not merged, or rather, their CF Groups are not merged.

5.3.3 Forecasting the GDL gain of a merge

As for forecasting the GDL gain from rule elimination as a result of a merge, we don’t count the actual number of eliminated rules, but rather the number of eliminated CF Groups after the merge. From our prior knowledge of the bracketing, we may consider every CF Group as constituting a potential rule, and we are therefore interested in the decrease of the number of CF Groups, rather than the decrease of the actual number of rules.

In practice, a weight is attributed to every set of duplicate rules in the grammar, corresponding to the number of CF Groups that are merged by merging the rules. In the above example, the set *Watch this monkey* and *Watch this now* gets a weight of zero. These weights are taken into account in the equation for forecasting ΔGDL_{MRG} 3.27, which we repeat here for convenience:

$$\begin{aligned} \Delta GDL_{MRG} = & (\sum_{SB_1}(|NT_R + 1|) + \sum_{SB_3}(|NT_R + 2|) + T + 2) \cdot \frac{\log(A_{UNT})}{\log(A_{UNT+1})} \\ & - (\sum_{\Omega_1}(|NT_R + 1|) + \sum_{\Omega_3}(|NT_R + 1|)) \cdot \log(A_{UNT}) \end{aligned} \quad (5.4)$$

In the second term, the summation over the duplicate rules of $S1$: $\sum_{\Omega_1} (|NT_R + 1|)$ is replaced by a weighted summation.

The equation for forecasting the change in DDL as a result of a merge, ΔDDL_{MRG} , remains without any change.

5.4 Experimental results

As before, experiments were carried out with OVIS, Wall Street Journal (WSJ) POSTAG sequences of length ≤ 10 , and the Adam corpus of CHILDES. For specifications, refer to chapter 4. For OVIS and WSJ the brackets from the treebank (but not the labels) were used as target parses. For the Adam corpus, for which an annotated treebank is lacking, we employed the brackets produced by the unsupervised dependency parser of Yoav Seginer [56]. Trees were converted to CNF, before they were presented to the induction algorithm (for the later version of MMULA, which can induce from trees with a branching factor larger than 2, this step was not necessary).

We intend to use the prosodic bracketing, discussed in the previous section, in a later stage of the research, because as yet the prosodic annotation of the Adam corpus has not been completed. All experiments were carried out with geometric structure prior, and without Dirichlet prior.

5.4.1 Evaluation

For evaluation again the PARSEVAL metric is used (see section 4.1.1), but this time with labeled precision and recall (LP and LR) rather than unlabeled. We follow Haghghi and Klein [30] who used greedy remapping of the experimental labels in order to compare them with the treebank labels. For the current application, this entails that after the induction of the labels, for every experimental label a best match is searched among the target labels, whereby it is allowed to map multiple experimental labels to a single target label.

There are some caveats with this measure, since it favors grammars with many distinct non-terminals. In the extreme case, where every non-terminal has a unique label, it allows a perfect mapping between experimental and target labels, yielding a LP and LR that equal UP and UR.

5.4.2 Results on Benchmark Tests

For OVIS the obtained PARSEVAL scores were: LP=89,5 and LR=81,5

More illustrative than the scores is to look at the categories that were found by the algorithm:

- The first category contains without a single exception, cardinal numbers, even in different dialects.
{ACHT, NEGEN, TIEN, TWAALF, EEN, VEERTIEN, TWEE, VIER, ELF, ZEVEN, NEGENTIEN, ACHTTIEN, DERTIEN, VIJF, DRIE, TWEEENTWINTIG, DRIEENTWINTIG, EENENENTWINTIG, ZEVENTIEN, ZES, TWINTIG, VIJFTIEN, ZEUVEN, VIJFENTWINTIG, ZESTIEN, VIERENTWINTIG, ACHTENTWINTIG, NEGENENTWINTIG, ZEUVENTIEN, DERTIG}
- The next category contains, with no exception, 125 place names:
{AMSTERDAM, GRONINGEN, ZWOLLE, MAASTRICHT, UTRECHT, ROTTERDAM, NIJMEGEN, *etc.*}
- There is a category containing the days of the week, with one foreign word *heel*, meaning *whole*:
{MAANDAG, ZONDAG, VRIJDAG, WOENSDAG, DINSDAG, DONDERDAG, ZATERDAG, HEEL, VRIJDAGMORGEN}
- a category containing relative reference to days (*the day after tomorrow, today, tomorrow*):
{OVERMORGEN, VANDAAG, MORGEN}
- and a category containing times of the day:
{SMORGENS, SOCHTENDS, SMIDDAGS, MORGENOCHTEND, ONGEVEER}
- a category of place names that have prefix ‘DEN’ :
{HAAG, HELDER, BOSCH, OEVER}
- a category of compound train station names:
{NOORD, LELYLAAN, NOORD, BIJLMER, SEGHWART, WTC, LAMMENSCHANS, SCHOTHORST, HOPPLEIN, TOE, ZUID}
- two categories of months:
{DECEMBER, JANUARI, FEBRUARI, APRIL, MEI, MAART}.
Weirdly, the summer months are in a separate category: {JULI, JUNI, AUGUSTUS}
- a category of confirmations {GOED (good), JUIST (just), CORRECT (correct), INDERDAAD (indeed), ACCOORD (d’accord), OK, ALSTUBLIEFT (please), DUIDELIJK (clear), ZO (so), BEGREPEN (understood), PRIMA (perfect), CHNK#NIET|NODIG (not necessary), BEDANKT (thanks), DANKUWEL (thank you very much), CHNK#DAT|KLOPT (that’s correct), CHNK#GOED|GOED, CHNK#IK|HEB|DAT|GOED (I’m right)} Subsequently, this category is intruded by a few foreign words.

- a category containing *yes* and *no*: {JA, NEE, NEEN}
- prepositions are scattered over a few small categories: {UIT, VANAF} and {ROND, OM, NA}

We may conclude that for OVIS MMULA's results on POSTAGGING are close to perfection.

As for the categories higher in the hierarchy: According to greedy label remapping, 20 of about 100 induced categories are best associated with prepositional phrases, 13 with noun phrases, 5 with verb phrases and 6 with the TOP constituent. Here are some examples of categories, which are best associated with VP:

```
MRG#WIL|VAN|GRONINGEN|NAAR|GRONINGEN|VERTREKKEN
MRG#WIL|VAN|VAN|GRONINGEN|GRONINGEN|NAAR|GRONINGEN
MRG#WIL|VERTREKKEN|VAN|ELST|NAAR|GRONINGEN
```

Categories are named after the most frequent member, and within a categorie member another category can be absorbed, as is the case with the GRONINGEN category, which contains all 125 cities. The first category thus consists of all sentences of the form: *want to leave from city X to city Y*, which is an acceptable VP. Here are some examples of induced categories that are best associated with prepositional categories:

```
MRG#NAAR|GRONINGEN
MRG#NIET|VAN|GRONINGEN
MRG#OM|ACHT|UUR|VERTREKKEN
MRG#OP|ACHT|JANUARI|VAN|GRONINGEN|NAAR|GRONINGEN
MRG#OP|ACHT|JANUARI
MRG#VAN|ELST|NAAR|GRONINGEN
MRG#VAN|ELST
MRG#VAN|GRONINGEN|CENTRUM|NAAR|GRONINGEN
MRG#VAN|GRONINGEN|CENTRUM
```

Again, from inspecting the POSTAG categories, we know that GRONINGEN stands for any of 125 cities, ACHT represents any number, and JANUARI represents any month. The above example categories thus express sentences such as *from city X, from city X to city Y, on the nth of the month X, at n a'clock leave*. These are mostly valid prepositional phrases.

Although we have not done a quantitative analysis, the impression is that also the 'higher' syntactic induced categories correspond quite well to the treebank categories.

5.4.3 MMULA Results on WSJ

As usual, the scores on WSJ are less good than on OVIS, WSJ being a much more heterogeneous corpus. With the ‘uniform distribution assumption’ we obtained scores of LP=LR=48.0. This is far below the scores of [30], who reported for their Prototype-Driven Grammar Induction an F-score of 71.1 on the treebank brackets.

Dropping the ‘uniform distribution assumption’ (see section 3.5.3 for details) significantly improved the scores. Without ‘uniform distribution assumption’ LP=LR=52.2 were obtained.

	LP	LR	F
uniform	48.0	48.0	48.0
non-uniform	52.2	52.2	52.2
unrestricted branching	57.6	57.6	57.6
Haghighi & Klein '06	64.8	78.7	71.1

Figure 5.1: Comparative PARSEVAL-scores on WSJ10

5.4.4 Induction from trees with branching factor greater than 2

The WSJ treebank employs a rather flat annotation. Among others, noun phrases are usually annotated flat, as in *She (saw (a green crocodile))*, and adverbial expressions are included in the VP, as in *He (had (no answers) then)*.

A quick test of the branching statistics of OVIS and WSJ10 reveals that unlike OVIS, which is almost entirely binary branched, in WSJ10 19 % of the branches are ternary, and 3.0 % of the nodes have 4 daughter nodes (see table 5.2).

# of branches	OVIS	WSJ10-postags
2	23080	27287
3	660	6758
4	96	1045
5	28	160
6	0	39

Figure 5.2: Branching characteristics of OVIS and WSJ10

Since the current implementation of the BMM algorithm could only induce binary rules, we needed to convert the target parses to binary format, using a right-branching CNF-conversion. However, this leads to linguistically undesirable analyses. We noted that in WSJ10 sentences ending in adverbial expressions (such as *now, then, at work,*

etc. or prepositions *You (must (take (your hat) off))* usually have constituents containing 3 or 4 non-terminals. For these sentences right-branching CNF-conversion entails that the adverbial expression is attached to the preceding NP rather than to the VP, which leads to nonsensical constituents, such as *(him up)*, *(it off)*, *(them there)*, *(me tomorrow)* etc. The problem doesn't occur for adverbs residing on the left side of the sentence, because then right-branching CNF-conversion attaches them correctly.

