*i*DOP:
Iterated Data-Oriented Parsing
as a Model for Diachronic Syntax

MSc Thesis (*Afstudeerscriptie*)

written by

Maarten Versteegh
(born November 15th, 1977 in Amstelveen, the Netherlands)

under the supervision of **Dr. Willem Zuidema**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
August 28, 2009	Dr. Peter van Emde Boas
	Prof. dr. Remko Scha
	Prof. dr. Frank Veltman
	Dr. Willem Zuidema

Institute for Logic, Language and Computation

Contents

List of Figures

Chapter 1

Introduction

Language change is a complex process, involving many factors, such as acquisition, language contact and innovations made by language users. These factors span the entire range of aspects of human language, from pragmatics to phonetics. Because of this complexity, historical linguistics is often practiced as an informal science, relying on qualitative descriptions of processes and theories. This thesis reports on the formulation of a general quantitative model for the study of syntactic change. The overarching goal of this thesis is to contribute to the development of formal models and computational tools for the study of language change.

There are two main motivations for developing computational models of language change. The first is that such models may inform our theories on language change. Simulating change on a computer allows us to take a closer look at the mechanisms at work in language change. The results may then be used to support and develop theories of diachronic linguistics. The second motivation is to use these models to test the validity of formal models of language and language acquisition. If a language formalism is to be regarded as an accurate model of human language, we should be able to use it in models of language change. Simulations can be used to study the behaviour of language formalisms under the conditions of repeated learning and use in a language community. This behaviour may inform our evaluation of the validity of a language formalism as a viable model of human language. In this thesis, we will mostly be concerned with the first of the mentioned motivations for the computational study of language change; to inform our understanding of linguistic theories of syntactic change.

There are relatively few computational studies of language. Notable studies are the agent-based computational models of the evolution of language through iterated learning [10, 33], of the social factors involved in change [43], and of the formation of the relation between language learning and language contact [49]. None of these studies employ a fully specified language model together with a linguistically motivated model of language transmission to study language change.

This thesis presents a quantitative model of language change that integrates a formal model of language, Data-Oriented Parsing, in an agent-based iterated learning simulation. A central role in the model is reserved for language acquisition, taken in the broadest sense. Both the transmission of language from one generation to the next as well as between agents in a single generation are accounted for. Analogy as a mechanism of syntactic change will be central to the transmission.

Our research has two goals. First, to develop and implement a general quantative model of syntactic change that can be applied to a wide variety of phenomena. The model has to be specific enough so that we can study the language output of each generation in detail. The second goal is to test the assumptions of the model by examining case studies from historical linguistics. We strive to build a model such that the case studies can yield linguistically plausible results for language change. The simulations will allow us to examine the mechanisms at work in the phenomena in these case studies in more detail than an informal discussion may accomplish. This aspect is an important motivation for the development and implementation of quantitative models for language change.

The thesis is outlined as follows. Chapter 2 will discuss linguistic theories of syntactic change. Insights from these theories will inform our model. The chapter introduces the general model of language transmission that underlies our research. From this model follow a number of demands that the language model must meet. Chapter 3 discusses the details of the language model, Data-Oriented Parsing. It also addresses technical details of the implementation. Chapter 4 contains discussions on the simulations and their results. In it, a full implementation of the model of transmission is gradually introduced. Case studies of historical linguistic phenomena are discussed to clarify and examine the model. Finally, in chapter 5, we discuss some general conclusions of the results presented in chapter 4 and some considerations for further research.

Chapter 2

Models of language transmission

This chapter discusses the linguistic background of the research in this thesis. The study of diachronic syntax and syntactic change is an active area in linguistics. Much like other areas in linguistics, there is no generally agreed on treatment of it. Instead several approaches exist. The approaches differ essentially in their treatment of language transmission. We first discuss two of the major approaches to syntactic change and the mechanisms these approaches identify as central to change. We focus our discussion on the different aspects of language transmission that the approaches highlight. Then we introduce the model of transmission used in this thesis, building on insights from both traditions.

2.1 Models of language change

2.1.1 The generative approach

Since the emergence of Generative Grammar in the 1950s, the generative approach to historical syntax has been actively researched. A central tenet of Generative Grammar has been the study of language acquisition and its implications for a wider theory of syntax. The research has focused on what Chomsky dubbed 'logical problem of language acquisition' [16]. Given that an adult speaker can give grammaticality judgements on an unbounded number of sentences of her

native language, the central problem is how language acquisition is possible *in principle*, given the fact that the primary linguistic data offered to a language learning child, crucially *underdetermines* its adult linguistic competence, being the understanding and knowledge of the language. The generative tradition assumes an innate, initial state of the language faculty, shared by the entire human species. From this initial state, the *Universal Grammar*, every human develops his or her native tongue.

This view on the cognitive abilities of humans has strongly influenced the generative approach to language change. This approach associates syntactic change with child language acquisition. It points to the discontinuity of grammars between generations as essential for change to occur. Syntactic change is an emergent feature, a result of what happens in the transmission of a grammar from one generation to the next. Put simply, the generative approach states that if the grammar of a generation has its parameters set differently from the previous generation, then a syntactic change has taken place. This idea is exemplified in the diagram in figure 2.1.1, based on [1].

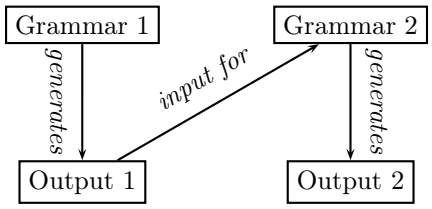


Figure 2.1: Diagram outlining the discontinuity of grammars in the basic model of language transmission from [1].

The diagram, which is the basis of many works in the generative study of syntactic change, amongst others the influential work of David Lightfoot, embodies a number of essential features to this approach to language change which we will discuss here in detail.

Firstly, the diagram points to the crucial interplay between child language acquisition and linguistic output. The grammar which a child acquires is not based directly on the grammar of its caretaker, but on the caretaker's linguistic output, acting as an intermediary.

In essence, there are two ways in which *Grammar 2* may come to differ from *Grammar 1*. First if *Output 1* is not representative of *Grammar 1* in such a way that the learning child cannot reconstruct *Grammar 1* from it. Second if, given

that *Output 1* is representative of *Grammar 1*, the child's learning mechanism interprets it the 'wrong way'. Since the child bases its grammar solely on the output, the parameters of the child's Universal Grammar may be set differently and so a language change will take place.

The second point of interest with this picture is that change is generally seen as abrupt. This view of abruptness is especially present in the works of David Lightfoot [39, 41]. The idea there is that over generations grammatical complexity builds up gradually through minor, relatively unimportant changes until a threshold is reached which triggers a major and far-reaching restructuring of the grammar. This restructuring is supposed to eliminate the complexity that made the language difficult for children to learn. Crucially, Lightfoot argued that syntactic change is autonomous, just as syntax in general is ascribed a high degree of autonomy in generative theory [39].

These views have received a fair amount of criticism (for an up to date overview see [19]). The first point of criticism we discuss is that the abrupt changes are actually quite rare and do not represent the majority of syntactic changes. The emphasis on the 'catastrophic' changes in historical data may even be said to mask a confusion about the locus of change and the origin of the data. While the discontinuity of grammars means that a change is indeed instantaneous from the perspective of an individual's competence, the picture as it stands fails to incorporate two important factors. First, the data in historical linguistics do typically not originate from single individuals. Historical texts do not, in general, allow us to closely inspect the development of single users of a language. Second, the data are *performance* data, that may or may not accurately reflect the individual competences of a group of language users. As such, changes that may be instantaneous from the point of view of individual users, may take hundreds of years from the point of view of a language community and may seem very gradual indeed. A focus on individual competence to the exclusion of communal performance, obscures this important fact in the study of historical syntax.

To counter this concern we may adapt the diagram from figure 2.1.1 to reflect the transmission of language in a community of speakers and learners, see figure 2.1.1. This diagram is not supposed to imply that children necessarily learn language from everyone in their community. Some subsets of *Collective output 1* may be vastly more important in the acquisition process than others.

One may object that the instantaneity of change is not necessarily inherent to the model of transmission in picture 2.1.1. A model of language acquisition

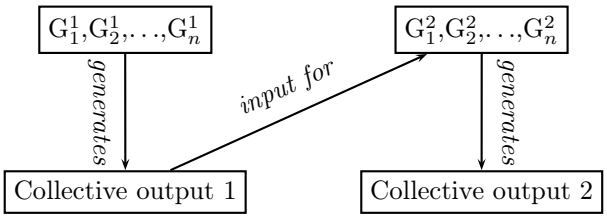


Figure 2.2: Revised model of transmission, showing collective outputs and sets of grammars.

which does not rely on parameter setting and is able to incorporate the frequencies of form occurences may produce a much more gradual picture of language change. This feature then crucially depends on the specific details of the learning algorithm. If one is willing to let go of the generative model of acquisition, one may obtain a more fine-grained model. However, this does mean that the generative definition of syntactic change, i.e. a difference in parameter settings from one generation to the next, becomes void. A new definition of change is then needed.

A second general shortcoming of the model so far is that it only explicitly incorporates *internal* factors in the change. Internal factors in language change are defined as factors promoting change *within* the linguistic system of a relatively homogeneous community, experiencing little outside influence. In contrast, external factors involve language contact, active language policy etc.

External factors, especially language contact, are important in diachronic syntax (as attested by the literature, see for example the collection in [24]), so we wish to account for them. We can adapt the model so that external factors (of any kind) can have influence on the linguistic output of the community, see figure 2.1.1.

External influence on the performance level may result in the output not being representative of the speakers' grammars. For example, suppose that *Grammars 1* are fully case marked. This is the grammar the adults acquired as child. Due to some influence external to the grammar, the case markings in the output start to disappear. Note that in the generative view *Grammars 1* are still fully case marked on the competence level. It is only on the *performance* level that the case markings have disappeared. A child presented with linguistic

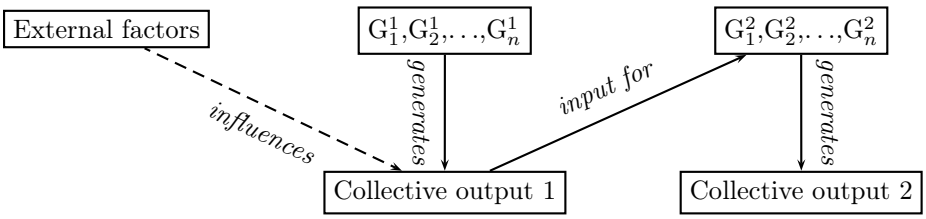


Figure 2.3: Second revision to the model: external influence on the speakers' output.

output from this speaker cannot but infer that the language it is learning does *not* support case marking. It sets its parameters accordingly and thus *Grammar 2* differs drastically from *Grammar 1* and a change has taken place.

We will come back to bith this model and the internal/external distinction later when we discuss other models of grammar.

Lastly, the model assumes, as does generative theory in general, a level of autonomy of (morpho-)syntax that has been questioned by researchers from other traditions (e.g. [25]). The mechanism of language acquisition is presumed to revolve around syntax. Therefore, phonological, semantic and especially pragmatic factors are difficult to incorporate into this model. The second important approach to language change, *grammaticalization*, discussed below, in part attempts to remedy this problem.