From these deliberations, we expected that removing the restriction on the branching factor from the MMULA algorithm would considerably enhance performance on WSJ and on CHILDES.

We implemented a generalization, such that MMULA could induce from trees containing nodes with branching factor 3 or 4, and removed from WSJ10 199 sentences that contained branches of more than 4 sister nodes. Indeed, the scores on WSJ improved significantly: from LP=LR=52.2 with the branching factor restricted to 2 (and CNF conversion of the target parses), to LP=LR=57.6 without the restriction (and no CNF conversion).

5.4.5 Results on CHILDES

Even though for the Adam corpus the unsupervised labeling algorithm couldn't draw its bracketings from a treebank, and bracketings were generally not very reliable, the performance on POSTAGGING was reasonable. In the 'non-uniform distribution' condition the main part-of-speech categories, nouns, verbs and adjectives, are neatly separated. (This was not true however in the 'uniform distribution' condition).

Generally, very many small categories are formed, some of which contain semantically related words; meaningful categories are preserved till the end, unlike the case with the completely supervised BMM algorithm, where categories collapsed into one supercategory. Here are some examples of 'semantic' categories: An animal category: {FISH, MONKEY, DOG, BROTHER, SNAKE, DOGGIE, BEAR}, body parts: {FINGER, HAND, ARM, HANDS, HAIR, MRG#STATION, NEST}, colors plus cardinal numbers: {BLACK, YELLOW, GREEN, BLUE, GOOD, SIX, EIGHT, TEN, BROWN, THREE, RED}.

Few function word categories are formed: {DAT'S, THAT'S, DERE'S, HERE'S}, {DAT, DIS} and {IT, THIS, THAT}, and modal verbs: {CAN, WILL, DIDN'T, MIGHT}. The adverbs, prepositions, and pronouns are not categorized very well.

5.4.6 Preliminary results with the push 'n pull algorithm

Figures 5.4.6 and 5.4.6 show some examples of the most frequent abstract constructions of depth>1 found by the push 'n pull algorithm. Among them are *I'm gonna X*, *I want ta X*, *I wonder X*, *I'm not X*, *Dis is X*, *Dese are MRG#SCISSORS*.

MRG indicates that at that position there is a variable slot, to be filled in by any of the members of the merge category, which carries the name MRG#SOME-NAME. Categories are named after the most frequent member. For instance, the MRG#SCISSORS category contains many different nouns, among which {*noodles*, *scissors*, *lots*, *puzzle*, *air*, *page*, *milk*}. The MRG#PLAY category contains different VPs, among others {*play*, *do it*, *open it*, *say it*, *see it*, *break it*, *cut it*, *turn it*, *keep dis*, *show you*}. The MRG#CREAM category contains the words {*hands*, *colors*, *clock*, *machine*, *no more*}, the MRG#HORSE category contains primarily animals, but also *mother*: {*fish*, *cat*, *ring*, *door*, *dog*, *doggie*, *monkey*, *snake*, *umbrella*, *mother*}, MRG#PAPER consists of {*paper*, *water*, *dese*} and so on.

It should be noted that the experiments are still in the testing phase, and the categories aren't optimally induced. Even though, many linguistically interesting constructions pop up in the top of the list, such as *I'm gonna X*, *I want ta X*.

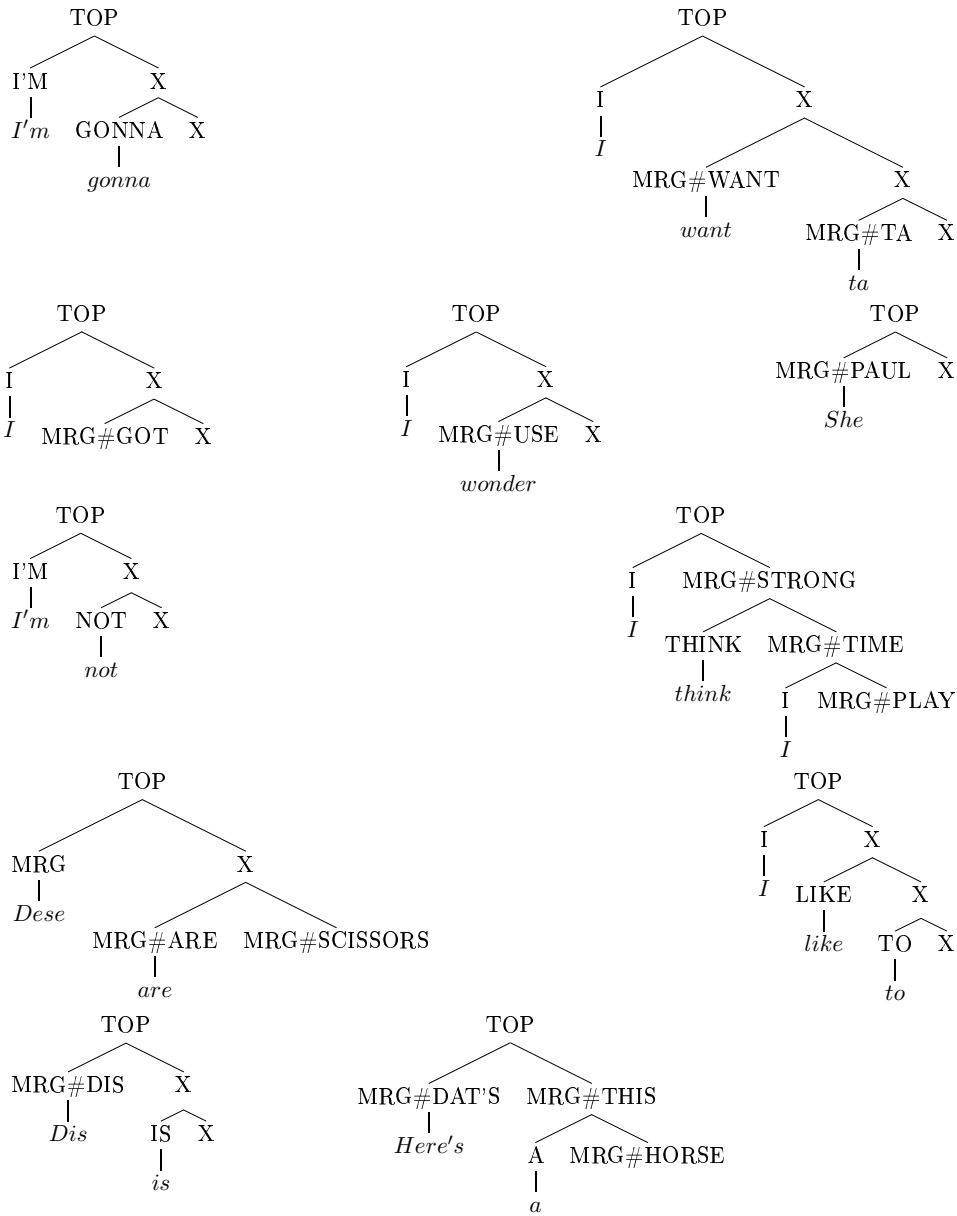


Figure 5.3: Frequent constructions in Adam corpus-1

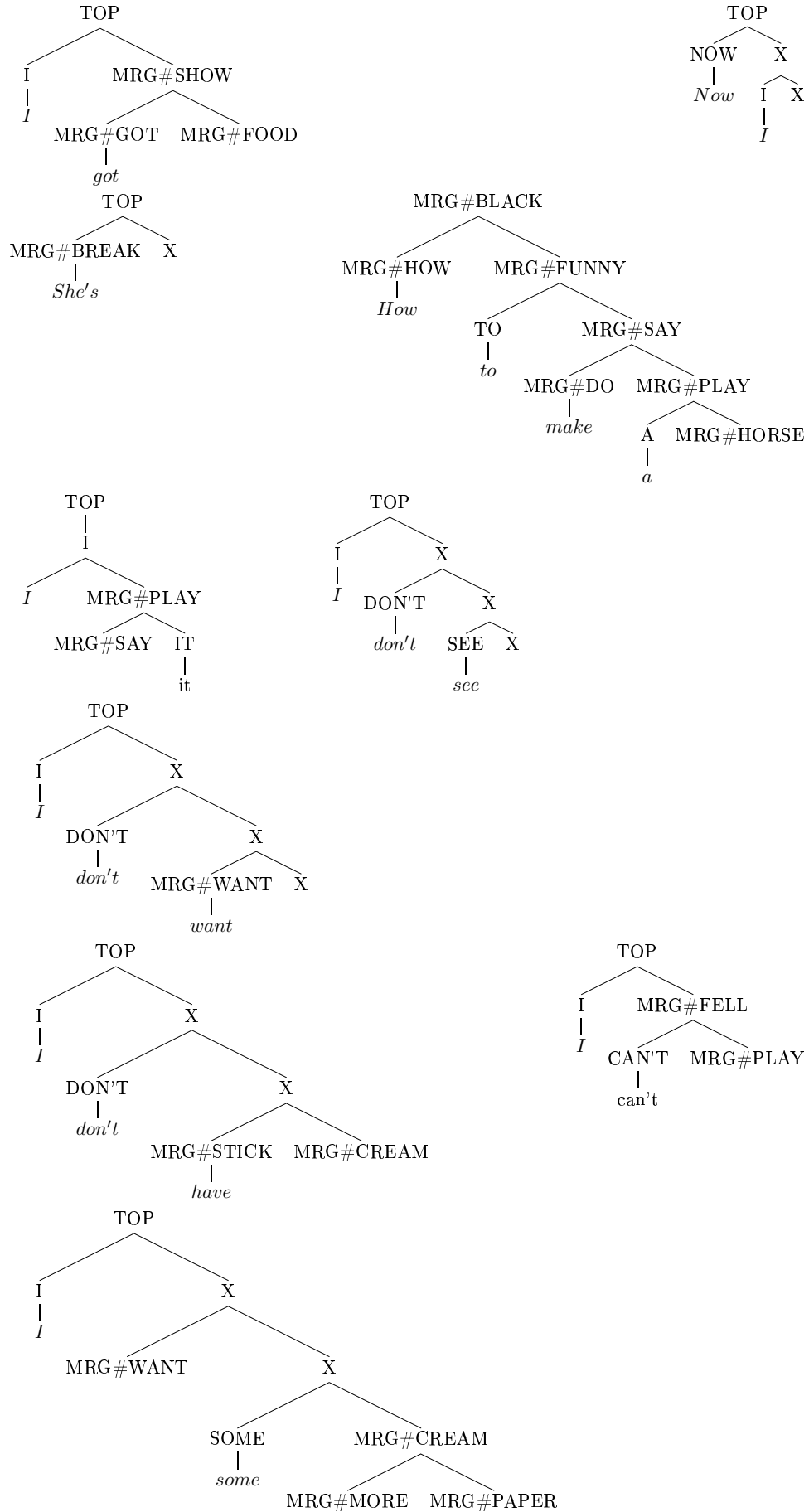


Figure 5.4: Frequent constructions in Adam corpus-2

Chapter 6

General discussion and suggestions for future work

The novel contribution of the current work consists of the application of the BMM algorithm to real world sized corpora of natural language, and the evaluation of the algorithm on standard bench mark tests. It was demonstrated to be computationally feasible to induce grammars on training sets as big as 50000 sentences having a vocabulary of tens of thousands of distinct words, within a manageable time and space complexity. This opens prospects for future studies involving grammar induction on realistic corpora.

One of the interesting findings reported here is perhaps the discovery that the treebank grammar of a natural language corpus such as WSJ is not a (local) minimum of the objective function, as is assumed by the Bayesian learning paradigm. We showed that the BMM algorithm can still optimize the objective function further, even if it is initialized with the treebank grammar, while the F-scores deteriorate, resulting in a sub-optimal induced grammar that deviates from the treebank grammar. This result suggests that either the proposed form of the objective function is inappropriate, and should be rethought, or that altogether the minimal description principle is not an adequate heuristic for guiding the search for natural grammars. After all, in construction grammar it is believed that a cognitively plausible grammar stores redundancies, so it is not minimal in any sense.

Our tests revealed that a weak point of the BMM algorithm is the discovery of constituency through chunking. Far too often, distituent, such as *of the* are not prevented from entering the grammar. Other work [37], [56], [3] has reported a fair amount of succes in this respect, but unlike the BMM algorithm, those algorithms do not deal with labeling. The CCM model [37] for instance uses both the yields and the contextual information of every constituent and distituent to find the bracketing of

the sentence (see section 3.6 for a detailed discussion of the CCM model). The BMM algorithm however exploits only the yield (that is, the bigram frequency) directly to find the best chunk. Contextual information is implicit in the merged categories. However, since merging sometimes produces overly general or noisy categories, that contextual information is lost by the time the algorithm starts chunking.

The question presents itself if perhaps it is too much to be asked from a single algorithm to deal at the same time both with bracketing and with labeling: as we saw, this causes a snow ball effect, in which errors from the merging phase accumulate and are carried over to the chunking phase and vice versa.

As a remedy, we resorted to a two-stage induction process, where we take the best of two worlds: the bracketings are induced by a specialized algorithm, such as [56], and the BMM algorithm was converted to an unsupervised labeling algorithm (MMULA), which operates on sentences with given bracketings.

Certainly, this approach trivially ‘solved’ the problem of the constituents, and led to much improved labels on OVIS, WSJ and CHILDES. Yet, there still remain many obstacles to be removed: the unsupervised bracketing algorithm does not do a very good job on the Adam corpus, and therefore it is required to perform manual corrections on the brackets before offering them as input to MMULA.

A further consideration that came across our mind was whether it is desirable or not to use conventional syntactic bracketing if the purpose of the research is to automatically discover constructions.

In chapter 5.2 we reflected on the possibility of annotating the Adam corpus using prosodic bracketing rather than syntactic bracketing. Aside from the fact that prosodic bracketing seems to be better correlated with the constructions from the literature, a bonus is, that prosodic bracketing can be forwarded as an argument in defense of the two-stage induction approach, because prosody may be considered a priori knowledge.

For this aim we even developed an interactive interface for semi-automatic annotation of brackets, integrated with the BMM algorithm. Conclusions on the subject of unsupervised labeling using prosodic bracketing are still premature, as we have not yet completed the prosodic annotation of the Adam corpus. Many issues regarding the nature of prosodic bracketing still need to be resolved. Also, at present MMULA makes sub-optimal use of the known bracketings to infer the labels, and work on the development of MMULA is still in progress.

Another major problem with the BMM algorithm is the ‘across the board’, global application of merges. Although across the board chunking is prevented in the unsupervised labeling condition, across the board merging is not. A fair proportion of the words in natural language corpora are ambiguous, belonging to multiple syntactic categories. For example, in the Adam corpus there is an abundance of words that can be classified both as verbs and as nouns, to mention a few: *love, mind, brush, call, dance, lock, mix, move, paint, press, screw, start, stick*. But there are also many less

obvious cases of words that play different roles in different contexts, e.g. prepositions can appear before a noun phrase, but also as an adverb at the end of the sentence, as in *She is going down*. Indicatives, such as *this, that, these, those* can play the role of an adjective, as in *Do you like this movie?*, but also the role of a noun, as in *I want this*. One would like to differentiate these cases by classifying tokens that occur in different contexts into distinct categories. Enabling association of a type with multiple categories helps preserve information about the training data that would get lost, if one substitutes all tokens of the same type with a single category ‘across the board’ in the entire corpus.

The present implementation of BMM does not allow for assignment of multiple categories to a single type, e.g. *love - verb* and *love - noun*, although the algorithm is sometimes capable to detect such cases, based on distinctive bracketings: refer to example 5.1. As a consequence, the implementation suffers from over-generalization.

An additional challenge is the fact that the child’s grammar (or constructicon) is not static, but in constant flux. Moreover, the dynamics of the grammar are precisely the topic of the current research. It thus seems appropriate for future work to implement an incremental version of the BMM algorithm, which parses sentences in chronological order. The incremental algorithm first attempts to parse the incoming sentences with the rules of the existing grammar, and only to the extend that parsing is unsuccessful new rules are added to the grammar.

An incremental induction algorithm would be cognitively more plausible as a model for language learning than the current BMM algorithm, which induces the grammar based on global statistics. However, since it has only access to partial information, such an algorithm must be allowed to draw back or correct its decisions, e.g. one must build in mechanisms to constrain overgeneralization. Wolff [70] proposes to monitor the use of categories and shrink, or rebuild the category for a certain context if in that context not all its members are used.

Many of the problems discussed so far, and in particular the matter of constituent annotation, may appear to be side-effects of our decision to induce a STSG grammar in two phases, via a PCFG grammar, and can possibly be circumvented if we could manage to induce a STSG directly. So far we have refrained from doing so, because it seemed too complicated to formulate learning operators for inducing tree fragments from flat sentences, and criteria for evaluating them within the Bayesian framework.

Nevertheless we’ll share some of our preliminary thoughts on this subject with you here. As a first attempt one could try to induce fragments on a corpus with a known annotation. One can formulate conditions within the BMM framework that decide when it is profitable to analyze a sentence in terms of fragments bigger than a single CFG rule. Typically, this would be the case if the fragment is very frequent, so using it as a whole rather than reducing it to flat CFG rules would reduce the likelihood of the corpus.

As for the learning operators, we propose to introduce a ‘split’ operator on the unprocessed tree for extracting functional units, rather than a chunk operation on words. The incentive to split may come either from prosodic cues, or from alignment of the sentence (or parsing) with previously discovered sentence fragments. In the latter case the fragment used to split the sentence serves as a context for classifying the remainder of the sentence into a category, much like the merge operator.

Top-down discovery of linguistic units, by splitting, is motivated in the first place, because it better settles with the research on language acquisition of Peter’s [48] and Tomasello [66]. From Ann Peters work [48] we know that children learn the linguistically relevant and substitutable units of the language by splitting up complete sentences into parts, rather than by chunking words together. Further, we may not assume that children know the word boundaries a priori.

In the second place, for extracting categories and meaning it makes sense to discover constituents from top to bottom, and from the whole to the parts. The reason is, that in order to discover structure within constituents by means of alignment and comparison between constituents, one must have previously classified the constituents used for the comparison, so that constituents of the same kind are aligned with each other. That would prevent wrong inferences such as would result from aligning *You paint* and *blue paint*. In the current implementation, which employs bottom-up induction of categories, these kinds of misclassifications cannot be prevented, unless one looks only at the context of the entire sentence.