In summary, the generative approach to language change is based on the generative approach to language acquisition. This approach entails a discontinuity of grammars. A central role is ascribed to language learners within the critical period of acquisition, as their unique learning method allows them to set parameters and thus to change grammars, whereas adults, with their parameters set, learn new constructions by a completely different mechanism with only superficial consequences. The model can be extended by incorporating the communal nature of a child's primary linguistic data and external influences on these data.

2.1.2 Grammaticalization

The second approach to language change that will be relevant to our model, grammaticalization, is part of a long tradition, going back to the linguists of the nineteenth century, such as Wilhelm von Humboldt [42]. A widely used definition of grammaticalization comes from Jerzy Kuryłowicz:

> Grammaticalization consists in the increase of the range of a morpheme advancing from a lexical to a grammatical or from a less grammatical to a more grammatical status. [44]

Grammaticalization studies typically involve themselves with two types of phenomena. The first are the changes of lexical items to morphemes and the second are the changes of discourse elements to morphosyntactical markers. These changes involve two mechanisms that are extensively studied.

The first is *semantic bleaching*. This involves the loss of meaning of lexical items. An example is the loss of meaning of the emphatic negation in Jespersen' cycle [30]. Consider the development of negation in French, that can be sketched as follows:[1]

1. *Je ne sais* (NEG + VERB)
2. *Je ne sais (pas)* (NEG + VERB (+ EMPHATIC NEG))
3. *Je ne sais pas* (NEG + VERB + OBLIGATORY NEG)
4. *Je (ne) sais pas* ((NEG) + VERB + NEG)
5. *Je sais pas* (VERB + NEG)

In the second stage, the item *pas* is optional, indicating emphatic negation and still retaining some of its original meaning. In the third stage, this meaning is lost and the item has become an obligatory part of the negation.

The second mechanism involved in grammaticalization is *phonological reduction*. In this thesis we are only interested in the form that is called *syntagmatic* reduction: 'forms become shorter as the phonemes that comprise them erode' [28, p.154]. It entails the dropping of vowels and consonants, loss of stress and/or assimilation of adjacent phonological segments. This process is important in the development of clitics into affixes.

[1] Modified examples of stages from [28, p.65-66].

When evaluating grammaticalization phenomena it is important to again clearly distinguish the point of view of the language and the point of view of the language user. The object of the study is the process of language as a dynamic interchange of meaningful communication. Taken this way, grammaticalization takes a *functional* approach to the study of language change, incorporating the *use* of language between users into their model. Some grammaticalizationists are inclined to ascribe to the phenomena they study an existence per se, and look at grammaticalization as an *agent* of change (cf. the discussion in [19], pages 115–124). We do not wish to go this far, but feel it is important to note this contrast to the generative approach, which completely focuses on the individual speaker's competence, leaving out the role of language as a communication device used by actual human beings.

Frequency

The focus on the performance level, where 'the variation and fuzziness is to be found which forms the beginning of change' [19], brings another aspect of language in sight of grammaticalization researchers: frequency. As Hopper and Traugott put it: frequency has 'assumed an important place in the empirical study of how lexical forms move into grammatical roles', i.e. in grammaticalization studies. [28].

Joan Bybee [11], working from a usage-based background, points out two important effects of the frequency of constructions. Firstly, high frequency constructions are more likely to undergo grammaticalization processes and secondly, high-frequency constructions retain their form longer under pressure from new formations. These two effects are strongly related to what Bybee and Thompson [12] term the *Reduction Effect* and the *Conservation Effect*.

The Reduction Effect states that high-frequency forms undergo erosion (both semantic and phonological) at a faster rate than less-frequent forms. This effect is attested by for example the phonetic shortening that English contractions such as *you're*, *I'll* undergo. The Conservation Effect states that highly frequent items are more likely to retain irregular forms and are less likely to be levelled by the pressure of regular forms. For clear evidence of this effect, Lieberman et al. studied the regularization of English verbs over the past 1200 years and showed that the rate of regularization of a verb depends on the frequency of its usage [38].

Basing themselves on these observations, Hopper and Traugott state that

the increased frequency of a construction over time is 'prima facie evidence' of grammaticalization [28, p.129]. This may not be a formally accurate definition, after all the word 'computer' probably increased significantly if frequency of use between, say, the 18th century and the present for reasons other than grammaticalization. Nonetheless, we take frequency to be an important factor in our model of change and will partly base our definition of change on it (see chapter 4).

Analogy & reanalysis

Two mechanisms that are inextricably connected to grammaticalization are analogy and reanalysis. Reanalysis has been called 'the most important mechanism for grammaticalization [28, p. 39], and one of the three only mechanisms for syntactic change [13, p. 283]. Analogy has been referred to as the mechanism which actually implements a change and spreads it across a language.

Reanalysis is best defined by a well-known quote from Harris and Campbell.

> Reanalysis ... is a mechanism which changes the underlying structure of a syntactic pattern and which does not involve any immediate or intrinsic modification of its surface manifestations. [27, p. 61]

A simple example of reanalysis is the modern use of the word *Hamburger*. Originally, it was constructed as *[Hamburg] + [er]*, meaning a foodstuff originating from Hamburg. Over time, the original meaning was lost and the word was reanalysed as *[Ham] + [burger]*. In itself, this reanalysis does not change the surface appearance of the word. But combined with analogy it opens up the possibility for innovations, such as *Cheeseburger*. Without the initial reanalysis, analogy would not be able to implement the innovation.

Analogy has been defined as the 'inference that if two thing agree in certain respects then they probably agree in others' [22, p. 17]. Analogy has two mechanisms to spread innovations, extension and levelling. Analogical levelling refers to the process by which allomorphs align themselves with a single instance. Often it refers to the pressure that highly frequent allomorphs exert on less frequent ones. Campbell [13] defines levelling as the disappearance of certain allomorphs in favour of others. As an example consider the the previously mentioned regularization of infrequently occurring verbs. Analogical levelling forces irregular verbs to conform to the regular paradigm, thereby eliminating allomorphs. This definition takes an *a posteriori* look at levelling, as it is only

able to identify a levelling having taken place if one or more allomorphs have completely vanished. In chapter 4 we will give a new definition of levelling in a probabilistic framework, that will alow us to look at levelling as a process.

Analogical extension refers to the application of rules to new forms, by analogy to existing forms. Examples are the regular declension implemented on new verbal forms such as *e-mail* or *twitter* or the formation of the plural for new nouns. This form of analogy is often considered relatively rare (cf. e.g. [13, p.284], as it is restricted to new forms. In the next chapter, however, we discuss a parsing model that works in part by the analogical extension of previously experienced syntactic structures to new sentences. If this model has psychological relevance, then extension is in fact the predominant form of analogy.

Consequences for the transmission model

The mechanisms identified by grammaticalization research as described above, have several consequences for our model of language transmission.

The first is the following. If we take analogy serious as a mechanism for change then we have to rethink the role of adult language users in our model. In the above discussion of the generative model we saw that there only the acquisition of language by children plays an important role in language change. However, language researchers, and sociolinguists in particular, are increasingly pointing out studies that show that adults continue to develop their language. This includes adapting their own language output to that of their community, but also innovating new forms (cf. e.g. [28, 36, 37]). This has consequences for the view that children are the most important 'learners' of a language.

While child language acquisition is undoubtedly special in that it differs qualitatively from adult language acquisition, the insights from sociolinguistics suggest it is wrong to dismiss the results on adult innovation and language development in a model of language transmission. The fact that adults can invent and spread forms through a language means that the basic model of transmission needs to be revised again. See figure 2.4 for a revised diagram of the model. This revision is represented by letting the linguistic output feed back into the speakers' model of their language, keeping in mind that we now also need to rethink the way the language model works in order to account for innovations on the part of adults. This means that there is yet another way that the output may not be representative of the grammar.

Note that the external influence is missing in the diagram in figure 2.4.

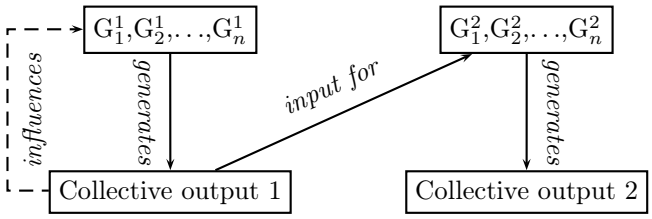


Figure 2.4: Third revision: adult speakers' grammar is influenced by the output of their community.

This is for the following reason. In our model we will only study one form of external influence, namely language contact. We will model contact between language users with different internal grammars by relaxing the condition that the language community $(G_1, \ldots G_n)$ be homogeneous. By identifying certain grammars to be radically different from others, we can in effect model a situation of contact between languages. From here on, consider $(G_1, \ldots G_n)$ to represent a potentially heterogeneous language community.

A second consequence for the model is more difficult to represent in the picture of language transmission we have built so far. It concerns the question of *how* learning takes place. This question relates to both child and adult language learning and it revolves around the question of how to adequately account for both frequency effects as well as analogy in syntax.

Analogy in syntax is often interpreted as consisting of the extension or levelling of syntactic rules across different domains, e.g. verbal or nominal paradigms. Taking the aforementioned definition to heart, syntactic analogy can also be interpreted more strictly in terms of similarity relations that exist between syntactic structures. This view of analogy, of similarity relations between structures (possibly in different domains), has been cited as having cognitive relevance [14] and has produced interesting results in the domain of language [21]. A rule-based grammar formalism is unable to adequately capture these relations however.

The next chapter delves more deeply into the matter of an adequate representation of individual speakers' language models. It introduces data-oriented parsing, a formalism originally used in natural language processing, and shows how it meets our demands in accounting for frequency effects as well as pro-

viding a framework in which we can deal with analogy, both in the sense of extension and levelling of rules and in the sense of structural similarity.

2.2 The final model of language transmission

The model we use in the simulations we will discuss in chapter 4 is basically the model of figure 2.4. Keeping in mind the caveats about the learning mechanisms involved, we extend it to span over more than just two generations, see figure 2.5. The output of the second generation feeds back to form the input of a new grammar. This iterative model forms the basis of our experiments.

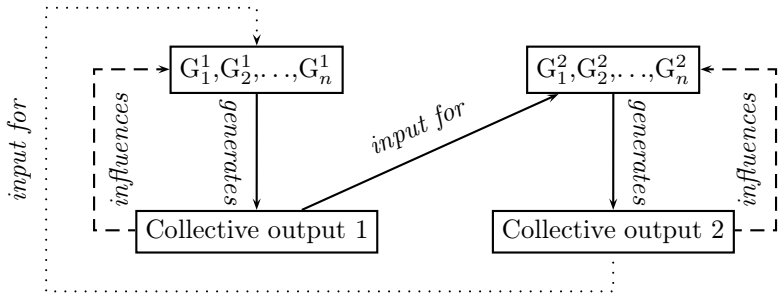


Figure 2.5: Final model of transmission

This diagram leaves open the question of the language model. For an effective implementation, we need a formally precise model of language. From the considerations of the previous sections, we can state a number of desiderate for such a model.

First, it has to be able to account for the frequency effect that were found to be important in language change. Second, the language model has to be able to incorporate the mechanisms of analogy and reanalysis, either as primitive concepts or as emergent features of the model.