Bibliography

- [1] B.K. Bergen and N.C. Chang. Embodied construction grammar in simulation-based language understanding. *Technical Report 02-004, International Computer Science Institute, University of California at Berkeley.*
- [2] R. Bod. *Beyond Grammar: An Experience-based Theory of Language.* Stanford: CSLI Publications, 1999.
- [3] R. Bod. An all-subtrees approach to unsupervised parsing. *Proceedings of the 21st International Conference on Computational Linguistics*, 2006.
- [4] J. B. Carroll. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf.* MIT Press, Cambridge, MA, 1950.
- [5] N. C. Chang. Learning grammatical constructions. *Thesis proposal, University of California at Berkeley*, 2001.
- [6] N. Chomsky. *Syntactic Structures.* Mouton, the Hague, 1957.
- [7] N. Chomsky. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, 1965.
- [8] N. Chomsky. *Cartesian Linguistics.* Harper & Row, New York, 1966.
- [9] N. Chomsky. Rules and representations. *Behavioral and Brain Sciences*, 1980.
- [10] N. Chomsky. *Lectures on Government and Binding.* Foris, Dordrecht, 1981.
- [11] N. Chomsky. *Knowledge of Language.* Praeger, Berlin, 1986.
- [12] N. Chomsky. *The Minimalist Program.* MIT Press, Cambridge, Mass., 1995.
- [13] F. Crick and C. Koch. Consciousness and neuroscience. *Cerebral Cortex*, 1998.
- [14] W. Croft. *Radical Construction Grammar.* Oxford University Press, Oxford, 2001.

- [15] P. Davidson. Concept acquisition by autonomous agents. *Cognitive Studies*, Vol. 12, 1992.
- [16] J.R. de Pijper and A.A. Sanderman. On the perceptual strength of prosodic boundaries and its relation to suprasegmental cues. *in: Journal of the Acoustical Society of America*, 1994.
- [17] Marcus M. et al. The penn treebank: Annotating predicate argument structure. *in: ARPA Human Language and Technology Workshop*, 1994.
- [18] C.J. Fillmore and C.F. Baker. Framenet: Frame semantics meets the corpus. *in: Linguistic Society of America*, 2000.
- [19] C.J. Fillmore and P. Kay. The goals of construction grammar. *Berkeley Cognitive Science Program Technical Report no. 50*, 1987.
- [20] C.J. Fillmore, P. Kay, and M. O'Connor. Regularity and idiomaticity in grammatical constructions: the case of let alone. *Language*, 1988.
- [21] J. Fodor. *The Modularity of Mind*. MIT Press, Cambridge, 1993.
- [22] J. Fodor, A. Walker, and C. Parkes. Against definitions. *Cognition*, 1980.
- [23] G. Frege. Über sinn und bedeutung (on sense and reference). *Zeitschrift für Philosophie und philosophische Kritik*, 1892.
- [24] L. Gerken, P.W. Juszyk, and D.R. Mandel. When prosody fails to cue syntactic structure: 9-month-olds' sensitivity to phonological versus syntactic phrases. *Cognition*, 1994.
- [25] E.M. Gold. Language identification in the limit. *Information and Control*, 1967.
- [26] A.E. Goldberg. Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences*, 2003.
- [27] A.E. Goldberg. *Constructions in Context*. Oxford University Press, 2006.
- [28] A.E. Goldberg, D.H. Casenhiser, and N. Sethuraman. Learning argument structure generalizations. *Cognitive Linguistics*, 2004.
- [29] A. Gout, A. Christophe, and J. Morgan. Phonological phrase boundaries constrain lexical access: Ii. infant data. *Journal of Memory and Language*, 2004.
- [30] a. Haghighi and D. Klein. Prototype-driven grammar induction. *Cognitive Psychology Proceedings of the 21st International Conference on Computational Linguistics, Sydney*, 2006.

- [31] A. Hodges, V. Krugler, and D. Law. A corpus study on the item-based nature of early grammar acquisition. *Colorado Research in Linguistics.*, 2004.
- [32] D. H. Hubel, T. N. Wiesel, and S. LeVay. Plasticity of ocular dominance columns in monkey striate cortex. *Phil. Trans. Royal Society London*, 1977.
- [33] R. Jackendoff. *Foundations of Language. Brain, Meaning, Grammar, Evolution.* Oxford University Press, 2002.
- [34] P.W. Jusczyk, D.G. Kemler Nelson, K. Hirsh-Pasek, L. Kennedy, A. Woodward, and J. Piwoz. Perception of acoustic correlates of major phrasal units by young infants. *Cognitive Psychology*, 1992.
- [35] S. Kirby. Spontaneous evolution of linguistic structure: An iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 2001.
- [36] D. Klein and C.D. Manning. A generative constituent-context model for improved grammar induction. *Proceedings ACL 2002, Philadelphia*, 2002.
- [37] D. Klein and C.D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. *Proceedings ACL 2004, Barcelona*, 2004.
- [38] D. Klein and C.D. Manning. Natural language grammar induction with a generative constituent-context model. *Pattern Recognition 38*, 2005.
- [39] G. Lakoff. *Women, Fire, and dangerous things: What categories reveal about the mind.* University of Chicago Press, 1987.
- [40] G. Lakoff and M. Johnson. *Philosophy in the Flesh.* Basic Books, New York, 1999.
- [41] E. Lieven, H. Behrens, J. Speares, and M. Tomasello. Early syntactic creativity: a usage-based approach. *Journal of Child Language*, 2003.
- [42] E.V.M. Lieven, J.M. Pine, and G. Baldwin. Lexically-based learning and early grammatical development. *Journal of Child Language*, 1997.
- [43] B. MacWhinney. *The CHILDES project: tools for analyzing talk.* Lawrence Erlbaum, 2000.
- [44] C.D. Manning and H. Schütze. *Foundations of Statistical Language Processing.* The MIT Press, Cambridge Massachusetts USA, 2000.

- [45] J.L. McClelland, D.E. Rumelhart, and the PDP Research group. *Parallel distributed processing: explorations in the micro-structure of cognition, Vol 2: Psychological and biological models*. The MIT Press, Cambridge Massachusetts USA, 1986.
- [46] M. M. Merzenich. Basal forebrain stimulation changes cortical sensitivities to complex sound. *Neuroreport*, 2001.
- [47] G. Petasis and et al. e-grids: Computationally efficient grammatical inference from positive examples. *Grammars*, 2004.
- [48] A. Peters. *The units of language acquisition*. Cambridge University Press, 1983.
- [49] J. Piaget. *The Construction of Reality in the Child*. 1954.
- [50] G.K. Pullum and B.C. Scholz. Empirical assessment of stimulus poverty arguments. *Linguistic Review*, 2002.
- [51] D. Purves. *Neuroscience*. Sinauer Publishers, Massachusetts, 2004.
- [52] W. Quine. *Word and Object*. MIT Press, Cambridge, 1960.
- [53] E. Rosch and C. Mervis. Family resemblances: studies in the internal structure of categories. *Cognitive Psychology*, 1975.
- [54] R. Scha, R. Bod, and K. Sima'an. A memory-based model of syntactic analysis: data-oriented parsing. *Journal of experimental and theoretical artificial intelligence*, 1999.
- [55] R. Scha, R. Bonnema, R. Bod, and K. Sima'an. Disambiguation and interpretation of wordgraphs using data oriented parsing. *Technical Report 31, NWO, Priority Programme Language and Speech Technology*, 1996.
- [56] Y. Seginer. An unsupervised dependency parser. *Unpublished manuscript*, 2006.
- [57] K. Sima'an. Efficient disambiguation by means of stochastic tree substitution grammars. ?
- [58] L. Steels. *The Talking Heads Experiment. Words and Meanings*. VUB, Brussels, 1999.
- [59] L. Steels. The emergence and evolution of linguistic structure: From lexical to grammatical communication systems. *Connection Science*, 2005.
- [60] L. Steels. The role of construction grammar in language grounding. *Unpublished manuscript, AI Laboratory, VU Brussel*, 2005.

- [61] L. Steels. What triggers the emergence of grammar. *Proceedings of the Second Int. Symposium on the Emergence and Evolution of Ling Communication EELC*, 2005.
- [62] L. Steels and F. Kaplan. Bootstrapping grounded word semantics. In: *Briscoe, T. (ed.) Linguistic evolution through language acquisition: formal and computational models*, 1999.
- [63] A. Stolcke. Bayesian learning of probabilistic language models. *PhD Thesis, University of California at Berkeley*, 1994.
- [64] M. Tomasello. *First verbs : a case study of early grammatical development*.
- [65] M. Tomasello. Do young children have adult syntactic competence? *Cognition*, 2000.
- [66] M. Tomasello. First steps toward a usage-based theory of language acquisition. *Cognitive linguistics*, 2000.
- [67] M. Tomasello and P. Brooks. Young children's earliest transitive and intransitive constructions. *Cognitive Linguistics*, 1998.
- [68] M. Tomasello and P. Brooks. Early syntactic development: A construction grammar approach. In: *Barrett, M. (ed.) The Development of Language Psychology Press, London. pp*, 1999.
- [69] M. Tomasello and P. Brooks. How children constrain their argument structure constructions. *Language*, 1999.
- [70] G. Wolff. Language acquisition, data compression and generalisation. *Language and communication*, 1982.
- [71] Z. Zolan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised learning of natural languages. *PNAS*, 2005.
- [72] A. Zollmann. A consistent and efficient estimator for the data-oriented parsing model. *Master of Logic Thesis, ILLC, UvA*, 2004.
- [73] W. Zuidema. What are the productive units of natural language grammar? a dop approach to the automatic identification of constructions. *Proceedings of the 10th Conference on Computational Natural Language Learning*, 2006.

Appendix A

The philosophy of the Talking Heads experiment

This appendix gives an exposure of some basic questions and theories from the area of philosophy of language. Subsequently we will sketch the philosophical assumptions underlying the Talking Heads experiment, and how the Steelsean framework proposes to deal with some of the questions in philosophy of language.

A.1 Some concepts in philosophy of language

Introduction

The philosophy of language is preoccupied with the relations between the *world* (or reality), *thought* and *language*. This relationship is often illustrated by the so-called triangle of meaning (figure A.1). In the figure *object* stands for *world*, *concept* for *thought*, and *symbol* for *language*.