A standard generative grammar is unable to meet either of these demands as it is crucially non-probabilistic and most generative theories outright reject the relevance of analogy in linguistics (cf. the remarks by Chomsky [15, p.32].

A third desideratum is that the language model needs to allow a formal mechanism for the acquisition of language, considering both child and adult language learning.

Lastly, we need formal methods for both the generation and interpretation of linguistic structures, if we are to model the generation of linguistic output and the influence this output may have on a language community.

The next chapter discusses a model of human language, Data-Oriented Parsing, which, as we will show, can meet these demands.

Chapter 3

Technical aspects of the language model

The previous chapter outlined a basic model of language transmission. It described a list of desiderata for a language model. A language model that can be used in our model has to incorporate frequency and has to be able to account for analogical factors. Reanalysis especially was found to be important.

From a computational perspective, we can add three more desiderata. The implementation of the language model has to be *efficient* (since we will use it in an iterated simulation of transmission) and it has to be *precise* (since imprecisions are likely to add up over repeated application). It also has to be fully formalized.

This chapter first gives a description of data-oriented parsing, which will be used as a language model. We will explain its basic properties and some variants. We also explain why it meets the demands. After that we describe to describe the parsing mechanism we use, which is efficient and precise. Lastly we discuss the issue of syntactic and semantic representation and their relation in the simulations.

3.1 Data oriented parsing

3.1.1 Introduction

This section gives a short introduction to data-oriented parsing, the formal model that underlies this thesis. Data-oriented parsing [46] is a particularly apt framework for modelling analogy in natural language. It is based on exemplars and attempts to capture the structural analogies between sentences and corpora.

Data-oriented parsing builds on the idea that there is a continuum between grammatical rules on the one hand and grammatical structures or *constructions*, represented as parse trees on the other. It considers neither to be primary but attempts to build a framework which subsumes both notions. The primitive units of the grammar are *all* possible subtrees, encompassing both grammatical rules (represented simply as subtrees of depth 1) and wider reaching constructions (represented as subtrees of greater depth). These units can be combined to form full parse trees by a substitution operation.

This approach has found some remarkable success in the field of natural language processing (for an overview, see [8]). It meets the demands for a language model stated in the previous chapters on two grounds. Firstly, the most widely used DOP models are probabilistic, thus incorporating frequency of occurence of linguistic forms (see section 3.1.2). Secondly, data-oriented parsing has been described as an attempt to construct the maximal ahalogy between an input sentence and a collection of exemplar structures [46]. Taken together, this means the language model is able to model analogy both in the extension/levelling sense and in the structural similarity sense.

The next section introduces the formalities of the DOP model. After gaining familiarity with its basic properties, we return to the question of how DOP can meet the demands we listed for a viable language model.

3.1.2 The probability model

A DOP grammar is constructued by extracting all possible subtrees from a training corpus of syntactically annotated sentences. As an example, we show how to extract a DOP-model from the simple training corpus depicted in figure 3.1.2 consisting of a single sentence and its tree structure.

For every tree t in this corpus, we extract *every* subtree and assign a weight to it [6].

The first DOP model [3], DOP1, defined the associated weights as follows.

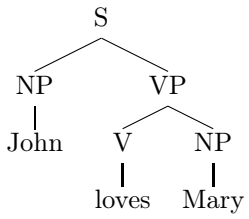


Figure 3.1: A simple example training corpus

Let $\mathcal{X}$ be a countable set of types, in this case tree fragments. Then a corpus is a function $f : \mathcal{X} \to \mathbb{N}$ and for each $x \in \mathcal{X}$, $f(x)$ is a type frequency. The type frequency indicates the number of times a particular subtree occurs in the corpus. Let $r(t)$ denote the root node of tree fragment t, then the probability assigned to t is defined by equation 3.1.

$$p(t) = \frac{f(t)}{\sum_{t':r(t')=r(t)} f(t')} \tag{3.1}$$

Figure 3.2 shows the set of extracted subtrees along with their probabilities.

This collection of subtrees along with their weights is used to form a Probabilistic Tree Substitution Grammar (PTSG). To parse a sentence, the tree fragments can be recombined by means of the labelled substitution operation, denoted by $\circ$. As an example, figure 3.1.2 show how the tree fragments can be combined to form a derivation for the sentence *Mary loves John*, which was not in the training corpus.

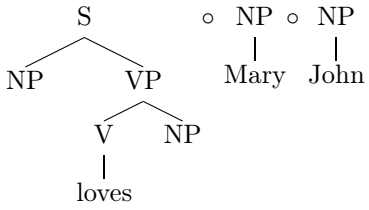


Figure 3.2: Example of labelled substitution; the derivation of the parse tree for *Mary loves John*

The probability of a derivation consisting of tree fragments $t_1, \ldots, t_n$ is given by equation 3.2.

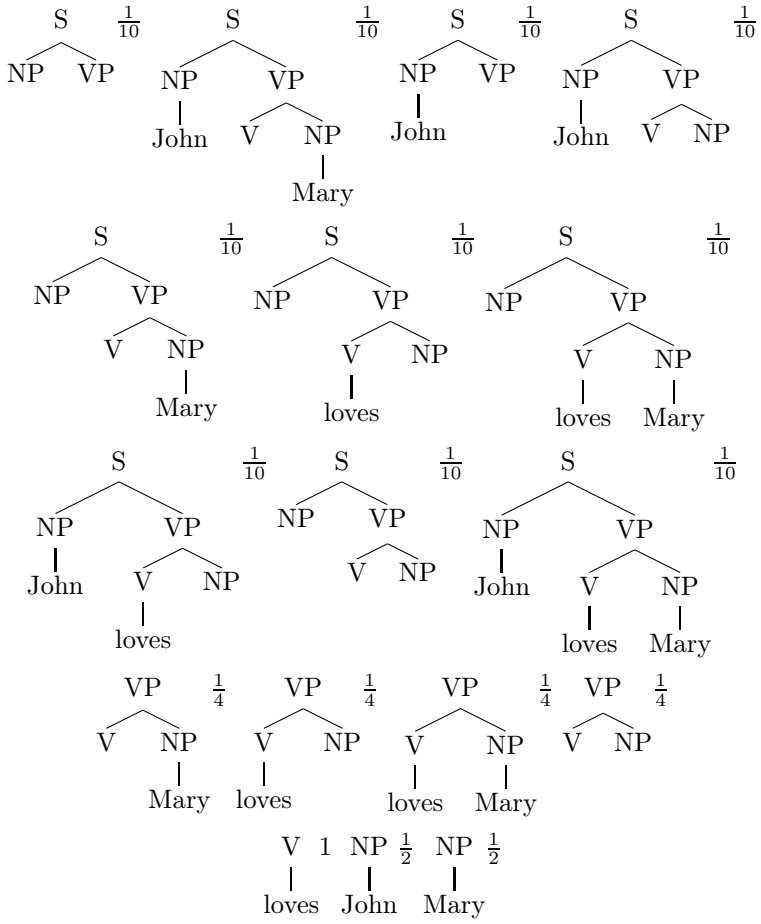


Figure 3.3: DOP1-grammar extracted from the corpus in figure 3.1.2, showing all extracted subtrees and their associated weights.

$$p(t_1 \circ \ldots \circ t_n) = \prod_{i=1}^{n} p(t_i) \tag{3.2}$$

A parse tree can be formed by many derivations. The probability of a parse tree T is given by the sum of the probabilities of the derivations D producing that parse tree, equation 3.3.

$$p(T) = \sum_{\text{D derives T}} p(d) \tag{3.3}$$

3.1.3 DOP and analogy

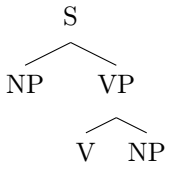
The data-oriented parsing framework was chosen as the language model in this thesis because of two reasons. Firstly, it incorporates the frequency of occurence of structures, found to be important in the previous chapter. Using DOP, the speakers in our simulation will exhibit a preference to use parts of highly-frequent structures above lesser-frequent ones. Over generations, the probabilities attached to structures may change and new structures may be preferred. In a probabilistic setting, this is what constitutes language change. Note that this is essentially different from the generative view on language change. There, a difference in parameter settings means that *new rules* come into play. Syntactic change is essentially rule change. In our model, new structures may come into existence, but the essence of syntactic change is a change in the frequency of structures in such a way that a different structure comes to be the preferred one.

But there is another factor to the use of probabilistic models. Reanalysis crucially relies on the possibility of more than one analysis of a given construction being available (cf. [13, p.284]), that is, it relies on the *ambiguity* of analysis. Most sentences in natural language are syntactically ambiguous. In general, the longer the sentence, the more possible parses exist. Yet we usually perceive only one of the parses as being relevant. This poses a problem for natural language processing systems. Probabilistic grammars have sought to account for this problem by attaching weights to parses and taking the highest-ranking parse to be the 'correct' one. This relates directly to the issue of reanalysis.

When multiple analyses of a given form are available (and they often are), the real question about reanalysis is why a certain analysis comes into favour at the expense of another. For this we need the fine-grained system that a probabilistic grammar can give us. Reanalysis is the essential mechanism of change in the model we use in this thesis. As subtree probabilities change, the preferred analysis may change. Analogy may then spread this analyis across paradigms and so an overt change in form may take place.

While the use of probabilities is for our purposes an advance over non-probabilistic grammar models, such as the standard generative grammar, it is by no means unique to data-oriented parsing. The advantage that DOP has over other probabilistic language models is its ability to take into account large constructions. For our purposes, this means that analogies between structures are naturally expressed within the system. For example, consider sentences that

share the following subtree in their parse:



These sentences have something important in common, that is not easily expressed in other systems. While this is a trivial example, it does show that using DOP means that the speakers in our simulation will have access to a large array of exemplars in the form of syntactic structures, with which they can form analogies to extend to new sentences. In fact, parsing under this framework is explicitly seen as maximizing the structural analogy between an input sentence and a corpus of previously observed structures [46]. Combined with the probabilistic nature and the importance of ambiguity this means that both analogical levelling and extension can be modelled by using DOP in an iterated setting.

A note about language learning. Above we discussed how a DOP-grammar can be extracted from a treebank. Naturally, this is not how children learn a language, since the primary linguistic data does not in general provide evidence of the syntactic analyses of its sentences. The best way to model language acquisition in this model would be to have an *unsupervised* mechanism of learning tree-structures (such as proposed in [6, 7]). However, these systems are prohibitively slow for our purposes. So we take a shortcut and assume that child language acquisition does in fact proceed by extracting a grammar from a syntactically annotated bank of sentences. It may be objected that this is an unwarranted oversimplification. Experiments with unsupervised DOP show, however, that good results on learning the correct bracketing can be achieved. We presume therefore that taken together with a system of category induction *in principle* methods can be designed that produce results that rival those of supervised DOP. This assumption provides a significant gain in computational speed.

3.1.4 Other estimators for data oriented parsing

The DOP1 estimation method was shown to be inconsistent and biased [31]. This means two things. (1) As the training corpus grows to infinity, the DOP1 estimator does not converge on the relative frequency estimate of the parse trees in the treebank that is used to train it. This is called *inconsistency*. (2)

The DOP1 estimator is able to assign non-zero probability to trees that do not occur in the training corpus. As we saw above, new parse trees can be derived by applying the substitution operation on subtrees that occur in the treebank. This means that it is *biased* with respect to the relative frequency estimate.

This is not necessarily a bad thing. If an estimator is not biased, it can only assign non-zero probability to trees that occur in its training corpus, thus eliminating the use of it for parsing novel sentences. However, the DOP1 estimator assigns unproportionally more probability mass to subtrees that are part of large parse trees, since the number of subtrees grows exponentially with the size of the parse tree.

Several efforts have been made to remedy these problems. This thesis investigates three other estimation methods by studying how they behave under the iterated transmission model.

The first variant estimator was proposed by Bonnema and Scha [9] (henceforth: the Bonnema estimator). They propose equation 3.4 to estimate the probability of a subtree, where $N(t)$ is the number of non-root non-terminals in t and $F(t)$ represents the number of times the root of t occurs in the training data. This estimator remedies the bias problem by

$$p(t) = \frac{2^{-N(t)} f(t)}{F(t)} \tag{3.4}$$

Another way to remedy the bias to deep trees comes from [26]. Assigning equal weight to each node in the training data and equal weight to each subtree produces equation 3.5, where $F(t)$ is the number of times the root non-terminal of t occurs in the training data and $n(t)$ is the number of subtrees headed by the root non-terminal of t.

$$p(t) = \frac{f(t)}{F(t)n(t)} \tag{3.5}$$

The estimator by Bonnema and Scha and the Equal Weight estimator from [26] share some desirable properties. Compared to the DOP1-estimator, less probability mass is reserved for the larger trees in the training data. The result is a more even distribution of mass amongst the subtree fragments.

The last DOP-model we consider here takes a different approach, being non-probabilistic. Bod [5] proposes to compute the parse tree which can be generated by the shortest derivation, i.e. the derivation which uses the least amount of subtrees. This has the effect of using the largest possible subtrees, thus

maximizing syntactic context. Since the shortest derivation is not necessarily unique, Bod proposes to back-off to probabilistic DOP as a tie-breaker. For the purposes of this thesis, tie-breakers will not be necessary, as we will never only use the 1st best parse. We therefore consider only a 'naive' version of the shortest derivation estimator, calculating only the smallest number of subtrees used.

The non-probabilistic shortest derivation version of DOP can be cast into the probabilistic framework by assigning the same probability to every subtree such that $p \in [0 \dots 1]$. This has the effect of making the shortest derivation the most probable one. A derivation using n subtrees will have probability p^n. The derivation using the smallest amount of subtrees will then have the largest probability.[1]

Other DOP models than the ones described above exist (e.g. [57] which has been proven to be consistent). The described DOP-models were chosen for the following reasons. The DOP1 model is generally used as a reference point, being the first and most straightforward probability model. The estimator by Bonnema and Scha and the Equal Weights estimator were used as counterpoints to the DOP1-model, in an attempt to relieve the bias problems the latter has. As [7] points out, the shortest derivation model provides a natural way of modelling the maximization of structural analogy. It gives a metric of the *simplicity* of syntactic structure, 'thereby maximizing the structural commonality between a sentence and previous sentence-structures' [7]. Since we are interested in exactly the structural analogy between sentences and previously observed utterances, this estimator is a natural choice.

3.1.5 PCFG reduction of DOP grammars

This section can be safely skipped by the reader that is mainly interested in the linguistic aspects of this thesis. It concerns a technical aspect of the implementation that is irrelevant for our linguistic argument.

The number of subtrees that can be extracted from a training corpus grows exponentially as the corpus grows [47]. To keep an implementation computationally tractable, two approaches can be taken.

1. Reduce the number of fragments, for example by only including subtrees

[1] Note that the use of the word 'probability' here is actually improper, since they do not form a probability distribution. It would be more proper to refer to them as *weights*, but keeping in line with common usage, we refraing from doing so.

up to a certain depth.

2. Reduce the PTSG to a Probabilistic Context-Free Grammar (PCFG) [26]

This thesis takes the latter approach.

Goodman [26] defines a simple scheme for reducing DOP PTSG grammars to an equivalent PCFG, that contains at most eight rewrite rules for each node in the training data. This means that the PCFG is linear in size to the training data, instead of exponential. This allows for much faster parsing using the full DOP grammar.

The conversion relies on relabelling the non-terminal nodes in the training corpus. Each node is assigned a unique address. For example, $A@k$ denotes the node at address k with non-terminal label A. For each non-terminal node in the training data, a new non-terminal is created, in the example this node is called A_k. Non-terminals of this form are called 'interior' non-terminals, while the original non-terminals are called 'exterior'. Figure 3.1.5 shows the relabelled tree from figure 3.1.2

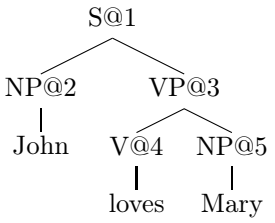
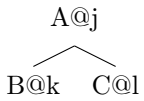


Figure 3.4: The relabelled tree from 3.1.2

Let a_j represent the number of subtrees headed by node $A@j$. Let a represent the number of subtrees headed by nodes with non-terminal A, i.e. $a = \sum_j a_j$.

Goodman next shows how to construct a PCFG that generates for every subtree in the training corpus headed by A a homomorphic subderivation with probability $1/a$. A PCFG subderivation is called homomorphic to an PTSG elementary tree if the subderivation begins and ends with external non-terminals, and uses internal non-terminals for intermediate steps. A PCFG derivation is homomorphic to an PTSG derivation if for every contributing subtree in the latter there is a corresponding subderivation in the PCFG. A PCFG tree is homomorphic to an PTSG tree if they are identical modulo the address labels at the nodes.

For a node like



eight PCFG rules will be generated as follows:

$$\begin{array}{llll} A_j \to BC & (1/a_j) & A \to BC & (1/a) \\ A_j \to B_k C & (b_k/a_j) & A \to B_k C & (b_k/a) \\ A_j \to BC_l & (c_l/a_j) & A \to BC_l & (c_l/a) \\ A_j \to B_k C_l & (b_k c_l/a_j) & A \to B_k C_l & (b_k c_l/a_j) \end{array} \tag{3.6}$$

Goodman goes on to show that subderivations headed by A with external non-terminals at the roots and leaves and internal non-terminals elsewhere have probability $1/a$, while subderivations headed by A_j with external non-terminals only at the leaves have probability $1/a_j$. Furthermore, the construction described above produces PCFG trees homomorphic to the PTSG trees with equal probability.

The PCFG described above corresponds to a DOP1 PTSG, but the other estimators introduced in section 3.1.4 can be treated in the same way.

The probability model from Bonnema and Scha [9] described by equation 3.4 is equivalent to the PCFG schema in equation 3.7, where $\overline{a}$ is the number of times non-terminals of type A occur in the training data.

$$\begin{array}{llll} A_j \to BC & (1/4) & A \to BC & (1/4\overline{a}) \\ A_j \to B_k C & (1/4) & A \to B_k C & (1/4\overline{a}) \\ A_j \to BC_l & (1/4) & A \to BC_l & (1/4\overline{a}) \\ A_j \to B_k C_l & (1/4) & A \to B_k C_l & (1/4\overline{a}) \end{array} \tag{3.7}$$

The shortest derivation model from [5] corresponds to the equivalent PCFG in equation 3.8.

$$\begin{array}{llll} A_j \to BC & (1) & A \to BC & (1/2) \\ A_j \to B_k C & (1) & A \to B_k C & (1/2) \\ A_j \to BC_l & (1) & A \to BC_l & (1/2) \\ A_j \to B_k C_l & (1) & A \to B_k C_l & (1/2) \end{array} \tag{3.8}$$

Lastly, the equal weights estimator described in equation 3.5 reduces to the PCFG in equation 3.9

$$\begin{array}{llll} A_j \to BC & (1/a_j) & A \to BC & (1/a_j\overline{a}) \\ A_j \to B_kC & (b_k/a_j) & A \to B_kC & (b_k/a_j\overline{a}) \\ A_j \to BC_l & (c_l/a_j) & A \to BC_l & (c_l/a_j\overline{a}) \\ A_j \to B_kC_l & (b_kc_l/a_j) & A \to B_kC_l & (b_kc_l/a_j\overline{a}) \end{array} \tag{3.9}$$

The last reduction from the equivalence in equation 3.5 is not proven explicitly in [26], so we take the opportunity here to show the reasoning behind the soundness of the reductions.

Theorem 1 (Subderivation probability for Equal Weight DOP)**.** *Subderivations under the PCFG of equation 3.9 headed by A with external non-terminals at the roots and internal non-terminals elsewhere have probability* $1/a_j\overline{a}$*. Subderivations headed by* A_j *with external non-terminals only at the leaves and internal non-terminals elsewhere have probability* $1/a_j$

Proof. Proof by induction on the depth of the trees.

Base case: depth = 1. There are two cases:

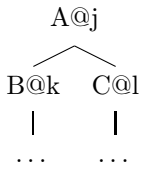
For these cases the theorem holds trivially.

Inductive step: assume that the theorem holds for trees of depth $\leqslant n$. We have to show that it holds for trees of depth $n+1$. Naturally, there are eight cases, one for each of the PCFG rules corresponding to the node. We show four of them. The others follow by comparable reasoning.

$$1. \quad A_j \to B_kC_l \quad (b_kc_l/a_j)$$

Let

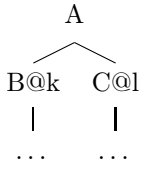
represent a tree of at most depth n with external non-terminals only at the leaves, headed by $B@k$. Then for trees like



the probability is $\frac{1}{b_k}\frac{1}{c_l}\frac{b_k c_l}{a_j} = \frac{1}{a_j}$.

2. $A \to B_k C_l \quad (b_k c_l / a_j \overline{a})$

For trees like



the probability is $\frac{1}{b_k}\frac{1}{c_l}\frac{b_k c_l}{a_j \overline{a}} = \frac{1}{a_j \overline{a}}$.

3. $A \to BC \quad (1/a_j \overline{a})$

For trees like

the probability is $\frac{1}{a_j \overline{a}}$.

4. $A \to B_k C \quad (b_k / a_j \overline{a})$

For trees like

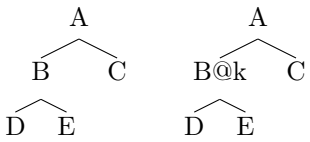
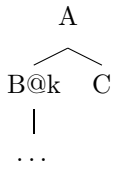


Figure 3.5: A PTSG subderivation (left) and a homomorphic PCFG subderivation (right). The PCFG subderivation has probability $\frac{b_k}{a_j \overline{a}} \frac{1}{b_k}$, while the PTSG tree will have probability $\frac{f(t)}{F(t)n(t)}$.



the probability is $\frac{1}{b_k} \frac{b_k c_l}{a_j \overline{a}} = \frac{1}{a_j \overline{a}}$.

The other four cases follow trivially. □

The proofs for the other estimators follow by similar reasoning, as outlined in [26].

Theorem 2 (PCFG to PTSG equivalence)**.** *The construction produces PCFG trees homomorphic to PTSG trees with equal probability.*

Proof. Assume every subtree in the PTSG occurs only once in the training data. The proof under this assumption is easy. Consider an arbitrary PTSG subderivation and a homomorphic PCFG subderivation. For example the ones shown in figure 3.5. The PTSG subderivation has a probability of $\frac{f(t)}{F(t)n(t)}$ as defined by equation 3.5. The homomorphic PCFG subderivation will have a probability of $\frac{1}{a_j \overline{a}}$ as was proven above. If $f(t) = 1$, as we assumed, the two subderivations will have equal probability.