It can be observed that *thought* (or *meaning*) is the black box, or hidden variable in the triangle, and most philosophical questions regard the problem of *meaning*. Here are some examples, each of them have been discussed extensively in philosophy literature. We will say much more about those questions in the course of the paper:

Questions in the philosophy of language

- How is the world represented in the brain?
- What are conscious thoughts made of?
- Do thoughts have the same internal structure as language? (*Language of Thought Hypothesis*)

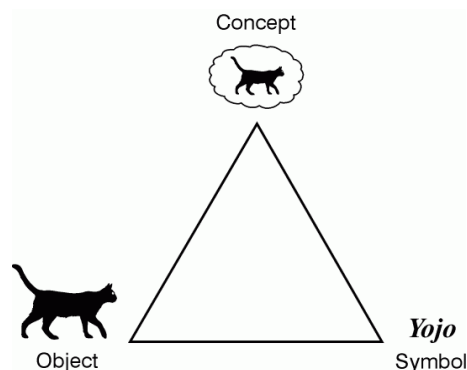


Figure A.1: Triangle of meaning

- Does language shape how to conceptualize the world, does it shape meaning? (*Sapir-Whorf hypothesis*)
- How can different individuals understand each other, if each individual acquires concepts autonomously? (*Quine, Gavagai-problem*). How do the concepts of different minds get coordinated?
- How are concepts acquired? Can concepts be acquired at all? (*learning paradox*)
- How do we know the objects in the world? Do objects pre-exist in the world; are there objects at all? (*Kant, Ding an Sich*).
- How do we know the properties of objects? And why are properties universal?

Sense and Reference, Sinn und Bedeutung

Let's start with an observation of the German philosopher Gottlob Frege [23], which has crucially influenced thinking in the philosophy of language, and which will also play an important role in the present discussion. At first thought, one might want to define the meaning of a word as its direct referent in the world. So, when one talks of a 'ball' its meaning is the object out there in the world that we perceive to be the ball. But, Frege observed that things are not so simple:

His famous example is the sentence 'the morning star = the evening star'. This sentence refers to the discovery that the star that was seen in the morning (*Phosphorus*) was in fact the same as was seen in the evening (*Hesperus*). If we would substitute the referents of the morning star and evening star on both sides of the equality sign, we would have $X=X$, and this would render the statement trivial. It doesn't convey any

information anymore. Frege therefore proposed to distinguish *reference* (*Bedeutung*) from *sense* (*Sinn*). The reference is, like before, the object in the outside world. The sense is the mode of presentation of the referent, the way one thinks of it, so it is embedded within the framework of thought.

Consequently, if you interpret meaning as *sense*, the sentence ‘the morning star = the evening star’ suddenly makes sense. It is the discovery that two objects which are viewed from within different contexts are actually the same object.

A *sense* thus expresses a *thought* or a *concept*, and it is needed to postulate it for understanding language, because typically in a language there can be multiple *concepts* that refer to the same thing, but which need to be expressed by different words depending on the context.

Anti-psychologism in formal logic

But Frege’s interest was only in public, objective thought and reasoning. He claimed that thought is *objective* and independent of individual minds, and that the *private* aspects of thought are not worth philosophical investigation, since they are not accessible and not communicable anyway. He concludes that the only thoughts worth investigating are the thoughts that can be shared, which is tantamount to language. It follows that studying language is a window to studying (objective) thought.

Therefore the meaning which is studied by Frege and his followers is stripped off all subjective aspects. This is the so-called *anti-psychologistic* view of meaning. Frege’s views started the linguistic turn in philosophy, which is characterized by the belief that through the logical analysis of language one can come to an understanding of concepts, thought and reason. It is believed that (formalized) language and (objective) thought have identical structure. Many types of logic and linguistic theories were and are still inspired by Frege’s ideas, among them various kinds of intensional logics, Lambda Calculus and Montague grammar, formal semantics, and formal theories of grammar. In the following we will refer to the collection of these theories loosely as *formal logic*.

However, by dealing only with objective, *public* concepts, and ignoring individual minds and the existence of *private* concepts, it is hard to see how such formal theories can account for the acquisition of concepts. In fact, Frege’s interpretation of a *concept* as a mind-independent and objective *sense* leads to a paradox in concept learning, which is usually associated with Fodor, who first formulated it [21]. In general, we don’t think about concepts as *objective* thoughts, but rather as a *private* entities. But in Fregean, anti-psychologistic thinking, private concepts are excluded from the theory.

Private concepts and consciousness

Although not the subject of formal logic, we will shortly discuss the cognitive perspective on concepts. This is also the view on concepts that is embraced in the TH experiment. It is generally accepted, that in order to pursue goals and to plan future actions efficiently, an intelligent organism must be able to classify and reason about objects, behavior and events. [15], [13]. As a solution nature equipped intelligent beings with a representational system, that serves as an internal model of the world they live in.

We share the view of [13] that this conceptual system is precisely what constitutes the system of *consciousness*. The function of *consciousness* is to be able to interpret a situation quickly, to anticipate to situations, to adapt to changing environments. Conscious thought operates through *concepts*, which are stored in semantic memory. The *concepts* are internal mental representations of the objects in the world, within an individual's representational model of the world, his consciousness.

Nativist and empiricist stand on concept acquisition

How can concepts be acquired? According to the *nativists*, of whom Fodor is the main protagonist, concepts are not acquired, but innate. The environment functions to trigger concepts that already exist in the mind. However, it is hard to believe this. There are many concepts, like computer, or play mobile, that are entirely contingent on cultural factors.

The other main position is that of the *empiricists*. They believe that all knowledge comes from experience. So, although complex concepts may be constructed from primitive concepts by some kind of a combinatorial mechanism, ultimately there must be some primitive concepts on top of which all other concepts are build. Since the primitive concepts cannot be formed out of combinations of other concepts, they must be *sensory*: they are acquired by the senses through induction, generalization from examples, which is a purely *bottom-up* process.

The learning paradox

So here is the learning paradox, which lead Fodor to believe that concept learning is impossible, and therefore concepts must be innate. Remember, that Fodor is an adept of formal logic, so he holds a Fregean view of concepts:

1. Concept learning should be hypothesis testing and confirmation. The hypothesis must be formulated in terms of the existing concepts in the conceptual system.
2. We cannot formulate a hypothesis for a primitive concept, unless we use other concepts in the hypothesis. But that would be a circular definition.

What Fodor is actually saying is that for learning new concepts, you have to be able to represent them in terms of the existing concepts. Learning is incorporation into the existing framework (scheme of thought). Primitive concepts cannot be incorporated, because they themselves serve as the building blocks. The reason you have to form a hypothesis is because you have to fit the concept into your existing representational scheme. This is what Fodor calls *inferential role semantics*: the meaning of the concept is (in part) determined by its relation to other concepts. And this is essentially what makes the concept a Fregian *sense*.

Another way to understand this: primitive concepts, if they are to be integrated in a conceptual scheme must simultaneously obey two constraints: they must originate from *bottom-up* and also from *top-down*.

So just as that a fishing net cannot catch a fish that is smaller than the smallest hole in the net, one's conceptual system cannot absorb a new concept if it is not within the range of the existing concepts. In conclusion, Fodor argues that you cannot learn a new concept from *bottom-up* sensory information only. Empiricism alone cannot explain concept acquisition.

So how can a conceptual system come off the ground? How can the apparent circularity be broken? Fodor's conclusion, which has exerted an enormous influence, is that it cannot, and that primitive concepts must therefore be innate. We'll show later why Fodor's conclusion is false, but for now it is important to realize that learning cannot work by *bottom-up* induction only, and that there must be an active search involved driven from *top-down*.

Piaget and interactionism

One of the first people to realize this was Jean Piaget [49]. He described children's conceptual development as an active exploration of the environment. In his theory of mental development the interaction between the exploring child and bottom-up input from the environment plays a central role. Concept acquisition is an interactive process. According to Piaget, children first try to fit new information into their current conceptual schemes. This is called assimilation. If this doesn't work, the conceptual scheme has to be extended or repaired in order to fit in a new piece of information. This is called *accommodation* in Piagetian terms.

Piaget's views, also known as *interactionism*, are adopted within the framework of the TH experiment as part of the solution for the concept acquisition problem. This will be discussed later. Let's first have a look at some other questions.

The Gavagai problem, Quine's meaning holism

Imagine a cultural anthropologist, who visits a tribe in Africa to study their language. At a certain moment the native says 'Gavagai' while pointing to a rabbit running by. Now the question is how the anthropologist is supposed to know what the native meant: rabbit, or fur, or animal, or moving thing, or white? With this example Quine [52] pointed out the problem with learning the meaning of a word. The same problem exists for children. They, too, can only guess what their parents mean by a certain word, but they can never be sure, since they only receive feedback about successful or unsuccessful communication. *They never have direct access to the meaning of a word.*

A related problem in philosophy is known as *meaning holism*. It follows if you take inferential role semantics (the idea that the meaning of a concept is only defined with respect to the entire conceptual system of the individual) to its extreme: Assuming that each individual autonomously builds his own conceptual framework, which is a network in which all concepts are linked to other concepts (by inducing concepts from his private experiences), then it is not guaranteed at all that the concepts of one individual are the same as those of another individual. How do concepts become shared between individuals? Is the color that I conceive of as 'red' the same as that you conceive of as 'red'? Possibly there is a complete shift of concepts between two individuals, and nobody is ever going to find out. Since an individual cannot explain a single concept by just pointing to a referent, unless he specifies its relations to all other concepts in his conceptual system, then how can two individuals ever understand each other's meanings? How can concepts be universal? There is no way to exchange individual concepts. Fodor's reply would be that therefore we are forced to accept that all primitive concepts are innate, but this, as said before, is a ridiculous assumption. (Quine's ideas were represented here in a very informal way. Originally, Quine published his conclusions as a mathematical result in formal logic [52])

Meaning holism points at a very deep problem for theories of formal logic: it is not possible to assign meaning to a concept (or sense) in an objective way. This contradicts Frege's central idea that a *sense* is an objective thought with a fixed, uniquely determined referent. It also undermines the ambition of formal logic to deal with meaning in general.

The problem of meaning holism is specific for formal logic. It does not arise if you don't subscribe to the presuppositions of formal logic, more about which later. The TH experiment, on the other hand, offers an explanation of how meanings /concepts can become coordinated and shared between individuals, and of the role that language plays in shaping the concepts of different individuals in a similar way. And this takes us to the next point.

A.2 The philosophical assumptions underlying the Talking Heads experiment

The Sapir-Whorf hypothesis

The Sapir-Whorf hypothesis states that language shapes the way we think about the world and conceptualize it. Numerous examples illustrate the point: A study examining the language of the Pirahã tribe of Brazil, which contains only three counting words, one, two and many, reveals that the people of the Pirahã tribe have difficulty recounting numbers higher than three. There are also studies on color conception in communities which use a language with only a few colors, and studies on the use of prepositions and the implication this has for the conception of space. For example, in Mixtec, a language of the Otomonguean family, there is not a unique word corresponding to the English ‘on’. By contrast, Mixtec uses body-part projections, such as ‘he is standing head of hill’ when they mean on top of the hill, and ‘she sits arm tree’ when they mean on a branch of the tree. On the other hand, about Eskimo’s it is said that they can distinguish many kinds of snow, since their language has an unusually high number of words for snow.

But most interesting to us are the philosophical implications that follow from the hypothesis if you follow it right to its logical consequences. Citing Whorf [4]:

We dissect nature *along lines laid down by our native languages*. The categories and types that we isolate from the world of phenomena we do not find there because they stare every observer in the face; on the contrary, *the world is presented in a kaleidoscopic flux of impressions which has to be organized by our minds*-and this means largely by the linguistic systems in our minds. *We cut nature up, organize it into concepts*, and ascribe significances as we do, largely because we are parties to an agreement to organize it in this way - an agreement that holds throughout our speech community and is codified in the patterns of our language... all observers are not led by the same physical evidence to the same picture of the universe, unless their linguistic backgrounds are similar, or can in some way be *calibrated*.

Here is a partial solution to Quine’s meaning holism: language is essential for calibrating the conceptual frameworks of the individuals, and that explains why different language communities conceptualize the world differently. But this is only part of the answer. In order to arrive at shared concepts language must interact with the environment. The environment offers disambiguating stimuli which, coupled with language utterances, cause the conceptualizations to be disentangled. Altogether, what is needed to coordinate shared concepts is an interaction between *top-down* input (from

language) and *bottom-up* input (from the environment). These ideas are formalized in the TH experiment.

Concepts and properties are imposed upon the world

Another key idea that follows as a consequence of the Sapir-Whorf hypothesis is that we need to turn around the way we think about concept acquisition: the world doesn't offer itself to us partitioned into objects that we need to recognize and assign a concept to; on the contrary, we cut nature up, and organize it into concepts.

We partition the world, which is a continuous stream of analogue impressions, into discrete objects, so concepts are *imposed* on the world from *top-down*. The objects are thus mental constructions, or creations; they don't pre-exist in the world, only in the mind. It is very important to realize that concepts and properties are *imposed* upon reality as opposed to *discovered* in reality: the bulk of the literature on concept acquisition misses this point, and therefore has no satisfactory explanation for concept acquisition.

The concept of color

To show that properties are not inherent in the external world, we take the example of color concepts (freely adapted from [?]). If color were a primary quality, a property which exists in the external world, then our mental representation of color should have been a function of only the physical properties that determine color, such as surface reflectance and wavelength. But those physical properties alone don't explain many properties of color which we experience. For example, the fact that we experience a certain shade (hue) of red as the 'reddest' red, has nothing to do with surface reflection, but it is explained by the fact that such hues correspond to frequencies of maximal neural response. Color constancy, the fact that we perceive a banana as yellow independent of the lighting conditions cannot be explained from the external, physical properties of color. Rather, it has to do with the fact that our neural circuitry enables us to compensate for variations in illumination. Also the opposition between red and green is a fact about our neural circuitry, and the chemical makeup of the color cones, not about reflectance properties of surfaces. What we experience as green, is created from an interaction of reflected light with the color cones in our retinas and the neural circuitry connected to these cones. Plant life has been important in our evolution and so the ability to place in one category all the things that are green is valuable for survival. As a result, our brains have, given the world, evolved to *create* color.

Lakoff and Johnson [?] argue that all concepts are *embodied*, which means that concepts are defined only as an interaction between the external world and our body

and brain. So there are no inherent properties in the external world. ‘Cognitive science and neuroscience suggest that the world as we know it contains no primary qualities, because the qualities of things as we can experience and comprehend them depend crucially on our neural makeup, our bodily interaction with them and our purposes and interests’.

Consequences of mental view on concepts and properties for formal logic

Giving up on primary qualities would entail abandoning one of the pillars of formal logic, the correspondence theory of meaning. This is the idea that meaning is a relation between words or symbols and the objectively real world external to any perceiver. The correspondence theory of truth relates the (symbols of the) formal theories to the real world. Without assuming its correctness, formal logic cannot tell us much about the real world. We shall come back to this later.

As a consequence of the view that we adopt on concepts, we have already ‘solved’ two of the questions from the introduction: how do we know the objects in the world, and how do we know their properties? The reason we ‘know’ objects and properties is because we create them ourselves. They exist only in our mental model.

Why conceptualize?

Now the question arises, since there aren’t any objects out there in the world, why should we bother to divide the world up into objects? Why do we actually conceptualize the world? The answer is that discreteness is required for representing the world within an internal model. An efficient representation of the world cannot be achieved by storing analogue signals. Concepts are thus created for the sake of efficiently representing the external world in the internal model. It is important to be able to distinguish a tiger from a snake, a food source from a stone. If, as in lower organisms all world knowledge is pre-wired and instinctive, chances for and survival are much lower. For higher organisms, concepts and consciousness were invented by evolution so that the organism can act on the world consciously, plan and anticipate, which is adaptively favorable to reflexive behavior. [13].

It should also be noted that the formation of concepts is task-dependent / task-driven. Concepts are there for a goal: adaptation and survival. In the TH experiment the task that drives concept formation is a discrimination game. Being able to visually discriminate between objects is essential for the survival of the organism.

Concepts as distinguishing features

The most efficient/compact way to represent and store concepts would be to store only the distinguishing features of the concept. Those are the features that enable you

to uniquely identify the object from among other objects. There is no evolutionary advantage to be gained from storing any information that is redundant for the task of distinguishing between objects.

There is neurobiological evidence that this is indeed how concepts are stored. For example, when rats are reared in an environment where they are exposed to a range of tones of varying loudness, and if at the same time perceiving the correct loudness determines their chances for survival, then a disproportional area of their brain is dedicated to discriminate tones of different loudness, the size of which is proportional to the range of different tones [46].

It should be noted that such a definition of concepts leads to a so-called non-monotonic definition of meaning. As new facts come in, the meaning, consisting of distinguishing features must be repaired. This implies that meanings are open-ended and not pre-determined. Consistent with these ideas, in the TH experiment the *meaning* of a concept is represented by the (minimal set of) features that distinguish the object of interest from the context.

What's wrong with the classical definition of concepts

In contrast to the present definition of a concept by distinguishing features, the classical definition in the concept literature holds that concepts are defined by a set of necessary and sufficient features. There is however a problem with the classical definition of concepts: it is in most cases impossible to pinpoint an essential property, such that if the property were missing it would not belong to the category. For example, a dog with 3 legs, and without ears and a tail is still a dog. The deeper reason behind the problem is that the classical theory of concepts typically takes the view that objects pre-exist in the world with certain fixed properties, and that we have to recognize the object from the properties: if that is the case you would certainly expect the object to have some essential properties from which you can recognize it.

Presuppositions of formal logic

This would only be a minor problem, were it not the case that the classical definition of concepts is exactly how meaning is defined in formal logic. Here meaning is assigned to the symbols of a formal theory by postulating that the world has the structure of a fixed set of elements (a so-called set-theoretic model). There is a one-to-one correspondence between every symbol (or word) in the theory and an element in the set, which is fixed beforehand. A property is defined (extensionally) as the subset of the elements that have that property, and the meaning of a concept is then defined as the intersection of the subsets of elements having the necessary properties. Thereby formal logic de facto postulates (presupposes) the primacy of objects in the world. This is the already

familiar correspondence theory of meaning. Without such a meaning relation between the symbols of the theory and the world, the symbols would be meaningless, and the theory would not bear any relation whatsoever to the real world. But it is the claim of formal logic that it does say something about the world.

The failure of formal linguistics to capture properties of natural language

However, for all the reasons mentioned above, Steels rejects the correspondence theory of meaning and the classical definition of concepts. Assuming objective concepts and pre-existing objects, formal systems cannot capture interactionism, the interactive way meaning is established by carving the world into objects, and therefore they cannot adequately describe reality. Neither can they describe natural languages, which reflect this kind of interactionism. That is the price the (formal logic) Fregean enterprise has to pay for its choice of objectivism.

For the same reasons, attempts to describe natural language by formal languages are doomed to fail. This includes Montague grammar, formal semantics, and in particular Chomsky's generative grammar, which relies heavily on principles from formal logic. For example, it assumes a one-to-one correspondence (isomorphism) between syntax and semantics, it assumes that syntax is an autonomous formal system (so no interaction), and it assumes the classical definition of meaning. As a consequence, the tools employed by generative grammaticians are not adequate to study natural language.