Naturally, actual treebanks are likely to contain multiple occurrences of some subtrees, so the assumption would seem to be unjustified. If the PTSG formalism is changed slightly so that the the bag of subtrees is a multi-set, i.e. subtrees can occur more than once and their counts are not merged, then the one-to-one relationship holds. This holds since the probabilities of parse trees do not

change since they are obtained by summing over all derivations, automatically taking into account the multiple occurrences of subtrees and unmerged counts. So, summing over all PTSG derivations yields the same probability as summing over all homomorphic PCFG derivations. □

3.2 k-best Parsing

Like the previous section, this section can be safely skipped by the reader who is mainly interested in the linguistic argument of this thesis.

The simulations we will introduce in the next chapter rely heavily on k-best Viterbi lists as output from the parser. As we explain there, we will use k-best lists as approximations of the occurrence frequency of alternative analyses of syntactic structures in the iterated model.

As in [5] we approximate the most probable parse by summing over k-best lists. Finding the most probable parse for DOP is NP-hard [48], so an approximation which allows us to efficiently parse sentences is very welcome indeed. Other solutions exist, for example Monte Carlo sampling of the parse forest [4], or aggressively pruning the search space. The former has the disadvantage that progressively larger numbers of trees need to be sampled as the sentence length grows, eventually becoming inhibitively large, while the latter method produces relatively low quality k-best lists.

So, for our implementation of the model to be efficient enough to be feasible, we need fast k-best parsing. In [29], Huang & Chiang elaborate on the proposal of Klein & Manning [35] to use weighted directed hypergraphs in probabilistic chart parsers. Huang & Chiang generalize this proposal to introduce a framework which uses hypergraphs to represent the search space of weighted deductive systems.

So for this thesis we used their hypergraph parser, implemented bottom-up as a variant of the CKY-algorithm [32, 56], outputting k-best Viterbi lists.

3.2.1 Hypergraph parsing

The key idea behind Huang & Chiang's hypergraph parser is to use hypergraphs to construct the chart of a parser instead of normal graphs. Simply put, the difference between standard directed graphs and hypergraphs is that while standard arcs connect a single tail node to a single head node, hyperarcs connect a

set of tail nodes to a set of head nodes. The definition of a weighted hypergraph is slightly to suit the purposes of a probabilistic parser.

Definition 1 (Weighted hypergraph)**.** *A hypergraph H is a tuple $\langle V, E, t, R\rangle$, where V is a finite set of* vertices*, E is a finite set of* hyperarcs *and R is a set of* weights*. Each hyperarc $e \in E$ is a triple $e = \langle T(e), h(e), f(e)\rangle$, where $h(e) \in V$ is the* head *and $T(e) \in V^*$ is a vector of* tail *nodes. $f(e)$ is a* weight function*, $f(e) : R^{|T(e)|} \to R$. $t \in V$ is a distinguished vertex called the* target vertex*. If $|T(e)| = 0$, then $f(e)$ is a constant and $h(e)$ is a source vertex.*

The parser traverses the hypergraph in the order defined by the topological ordering of the hypergraph's graph projection, while updating the result of the weight functions f along the hyperarcs. The update process is simply the application of the normal Viterbi update, i.e. combining the results of subspans with the probability of a grammar rule to find the result for the superspan. If successful, i.e. the prespecified target vertex t is reachable from the source vertices, this results in a packed forest represented by the hypergraph (see the example in figure 3.6).

The goal is to find the k best *derivations* from the resulting packed parse forest.

Definition 2 (Derivation)**.** *Let $inc(v)$ be the set of incoming hyperarcs to a vertex, i.e. $inc(v) = \{e \in E | h(e) = v\}$. A* derivation *$D$ and its weight $w(D)$ are defined as:*

- *if $e \in inc(v)$ with $|T(e)| = 0$, then $D = \langle e, \epsilon\rangle$ is a derivation of v and its weight $w(D) = f(e)()$.*
- *if $e \in inc(v)$ with $|T(e)| > 0$ and D_i is a derivation of $T_i(e)$ for $1 \leqslant i \leqslant |T(e)|$, then $D = \left\langle e, D_1, \ldots, D_{|T(e)|}\right\rangle$ is a derivation of v, and its weight $w(D) = f(e)(w(D_1), \ldots, w(D_{|T(e)|}))$*

Derivations are ordered by their weights, so we can define $D_i(v)$ to be the i^{th} best derivation of v. The goal of the parser can then be restated as: find $D_1(t), \ldots, D_k(t)$. The derivations can be extracted from the parse forest in a way that is analogous to the use of backpointers in other parser implementations.

So far the concepts we described are just methods to describe the encoding of an exponentially large search set in polynomial space. The ingenious innovation in Huang & Chiang's article lies in the application of dynamic programming techniques to the final search for the ordered set of best derivations. Instead of

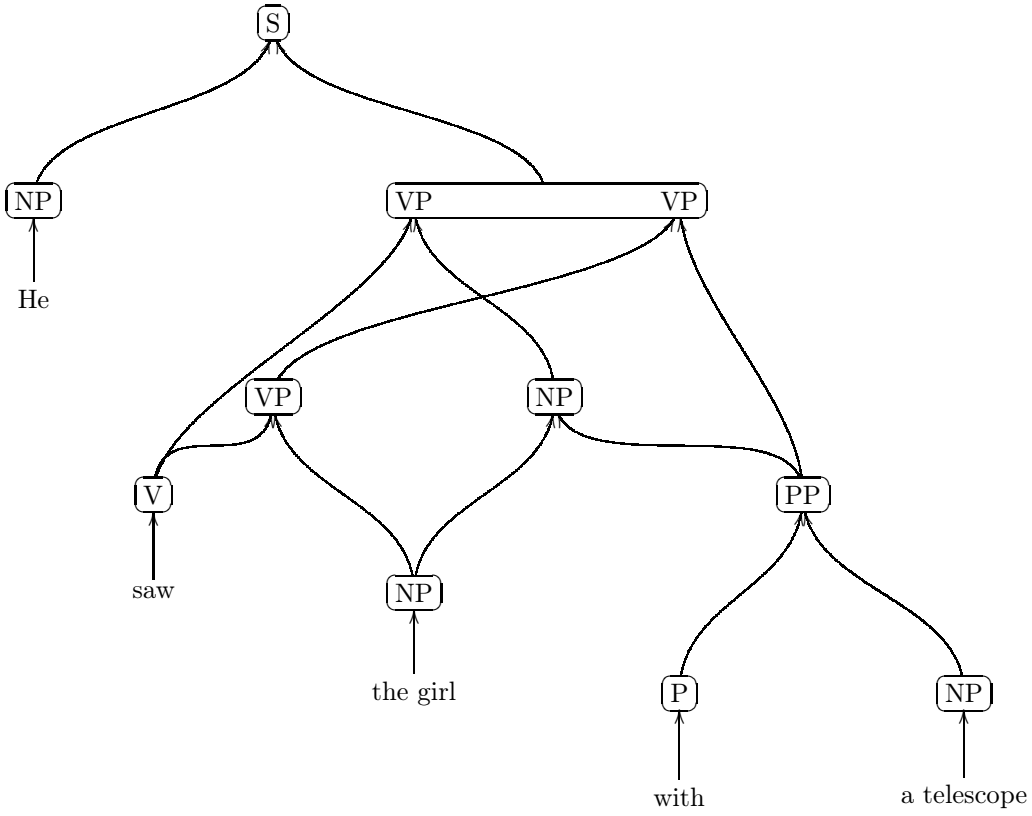


Figure 3.6: Hypergraph representation of the packed parse forest for the sentence *He saw the girl with a telescope*. Items in rounded boxes represent the *vertices* of the hypergraph, joining arrows represent the *hyperarcs*.

finding the k-best derivations for each vertex, the computation is delayed until the target vertex is reached. The algorithm then consists of two phases:

1. A forward phase, which produces the 1st best derivation, but also stores all alternative derivations.

2. A backward phase, which recursively searches for the second best derivation, until k is reached.

The complete algorithm then has overall complexity of $O(|E| + nk \log k)$, where n is the number of words in the input sentence (details and proof of complexity in [29]). We made a small improvement on the space requirements, by using

a k-bounded priority queue[2] for storing the candidates for each vertex. The overall space complexity becomes $(|E| + kn)$

In summary, the Huang & Chiang algorithm allows us to do precise k-best parsing in a reasonable time.

3.2.2 Flip-reverse parsing

The models we introduce in the next chapter frequently require us to deal with input that is unparsable due to word order differences. As an example, consider the grammar in figure 3.2. Equipped with this grammar, the parser will not be able to derive the target vertex "S" for the sentence *John Mary loves*, meaning `loves(John,Mary)`. However, for our purposes, we presume that speakers *will* be able to make a guess about the meaning of the sentence and therefore will need to be equipped with a parsing mechanism that will always output *some* result in case parsing fails, however suboptimal that result may be from the point of view of the grammar.

We propose the following, simple method for dealing with this kind of unparsable input. If, after the first pass through the chart of the projected hypergraph (as described in the previous section), the top cell does not contain the TOP label of the grammar ("S" in this case), a second pass through the chart is made. This pass, still in topological order, is allowed to modify the chart in the following way. For any two nodes with non-overlapping spans it passes over in the chart, the parser checks whether the grammar contains a rule that contains the node labels in reverse order as the right hand side. If this is the case, the parser adds a new hyperarc to the chart. This hyperarc has the rule's left hand side as head node in the usual way and the rule's right hand side as the tail vector, effectively 'flipping' the construction around. Figure 3.2.2 gives a simplified example of a flip in the parsing of *John Mary loves*.

The transitional probability associated with the hyperarc is adjusted from the rule's probability by a factor a, to give preference to lesser numbers of flips over greater. Once this second pass is completed, the parser checks whether the target vertex has been reached. If it has the parser is done, if it hasn't, the process is repeated until either the target vertex has been reached or the last pass did not add new hyperarcs to the chart. In the latter case, the sentence is declared unparsable by the parser. This method provides a simple way to construct trees for constructions that are not allowed by the grammar, while

[2]Backed by a min-max heap[2]

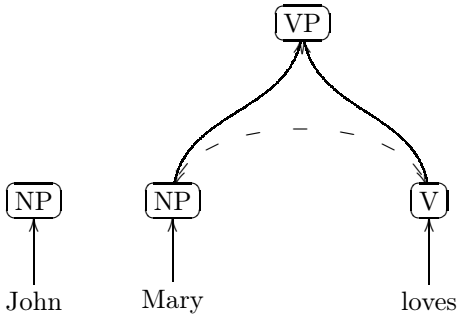


Figure 3.7: Operation in second traversal of the hypergraph. The nodes *NP* and *V* are being flipped to form a *VP* that was previously not derivable from the grammar.

still giving preference to the ones that are.

Note that a similar way to do this, would be to enrich the grammar with copies of all the rules with their right hand side flipped around. The advantage to this approach would be that the probabilities of the rules with the same left hand side form a proper probability distribution. This distribution is made improper by the approach we take here. There are, however, two reasons for not doing so. The first is that doing so will naturally double the size of the grammar, which will in practice already be quite large. Our approach allows us to generate the 'flipped' rules only when needed, thus retaining the original size of the grammar. A second reason for not doing so requires us to look ahead to the next chapter. There we will discuss that the grammar, aside from parsing, will also be used to calculate the probability of existing parse trees in order to rank them. For this application we do *not* want to include flipped rules in our calculations. We would therefore need to store two separate grammars for each of the agents in our simulation. This is unwanted as we wish to keep our memory usage to a minimum.

3.3 Representations of syntactic and semantic structures

As outlined above, syntactic structure in our model is represented by phrase structure trees. For the implementation we introduce in the next chapter we also need a representation of semantic structure and algorithms for conversion

between the two. We chose to use dependency trees to represent semantic structures as this allows for simple and fast conversion to and from phrase structures.

The semantic representations are used as a generative system for DOP, by first generating all possible trees corresponding to a semantic structure and then letting the grammar decide which tree structure is the most probable. In practice we predefine the semantic structures for a given simulation, depending on the type of phenomena we are interested in.

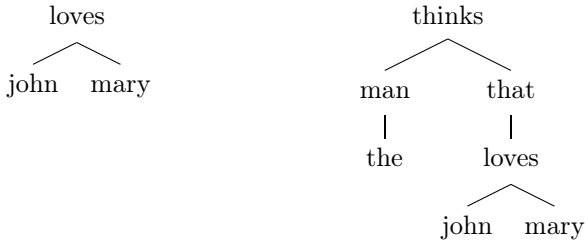
Dependency structures form a very limited means of semantic representation. For our purposes, however, we are only interested in representing simple semantic structures, such as

```
loves(john, mary)
```

or slightly more complicated, a definite description à la Russell, with an embedded predicate:

```
(thinks(ιx(man(x)),loves(john,mary)))
```

These two formulas are represented (roughly) as follows:



We define a symmetric equality relation on the nodes in these structures:

Definition 3 (Equality of nodes in a semantic structure)**.** *Two nodes, n_1 and n_2, in a dependency tree are equal iff:*

1. *The label of n_1 is the same as the label of n_2*
2. *The semantic annotation of n_1 is the same as the semantic annotation of n_2*
3. *For each node dependent on n_1 there is a node dependent on n_2 that is equal to it and vice versa*

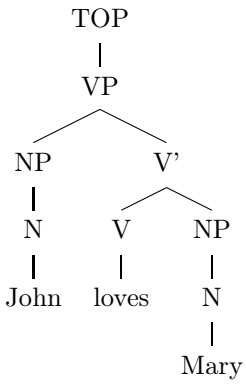
From this definition, we define equality of entire structures:

Definition 4 (Equality of semantic structures)**.** *Two dependency trees are equal iff their top nodes are equal*

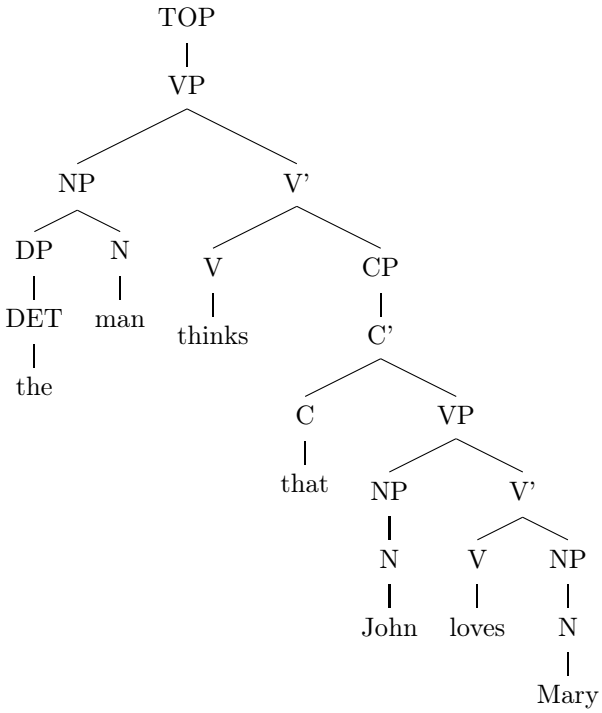
For our syntactic representation the following conventions are used, which bear some resemblance to the standard x-bar conventions

1. Categories always project maximally, e.g. *N* projects up to *NP*.
2. Arguments attach to an intermediate projection of the head.
3. Specifiers attach to the maximal projection of the head.
4. Categories only project if necessary (save for rule 1), so if a category has no arguments, it has no intermediate projections.
5. The root node of a complete phrase structure tree is attached to a *TOP*-node, for parser purposes.

As examples, the phrase structure tree corresponding to *John loves Mary* looks like this:



while the tree for *The man thinks that John loves Mary* looks like this:



The conversion from phrase structure tree to dependency tree (i.e. semantic structure) is straightforward and follows the algorithm outlined in [55], with the caveat that the phrase structure conventions used in that article are slightly different from the ones used here. Let the *lexical head* of a projection node *XP* be the daughter of the preterminal category node that produced the projection and the *head child* be the daughter node of the *XP* node that is the ancestor of the lexical head. The conversion algorithm then takes the following steps:

1. Mark the lexical head of each node in the phrase structure tree.
2. Construct the dependency structure by making each lexical head of each non-head-child depend on the lexical head of the head-child.

The conversion from dependency trees to phrase structure trees is less straight forward. The reason for this is that we want the conversion to be one-to-many. In the next chapter, we will model word order variety. For that we will need each semantic structure to correspond to a set of parse trees, reflecting all possible word orders. We use the following method for obtaining the unrestricted set of all phrase structure trees S from a dependency node V, that is to say, the set that puts no constraints on the scrambling of arguments, modifiers or specifiers.

The elements of this set may therefore violate rules 2 and 3 of the conventions for phrase structure trees we outlined above. The set is defined recursively.

Definition 5 (Unrestricted set of phrase structure trees corresponding to dependency node V)**.** *Base case: V has no daughters. Then S is the singleton set consisting of the maximal projection of the category of the node's label.*

Recursive case: V has daughters $d_1, \ldots, d_n$. Let $D_1, \ldots, D_n$ be the sets of phrase structure trees corresponding to these daughters. For each permutation of $\langle D_1, \ldots, D_n \rangle$ generate all combinations of elements of $D_1, \ldots, D_n$ with V by attaching elements from $D_1, \ldots, D_{n-1}$ to intermediate projections of V's label's category and attaching elements of D_n to V's label's category's maximal projection.

As an example, figure 3.8 shows the set of phrase structure trees corresponding with `loves(john, mary)`. Note that S quickly grows very large as $|S| = n! \cdot |D_1| \cdot \ldots \cdot |D_n|$.

This concludes the discussion of the language model. The next chapter describes the implementation combining these ideas with the transmission model as well as several experiments.

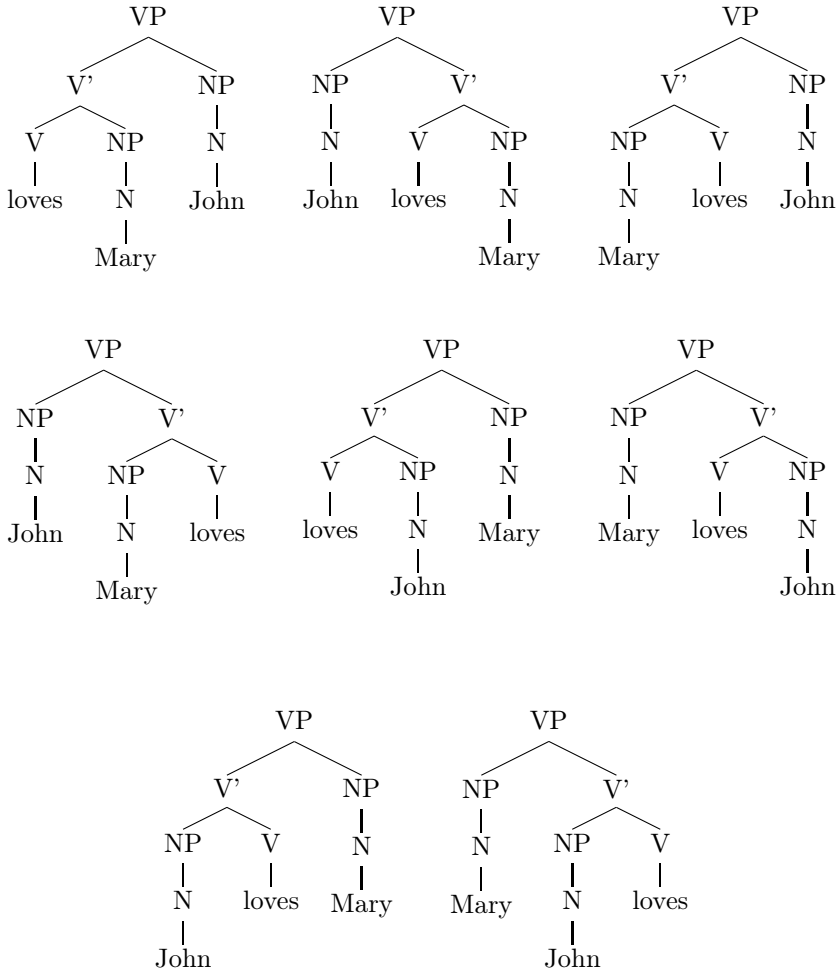


Figure 3.8: Phrase structure trees generated from `loves(john,mary)`

Chapter 4

Simulations and Results

The two previous chapters explored the features that an implementation of a model of language change should have. We looked at models of language transmission and discussed the properties of a suitable language model. This chapter describes several experiments based on these models. The first section describes a simulation that models reanalysis and analogical levelling. The next section adds semantic and phonological features and simulates the relation between word order freezing and loss of inflection. The final section models a situation of language contact and adds a heterogeneous language community. This last simulation closely adheres to the final model of transmission set out in chapter 2. All simulations are used to provide examples of how known phenomena in historical linguistics can be studied using computational methods.

4.1 Reanalysis and analogical levelling

This section describes a simulation focusing on reanalysis and analogical levelling. The model used is fairly simple, but we will show that even so, some interesting results can be obtained. We will first describe the basic implementation. After that we will discuss two examples from historical linguistics that will serve as the starting point for our simulation. Lastly, we will discuss the results from these simulations.

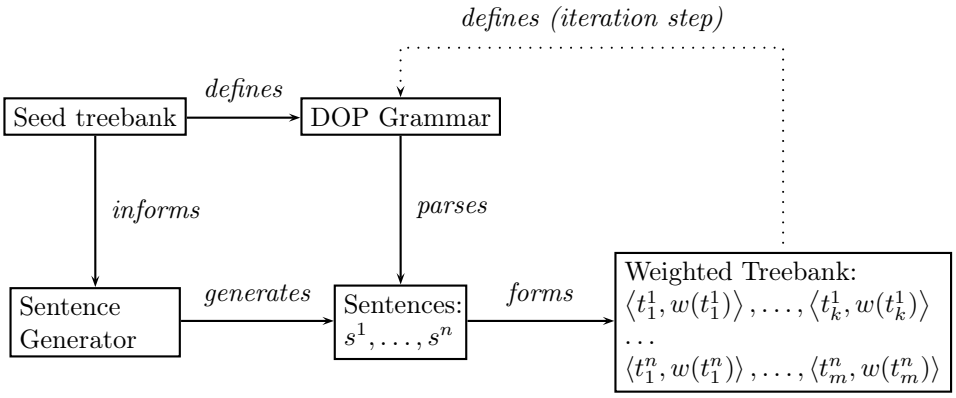


Figure 4.1: Diagram showing the basic model for the simulation of reanalysis and analogical levelling.