The TH experiment should be seen in this light: It is a description of natural language from an interactionist point of view. The TH experiment challenges formal theories of language, and in particular generative grammar, by introducing a non-formal, dynamic-systems approach to language and language acquisition. The theory of language envisaged in the TH experiment is compatible with a different approach to linguistics, known as construction grammar [?], or usage based grammar [?], the details of which are out of the scope of the present paper.

Natural language according to Steels

The TH experiment sketches a very different picture of natural language than the orthodox picture. Conventionally, in generative grammar, language is seen as a fixed/static system of rules (grammar) and words (lexicon). There is only one perfect and ideal language, which all competent speakers speak, so the language community is assumed to be homogeneous. All words have a fixed meaning, the grammar is static (whether it is innate or not). Learning a language amounts to learning the ideal, objective language, in a passive way. Theorems about language learning, for example Gold's theorem, treat language as such a formal system.

In the view outlined here, language is a dynamic and adaptive system, like an ecology. It lives as a kind of parasite (meme) on a language community, independent of any single individual speaker. No single individual language user is in control of (nor has an overview of) the ideal language. In contrast, if you want to study natural language, you must consider the entire population, because language is a phenomenon of a population, not of an individual. Language is in constant flux and evolution, word forms and meanings constantly change. Learning is seen as participating in the dynamic system.

The idea that there is not one ideal, objective language extends to thought and concepts. Concepts, too, interact within the dynamical system of a population. Objective thoughts, independent of individuals, don't exist, for the same reason that objective language doesn't exist. The Fregian *sense*, as an expression of objective thought, might therefore not be the appropriate starting point for modelling human thought.

So how does interactionism solve the learning paradox?

The learning paradox as formulated by Fodor presupposes, like any formal theory, that objects and their properties are objective, and fixed in the external world, and that learning a concept means that you have to find it in the external, real world. However, if one assumes that concepts and properties are mental constructions imposed on reality, that they are inventions, then they do not have to be 'learned' in the sense that they do not have to be extracted or recognized from the data. This radically changes the perspective on concept acquisition and on the empiricist/nativist dispute. It now becomes obvious that concepts cannot be acquired by induction alone, and that they cannot be innate either, because they need the raw data to impose structure on.

A second presupposition in Fodor's argument for the impossibility of learning concepts is that concepts are fixed and static, and this turns out to be false. According to the view sketched above, concepts are no longer absolute, fixed entities, but rather task dependent and dynamically changing. Meaning is non-monotonic.

There is a lot of evidence from developmental psychology, that children's concepts change and become more adult-like as they grow. A case in point is over-extensions. For example, very young children would call every man 'daddy', and every round thing 'ball'. As the child collects more and more disambiguating instances, coupled with a word, their concepts gradually converge to adult-like concepts. And also the *distinguishing features* that children have understood to define the concept, change. My brother, for example, called very many different types of vehicles 'daddy car', until it became clear, that he used a small bolt on the wheel to identify the vehicles as belonging to the same category.

Co-evolution of language and concepts

These are all the assumptions that we need to solve the learning paradox, which, once again, states that you need to represent new concepts in terms of existing concepts: new concepts are acquired through a dynamic and incremental process of fine-tuning. The child starts his life with very coarse and inadequate concepts, and during his development refines and differentiates those, until they converge to adult-like concepts, under the influence of language. This obviates the need for innate concepts.

An important consequence of this acquisition process is that language and concepts co-evolve. They are intertwined. This is true for the first language, but not for languages learned at a later age. That would explain the fact that second languages are usually not learned as well as first languages. In terms of the fishing net metaphor: the holes in the net are made finer and finer, until the net catches the same kinds of fish as the adult fishermen.

Selectionism

But we are not there yet: we still need a mechanism by which the concept acquisition apparatus can get off the ground. Before, we noted that neither nativism nor empiricism offers an adequate theory of concept acquisition. If we propose to impose a *top-down* organization into categories on the world, how do we get those categories in the first place if they are not innate? And how do we achieve the desired interplay between *top-down* and *bottom-up* input? The idea is to use the principles of selectionism. Those entail:

1. an internal growth process, which generates a variety of categories in a random fashion, even in the absence of examples.
2. a process for preserving categories so that there can be a gradual build up of more complex categories
3. a selectionist force, which uses feedback from the environment, and exerts pressure on which categories to preserve and which categories to prune.

Essentially, selectionism describes the interaction between genetic factors and environment: it adopts the principles of interactionism, proposed by Piaget to explain growth of mental capacity. The selectionist view is thus neither a nativist (*top-down*) nor an empiricist (*bottom-up*) account of concept acquisition. Rather, it explains concept acquisition as a dynamic interaction between *bottom-up* and *top-down* processes. In contrast to induction, selectionism is a rapid process, and it can explain the instantaneous learning of categories seen with children, which induction cannot.

Examples of selectionism in action are abundant in biological, and, in particular, ecological systems. Of most interest for us is the way genetic factors and environment, interact during brain development of higher organisms. Since there is no blue print for the brain of higher organisms, stimuli from the environment play a critical role in guiding brain development: there is an initial overproduction of synaptic connections to a single target neuron, called polyneuronal innervation [51]: many spurious connections between various brain areas are tried out. This parallels the random generation of variety. But in order to survive, those innervating synapses need trophic feedback from target neurons located nearer to the sensory periphery. They will only get this through electrical stimulation from the environment. In the process of elimination of inputs synapses originating from different neurons compete with each other for ownership of an individual target cell [51]. Synapses that don't receive trophic feedback atrophy, and the corresponding nerve cells will eventually die. Thus, by selectionist pressure, the environment supplies feedback which controls which connections are sustained.

Many (cruel) animal experiments demonstrate the interplay between the environment and neuronal development of the cortex: It has been shown for instance, that visual deprivation in one eye of a kitten, being blindfolded during the early months of development, causes blindness in the deprived eye, due to the fact that the good eye takes control of most of the neurons in the ocular dominance columns belonging to the deprived eye [32]. Thus, stimuli from the environment are necessary for normal brain development. Remarkably, with the loss of cortical function, those animals lose the ability to represent certain concepts (or categories). We leave the implementation of the selectionist principles in the TH experiment for later.

Appendix B

Learnability and the Gold Theorem

A theorem about learnability of formal languages by Gold [25] seems to support the innateness claim. The Gold theorem states that even the set of regular grammars is not identifiable in the limit by means of positive examples alone. In the proof, learning a language is considered to be tantamount to identifying the correct grammar from a set of grammars. The target grammar is identified by enumerating one by one the grammars from a hypothesis space, which is assumed to be innate, until all the examples are consistent with the hypothesized grammar. In short, the theorem states that if the set contains at least one grammar that generates an infinite language, there would be no means for the learner to find out whether the language is generated by a finite language generating grammar or by an infinite language generating grammar that subsumes the finite language of the target grammar.

Some comments are in place here concerning the assumptions made by Gold's proof about the nature of language and language acquisition: First, it is assumed that the input for the learner consists only of sentences without additional contextual cues from the real world, such as vision, and without a hypothesis about the sentence meaning based on the learner's existing conceptual framework. Second, it is assumed that a grammar has to be learned to perfection. This implies a view according to which there exists only a single correct grammar without which linguistic communication would not be possible. Third, it is assumed that every adult individual possesses exactly the same grammar. In the alternative view on language, introduced in section 2.4.3, there is not a single perfect adult grammar that all language learners must acquire. It is thought that a grammar is dynamically constructed by every individual through communicative interaction with the other members of a language community, while no single individual possesses a complete overview of the grammar. Thus, children's grammars are not considered inferior to adult grammars, but they contribute equally to the 'group grammar', a kind of average grammar of the community. In the learning

process the individual adapts her grammar to the group grammar.

Appendix C

Developmental stages in UBG

In Tomasello’s Usage Based Grammar [66] four main developmental stages in the child’s use of constructions are distinguished, which become more and more complex over time [68]. See figure C.1 for an characterization of the four stages.

	Lexical partitioning of scenes	Syntactic marking of participant roles	Categorization of specific scenes
Holophrases	-	-	-
Word Combinations	+	-	-
Verb Island Constructions	+	+	-
Adult-like Constructions	+	+	+

Figure C.1: Characterization of phases in early syntactic development

1. In the ‘holophrasic’ stage, at around 12 months, children attempt to reproduce entire utterances by means of imitation. A holophrase is a single linguistic symbol functioning as a whole utterance, for example single adult words such as *Ball?* meaning *Where’s the ball?* or *Give me the ball*, or multi-word patterns, such as *Lemme-see*, or *I-wanna-do-it*. Children do not distinguish single words from more complex constructions.

Children attempt to reproduce by imitation the whole utterance, as opposed to generating the utterance from single words by means of production rules. They pick out the most salient aspects of the sentence, aided by cues such as intonation and stress. Evidence for imitation comes from child speech, for instance *Her open it* is based on hearing the adult say *Let her open it*. In contrast, children would

never make parallel errors like *Mary hit I* because they never hear it as part of an adult utterance.

2. After the holophrastic stage, around the age of 18 months, children learn to use combinations of utterances: they use different words to indicate different semantic components of the scene, thus partitioning the scene into at least two components. Usually the word combinations center around one constant element (e.g. *more milk, more grapes*) - this is called the 'pivot-look'. Although word order is mostly used consistently, there is not yet any indication that it is used to do syntactic work (e.g. *gone juice* means the same as *juice gone*).