4.1.1 The basic model

The diagram in figure 4.1 shows the basic model used in this simulation. The idea is to model the repeated generation and parsing of sets of sentences. The parses of one set of sentences is used to extract the grammar for the next iteration. For this we need two components. (1) A grammar. (2) A module that generates sentences for the grammar to parse.

The initial grammar is extracted from a pre-defined treebank. From this treebank is also extracted a lexicon, consisting of words with associated part-of-speech tags. These are stored in the Sentence Generator module together with all the POS-tag sequences of the sentences in the seed treebank. The POS-tag sequences are used to generate sets of sentences $S = \{s^1, \ldots, s^n\}$, by replacing each POS-tag in a sequence with a randomly chosen word of that category. This procedure is used as a kind of faux semantic module, making sure that at every iteration, the grammar has a new set of sentences to parse. Each sentence $s^i \in S$ is subsequently parsed using the extracted grammar, keeping the full range of non-optimal parses output by the k-best parser. The resultant k-best list forms the input for the extraction process of the grammar of the next iteration.

The k-best list contains all the analyses of a sentence that are licensed by the DOP grammar. Since we are interested in modelling how ambiguous forms may behave over repeated cycles of parsing and learning, we need a way to transmit

all allomorphs to a next generation. We therefore interpret the probability of an analysis as an indication of the relative frequency of its occurrence in the grammar's output. As an example, consider the form xx, that can be parsed in two ways under a certain grammar, with the following probabilities.

In the simplest form, we then consider the left analysis to occur twice as often as the right one, since its probability is twice as high. In this simulation, the next generation will therefore be presented with a treebank in which the analyses occur in a 2 : 1 ratio. We may, however, wish to allow for the parse with the highest probability, the most salient analysis, to be significantly different from the other parses. This can be implemented by optionally reserving an amount of probability mass for the 1st-best parse. This mass will 'boost' its probability and is therefore designated by b, for *boost mass*. The weighted treebank then defines the grammar for the next generation. All subtree-frequencies are simply scaled by their weight in the weighted treebank.

The formal definition of a weighted treebank is as follows. A Weighted Treebank W holds a number of sequences, each corresponding to a k-best list (each of which corresponds to a sentence $s \in S$). Each sequence in W consists of a number of pairs, consisting of a parse tree t and an associated weight $w(t)$. The weights are calculated as follows.

Let $K = \langle t_1, \ldots, t_k \rangle$ be the ordered sequence of parse trees output by the parser for a particular sentence and for each t_i, let $p(t_i)$ be its probability. Let b be the amount of boost mass we wish to assign. Equation 4.1 shows how the weight $w(t)$ is calculated for each tree t on the k-list.

$$w(t_i) = \begin{cases} \frac{p(t_i)\cdot(1-b)}{\sum_{j=1}^{k} p(t_j)} + b & \text{if } i = 1 \\ \frac{p(t_i)\cdot(1-b)}{\sum_{j=1}^{k} p(t_j)} & \text{else} \end{cases} \tag{4.1}$$

Note that if $b \in [0, 1]$, then the set of weights of trees belonging to a single

k-list actually form a proper probability distribution, since $\forall i(w(t_i) \in [0,1])$ and $\sum_{i=1}^{k} w(t_i) = 1$.

As a sidenote, this is reminiscent of the works of Simon Kirby et al. [34] on iterated Bayesian learning. Stated in the terms of that framework, the weighted treebank forms a posterior distribution over parse trees conditioned on the set of sentences. The next iteration's grammar is learned from the entire distribution. If this distribution is unboosted, i.e. $b = 0.0$, then the effect is the same as if the grammar is obtained by sampling trees from an infinitely large treebank, with the relative frequencies of the trees equal to their weights. Another possibility for obtaining the input for the next iteration is by taking the maximum a posteriori probability (MAP), instead of a sample. Increasing the boost mass has the effect of moving from the sampling to the maximizing strategy.

4.1.2 Measuring reanalysis

Chapter 2 discusses a definition of reanalysis by Harris and Campbell that will serve as the starting point for the measurements in our simulation.

> Reanalysis ... is a mechanism which changes the underlying structure of a syntactic pattern and which does not involve any immediate or intrinsic modification of its surface manifestations. [27, p. 61]

Since the overt form of the syntactic pattern is not modified by reanalysis until analogical levelling sets in, it is essentially unobservable in historical language data. However, the simulation setup described above has the advantage that for each generation we have access to both the surface manifestations of patterns and their analyses.

This means we can study reanalysis as a *process*. In any linguistic framework, alternative analyses of a form are commonly licensed by a grammar. In a probabilistic framework, however, we can quantify their position relative to the salient analysis. We first redefine reanalysis to fit our simulations.

Definition 6 (Reanalysis)**.** *A syntactic pattern is reanalyzed if the 1st-best structural analysis of the pattern (i.e. its parse tree) differs from the 1st-best analysis of the previous iteration in the simulation.*

The fact that the simulation involves weighted treebanks as a means of transmission means that we can track the process by which such a change takes place. We will focus on two aspects. Firstly, we will look for situations where a

new parse tree occupies the 1st position on the k-list and secondly, we will look for situations where non-optimal analyses lose probability mass and disappear as contenders. The first aspect signifies reanalysis, while the latter means that non-optimal analyses are no longer available for analogical levelling.

There are two basic forms in which the reanalysis of a syntactic pattern may appear. The first, *rebracketing*, will refer to a change in the unlabelled structure of the 1st-best parse tree of a pattern. The second, *relabelling*, will refer to a change in the labelling of nodes in the 1st-best parse tree. Actual reanalysis is likely to involve combinations of the two. For the simulations in this section however, we isolate these two methods and study them one by one.

4.1.3 Simulation 1: reanalysis as rebracketing

The first simulation in this section focusses on the rebracketing of a syntactic structure under the pressure of similar structures in the grammar.

Linguistic motivation

The linguistic example we use comes originally from Jespersen. It concerns a hypothetical example that is nevertheless used by David Lightfoot to illustrate a part of his theory of syntactic change. It also fits our purposes in that it illustrates how rebracketing of syntactic structures may take place in our model of language transmission.

The English verb *like*, already existant in Old English, has changed its argument structure through the period of Middle English. The following four instances illustrate the stages of the change[1]

1. þam cynge [DAT] licodon [PL] peran [NOM PL] (OE)
2. the king liceden [PL] peares (Early ME)
3. the king liked pears (Late ME)
4. he liked pears (Early ModE)

In the first stage, the number-agreement and case-marking unambiguously identify *peran* as the subject. At this stage, the verb *like* has a similar argument structure to French *plaisir* or Dutch *bevallen*. In the second stage, case marking has disappeared and only number-agreement helps identify *peares* as the subject.

[1] Discussion adapted from [19]

By the late Middle English period, the construction has become ambiguous and both *king* and *pears* can be analyzed as the subject. In the fourth stage, what was first a dative argument is always interpreted as subject and the sentence is again unambiguous.

The etymology of *like* has been used by David Lightfoot [40] in his argument for the role of the Trace Erasure Principle in his theory of syntactic change. The Trace Erasure Principle (TEP), from an older version of generative theory (cf. [45]), prohibits a moved constituent from filling a position which is already occupied by a trace. According to Lightfoot, this principle triggers the reanalysis pattern 3 in the late Middle English period. By this period, the basic word order of English is SVO, so the indirect object *the king* and the subject *pears* would have to switch positions around the verb from their base position. This is impossible according to the TEP, since these positions are already filled by the traces of the moved constituents. Therefore, *the king* is reanalysed as the subject of the sentence.

This analysis of the syntactic change is not without its critics (e.g. see [19, p.23-23] for a discussion). Below we will show that a simpler approach is possible, where the reanalysis is carried out on the basis of analogical pressure from other transitive verbal constructions and canonical sentence patterns.

The simulation

The simulation in this section models the transition from the third to the fourth stage. The seed treebank contains 100 three-word transitive sentences, like *king likes pears*, *mary loves john* or *queen eats apples*. All but the sentences with *like* as the verb are given a tree structure corresponding to an SVO-analysis. Sentences with *like* as the verb have an OVS-analysis. The treebank is automatically generated, with a set of 100 nouns distributed randomly over the argument positions in the sentences. Figure 4.2 shows parse trees for the SVO and OVS analyses.

The sentences with *like* make up 1/3 of the treebank. This distribution is rather generous to the OVS-analysis, but it simplifies our exposition. If reanalysis and levelling of *like*-sentences to SVO takes place under these conditions, the model predicts they will also take place under circumstances where *like*-sentences form a much smaller minority in the original treebank.

Note that the *like*-sentences are likely to be ambiguous from the very first grammar extracted from the seed treebank. We can see that this is the case

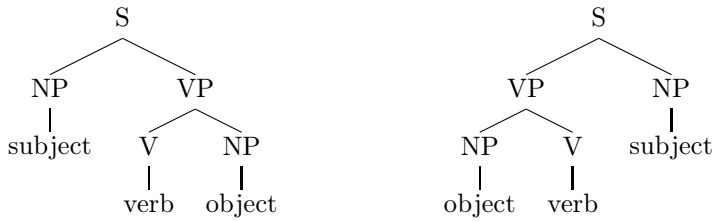


Figure 4.2: SVO and OVS parse trees

if we consider the nouns that occur in them. These are likely to also occur in sentences with an SVO-analysis, therefore an SVO-analysis for *like*-sentences is also likely to be licensed by the first extracted grammar.

The simulation tracks the relative mass of the OVS-analysis of the argument structure of *like*. The starting point will lie around 1/3 plus any boost mass we wish to assign. Under the DOP1 and Equal Weights estimators we expect to see a rapid decline in this mass from its starting point. The case is different for the Bonnema estimator and the shortest derivation estimator. Consider first the shortest derivation estimator.

The shortest derivations for a sentence patterned as *noun1 likes noun2* are shown in figure 4.3. It shows that both the SVO and the OVS analysis will require on average three subtrees if the nouns are distributed randomly. Recall that we do not use a tie-breaker in case multiple winning derivations. This means that we can expect the relative mass of the OVS analysis to settle around 0.5 under repeated application of the shortest derivation estimator.

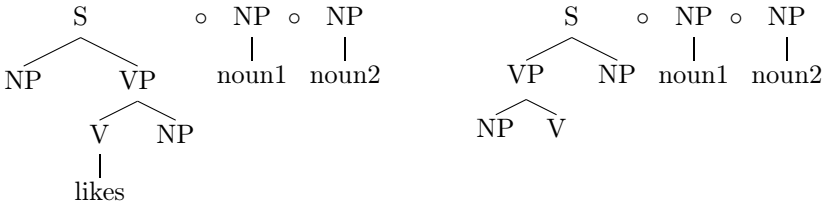


Figure 4.3: Shortest derivations of *noun1 likes noun2*

The results for the Bonnema estimator are expected to closely resemble those of the shortest derivation estimator due to the artificial nature of our seed

treebank. Recall that the probabilities of external rules in the PCFG-reduction of the Bonnema estimator rely only on one parameter: the number of times the root non-terminal occurs in the training data. Since the SVO and OVS parse trees have the same non-terminal node labels and only differ in the structure of the trees (see figure 4.2), the Bonnema estimator is expected to show the same behaviour as the shortest derivation estimator.