3. In the next stage (around 24 months) children learn constructions with open positions (slots) where variable words can be integrated. Examples are *Where's the X?*, *I wanna X*, *More X*, etc. These utterances are built up around item-based constructions, which are learned and used independently of each other.

In order to do so, the child must be able to 'break down' or 'fill out' her holophrases. For example, breaking down *Lemme-see* yields *Let me see*, and filling out *Ball?* yields *Where's the ball?*. Adult words are first learned by comparing constructions on the variable position and extracting the variable constituents, e.g. comparing *where is the ball* with *where is the car* leads to ball and car to be identified as autonomous constituents.

The same process leads to the beginning of abstraction and category formation. Type variation across a position in an item-based expression causes abstraction over the position (slot). Constructions with a single variable slot, such as *where's the X?* can be regarded as the first kind of rules. Slots mark the beginning of grammar. The grammar consists of a growing collection of item-based constructions with (multiple) slots. The entries of the lexicon are whole constructions, not just primitive words as in generative grammar. As remarked earlier, there is no distinction between the syntax and the lexicon (both are integrated in constructions with slots), and therefore no modular design has to be postulated.

4. In the final stage (3 years and older) children acquire an adult-like grammar, with adult-like semantic and syntactic categories. In a process called grammaticalization they abstract across verb-islands to form adult-like categories. For example, combining schemas for *the cutter, cut* and *something-cut* with *the breaker, breaks* and *something broken*, they arrive at: subject - verb - object. Adult categories and rules are thus induced by abstracting over input; they are not innate. Abstractions are made by analogy: two constructions are analogous if they can be mapped onto each other both by form and function. This way categories obtain functional role

Tomasello suggests two ways by which overgeneralization is restricted [69]. The first is entrenchment: A high token frequency of a particular expression causes it to become entrenched, and causes the associated rule to become stable. The second is preemption [28] This means that if the child is exposed to an alternative way of expressing a meaning she infers that the standard way is not conventionally used. For example, hearing the expression *the magician made the rabbit disappear* may prevent the child from using *the magician disappeared the rabbit*. Both strategies have been confirmed by empirical studies using nonce verbs [69].

Appendix D

The Poisson distribution in e-Grids

The Poisson distribution replaces the STOP bit in the bodies of the rules of $SB1$, while the STOP bit of the rules of $SB3$ is completely removed. Therefore, every symbol in the bodies of $SB1$ and $SB3$ requires $\log(A_{UNT})$ rather than $\log(A_{UNT} + 1)$ bits to encode. With the Poisson adaptation, the GDL from equation 3.19 becomes:

$$\begin{aligned} GDL &= (\sum_{SB3}(|LHS|) + T + 2) \cdot \log(A_{UNT} + 1) + \\ &+ (\sum_{SB1}(|NT_R|) + \sum_{R \in SB3}(|NT_R|) \cdot \log(A_{UNT})) + \\ &+ \sum_{SB1} Poisson(|NT_R|) + T \cdot \log(T) \end{aligned} \quad (D.1)$$

where $|LHS|$ represents the single non-terminal in the LHS of the rule and $Poisson(|NT_R|)$ is the Poisson distribution

$$Poisson(k) = {}^2 \log \left(\frac{e^{-\mu} \cdot \mu^{k-1}}{(k-1)!} \right) \quad (D.2)$$

as a function of the number of non-terminals in the body of the rule, k .

As for the DL Gain of a chunk, the first term of equation E.2 is modified. The RHS of rules of $SB1$ doesn't require a STOP symbol anymore, so each symbol can be encoded by $\log \left(\frac{A_{UNT}+1}{A_{UNT}} \right)$ bits (the extra bit is for the additional chunk non-terminal). That leaves us with $(A_{NT} - A_{RHS \text{ of } SB3} - A_{RHS \text{ of } SB1} + 2) = (\sum_{SB3} |LHS| + T + 2)$ symbols that are encoded by $\log \left(\frac{A_{UNT}+2}{A_{UNT}+1} \right)$ bits. The second term of equation E.2 remains unchanged for Poisson. Altogether we have

$$\begin{aligned} \Delta ML_{CHUNK(X,Y), Part 1} &= (\sum_{SB3} + T + 2) \cdot \log \left(\frac{A_{UNT}+2}{A_{UNT}+1} \right) + \\ &+ (A_{RHS \text{ of } SB1} + A_{RHS \text{ of } SB3}) \cdot \log \left(\frac{A_{UNT}+1}{A_{UNT}} \right) + \\ &+ \log(A_{UNT} + 2) + (2 - BF(X, Y)) \cdot \log(A_{UNT} + 1) \end{aligned} \quad (D.3)$$

In addition, for all the rules for which as a result of the chunk the length is changed, we must compute the difference in Poisson between the old and new length of the rule, and we must add the Poisson term for the new chunk rule. These give a contribution to the DL Gain of

$$\begin{aligned} \Delta ML_{CHUNK(X,Y), Part 2} &= \sum_{\Theta} (Poisson(k_{After}) - \\ &\quad - Poisson(k_{Before})) + Poisson(2) \end{aligned} \quad (D.4)$$

where Θ is the set of rules whose lengths have changed as a result of the chunk.

Equivalently, the first term of the DL Gain for a merge is modified as follows:

$$\begin{aligned} \Delta ML_{MRG(X,Y), Part 1} &= (\sum_{SB3} + T + 2) \cdot \log\left(\frac{A_{UNT}}{A_{UNT}+1}\right) + \\ &\quad + (A_{RHS \text{ of } SB1} + A_{RHS \text{ of } SB3}) \cdot \log\left(\frac{A_{UNT}-1}{A_{UNT}}\right) \end{aligned} \quad (D.5)$$

The second term becomes:

$$\begin{aligned} (\sum_{j \in \Omega_1} (|LHS|)) &+ \sum_{j \in \Omega_3} (|LHS|) \cdot \log(A_{UNT}) + \\ &+ (\sum_{j \in \Omega_1} (L_j) + \sum_{j \in \Omega_3} (L_j)) \cdot \log(A_{UNT} - 1) \end{aligned} \quad (D.6)$$

where Ω_1 and Ω_3 respectively are the sets of rules from $SB1$ and $SB3$ that are eliminated as a result of the merge.

Additionally, for all removed rules, we must subtract the Poisson contribution:

$$\Delta ML_{MRG(X,Y), Part 2} = - \sum_{j \in \Omega_1} (Poisson(|NT_j|)) - \sum_{j \in \Omega_3} (Poisson(|NT_j|)) \quad (D.7)$$

Appendix E

Removal of STOP bit from SB3 in e-Grids

Since all the rules of *SB3* have length 2, a STOP bit is not required for rules in *SB3*. This slightly complicates the e-Grids equations: The fifth term in equation 3.18 for the GDL must be replaced by

$$\sum_{R \in SB3} (1 \cdot \log(A_{UNT} + 1)) + \sum_{R \in SB3} (|NT_R| \cdot \log(A_{UNT})) \quad (\text{E.1})$$

The first term reflects the LHS, where each symbol needs $\log(A_{UNT} + 1)$ bits to encode: the extra 1 is for the *TOP* non-terminal. The second term reflects the RHS of the rules of *SB3*, where non STOP symbol is necessary.

Equation 3.25 must be changed correspondingly. A_R , which represents the number of STOP bits in the rules of the grammar, becomes identical to A_S , so they cancel out. A_{NT} , the number of occurrences of all non-terminals in the grammar, must be split in two parts: the first part, which represents all non-terminals in *SB1*, *SB2* and in the LHS of *SB3* is still multiplied by $\log\left(\frac{A_{UNT}+2}{A_{UNT}+1}\right)$; The non-terminals occurring in the RHS of *SB3* must however be multiplied by $\log\left(\frac{A_{UNT}+1}{A_{UNT}}\right)$. The second term, which represents the contribution of the non-terminals in the additional rule for the chunk (originally 4 non-terminals, but now only 3) minus the contribution of the chunked bigrams (*BF*), must also be split: the contribution of a non-terminal in the LHS of the chunk rule is $\log(A_{UNT} + 2)$ as before (1 for the *START* symbol and 1 for the additional non-terminal), but the other symbols occur on the RHS and are therefore

encoded by $\log(A_{UNT} + 1)$ bits. Altogether we have:

$$\begin{aligned} \Delta ML_{CHUNK(X,Y)} &= (A_{NT} - A_{RHS \text{ of } SB3} + 2) \cdot \log\left(\frac{A_{UNT+2}}{A_{UNT+1}}\right) + \\ &+ (A_{RHS \text{ of } SB3}) \cdot \log\left(\frac{A_{UNT+1}}{A_{UNT}}\right) + \\ &+ (1 - BF(X, Y)) \cdot \log(A_{UNT} + 2) + 2 \cdot \log(A_{UNT} + 1) \end{aligned} \quad (\text{E.2})$$

(this assumes that bigrams don't occur in the RHS of SB3: those have frequency 1)

Similarly, equation xxx for the contribution of a merge to the change in GDL must be changed. The first term becomes, analogously to the first term of E.2,

$$\begin{aligned} \Delta ML_{MRG(X,Y), \text{ first term}} &= (A_{NT} - A_{RHS \text{ of } SB3} + 2) \cdot \log\left(\frac{A_{UNT}}{A_{UNT+1}}\right) + \\ &+ (A_{RHS \text{ of } SB3}) \cdot \log\left(\frac{A_{UNT-1}}{A_{UNT}}\right) \end{aligned} \quad (\text{E.3})$$

In the second term, $\sum_{j \in \Omega_3} (L_j + 2) \cdot \log(A_{UNT})$ must be replaced by:

$$\sum_{j \in \Omega_3} \log(A_{UNT}) + \sum_{j \in \Omega_3} (L_j) \cdot \log(A_{UNT} - 1) \quad (\text{E.4})$$