Results

Figure 4.4 shows the basic results for the simulation. It shows the average results of 100 simulation runs with $b = 0.0$. The relative mass of the OVS analysis under the DOP1 and Equal Weights estimators reduces to 0 after about 40 iterations. The relative mass under the shortest derivation and Bonnema estimators rapidly settles to 0.5.

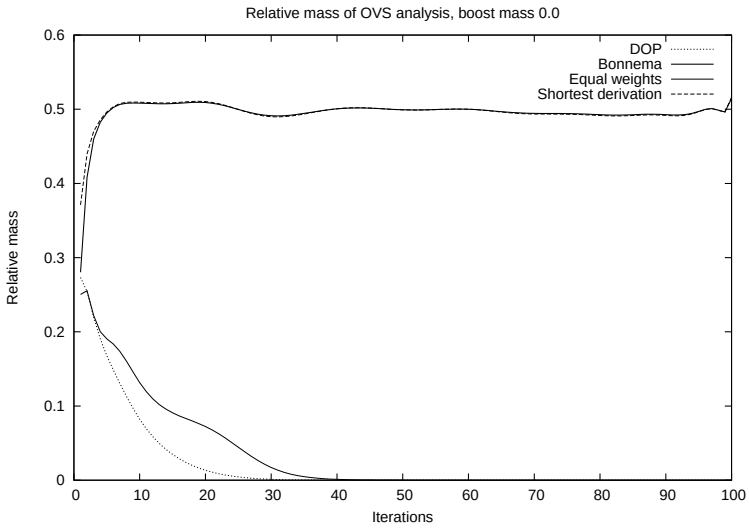


Figure 4.4: Relative mass of the OVS analysis for $b = 0.0$, averaged over 100 simulation runs

What does this show linguistically? Under the DOP1 and Equal Weights estimators, the reanalysis of sentences *Noun1 likes noun2* as SVO, inherent to the grammar, is completed fairly rapidly. That is, after about 40 iterations, the only analysis is SVO and the verb *like* has been pulled level to the other verbs in the original treebank. In other words, we have here an indication that a simple model based on analogy can provide an alternative explanation for the

change discussed by Lightfoot. The trigger for the reanalysis here is not an abstract principle of the grammar, but the pressure from analogy with other verbal constructions in the language. The verb *like* is levelled to the rest of the language data in conforming to a simple SVO pattern.

However, not all single runs of the simulation conform to this outcome. Since the treebank is relatively small and the set of sentences that is parsed at each iteration is generated randomly under a set of lexical constraints, we can expect there to be runs which show different results. If the amount of boost mass, the extra probability mass reserved for the 1st-best parse tree, is increased, the outcomes of the runs start to diverge from that shown in figure 4.4. This holds only for the Equal Weights and DOP1 estimators. The higher the boost mass, the more likely it is that *like* will hold on to its argument structure and retains its deviant analysis in the grammar.

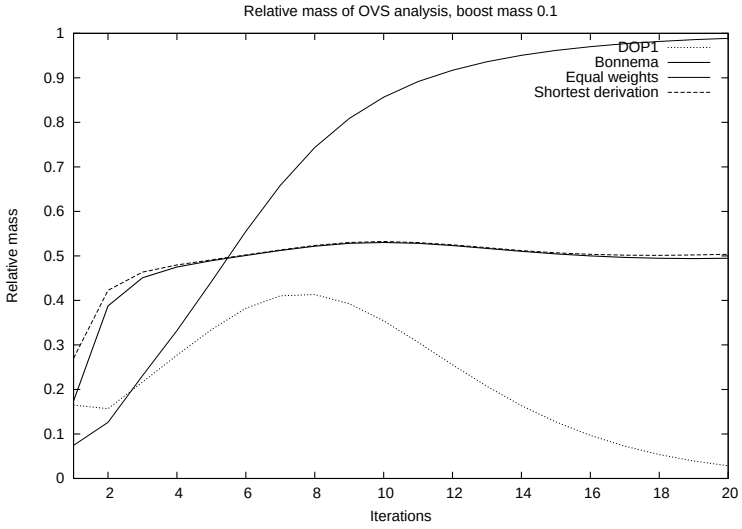


Figure 4.5: A smoothed view of a deviant run for the Equal Weights estimator for $b = 0.1$

Figure 4.5 provides a (smoothed) view of a part of a run in which the relative mass of the OVS analysis under the Equal Weights estimator actually showed the opposite tendency from that in figure 4.4, it approached 1. Note that in the same run, the DOP1 estimator also showed deviant behaviour, initially following the curve of the Equal Weights estimator, but eventually settling down to 0. These runs are relatively rare for $b < 0.25$ but become more frequent for $0.25 < b < 0.4$.

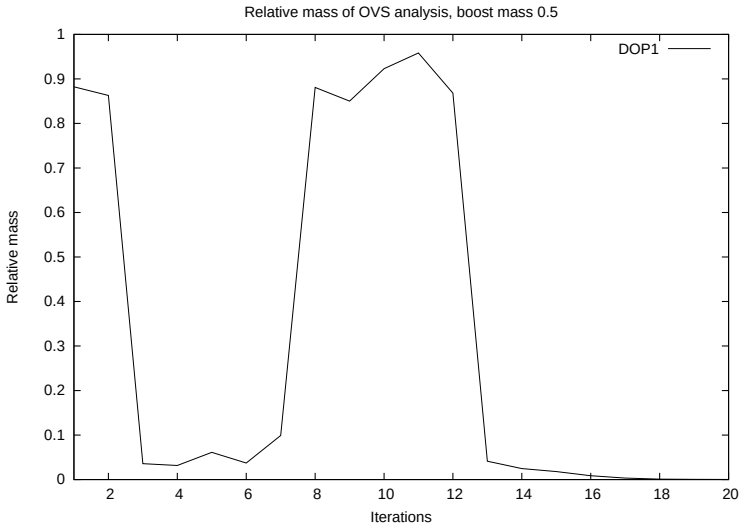


Figure 4.6: An unsmoothed view of a run for the DOP1 estimator for $b = 0.5$, showing oscillations in the stable states.

If b is increased to even higher levels ($b > 0.4$), both the EW and the DOP1 estimators provided even more interesting behaviour. Figure 4.6 shows the results of the DOP1 estimator for a run at $b = 0.5$. In this run, a multitude of reanalyses occurred in sequence. Over several hundred iterations the run exhibited distinctive oscillating behaviour. Semi-stable states where one analysis was dominant were followed by rapid changes to the opposite situation.

These latter results point to the importance of the boost mass parameter. Recall that it was introduced as a way to account for the linguistic and cognitive importance of the first item on the k-list of the parser. Increasing the value of b also corresponds to a move towards the maximizing strategy in the terms of the Bayesian learning framework. As we have seen, the amount of boost mass turns out to radically influence the results of the simulations, producing much wider variance in the possible outcomes of the runs. This may have linguistic implications, depending on how we interpret the model.

If we interpret the simulation to represent a single learner, then the effects are an artifact of its discontinuous learning strategy. On the other hand, if we interpret the grammar to represent a language *community*'s homogeneous

linguistic system, then we may tentatively make some conclusions from the effects.

Firstly, it may show that analogical levelling is not necessarily a deterministic process. Minor fluctuations in the initial conditions and the circumstances of transmission may produce hugely varying outcomes. While the isolated and artificial nature of our simulations is to a large degree responsible for this state of affairs, it does ties in nicely with an ongoing debate in the literature on grammaticalization theory about the dependence of the manifestation of clines on (from the viewpoint of some theorists) seemingly unpredictable events (cf. the discussion in [19, p.115–124]).

Secondly, the run shown in figure 4.6 may have implications for the debate on the *unidirectionality* of grammaticalization processes. According to some researchers (exemplified by [28], but see pages 99–139 for a critical discussion), grammaticalization is essentially a one-way process. While a form undergoing grammaticalization may or may not reach the end state, these researchers see the process as irreversible. Part of this irreversibility is the assumed impossibility of returning to a previous analysis of a syntactic pattern once reanalysis and levelling have taken place. Other literature provides counter-examples and call this hypothesis into question on theoretical terms (cf. [18]). The existence of runs figure 4.6 seems to provide support for the latter position.

4.1.4 Simulation 2: the spread of relabelling

In this simulation we isolate a type of reanalysis involving only the relabelling of node labels in a phrase structure tree. The unlabelled tree is not changed over the course of the iterations.

Linguistic motivation

Our linguistic cue for this simulation will be the change in the 'be going to' construction in English.[2] Originally 'going to' had a purposive, directional meaning, as in *She is going to London to marry Bill*. Following the futuritive interpretation associated with this directional reading, a reanalysis occurred in which 'be going to' as a whole came to be interpreted as an auxiliary. This reanalysis opened up the way for phenomena typical for auxiliaries, such as a phonological reduction into *gonna*, since the phrasal boundary between *going* and *to* has disappeared.

[2] Although many discussions of this change exist, we follow the discussion in [28].

In the last situation, a sentence like *She is going to marry Bill* is syntactically ambiguous. Figure 4.7 shows two of the parse trees for this sentence. Note that while these analyses have the same overt syntactic pattern, the difference in the labels implies a difference in semantic interpretation and may denote a difference in phonology, due to the aforementioned reduction.

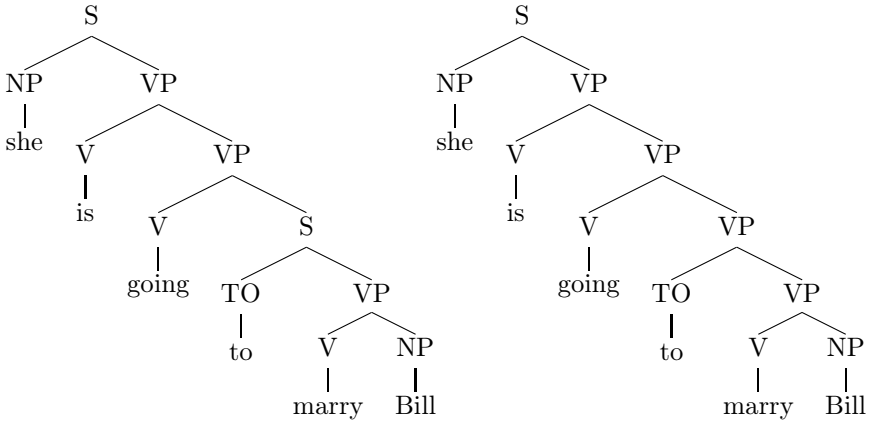


Figure 4.7: Two parse trees for *She is going to marry Bill*.

The reanalysis of *be going to* as an auxiliary is often attributed to the analogy in meaning between the purposive aspect of the directional reading and the intentional reading of modal auxiliaries [28]. We do not model this semantic analogy in the simulation in this section. We will show, however, that once the reanalysis has become viable for some verbs, structural analogical pressure can help spread the innovation across all verbal domains, on purely syntactic grounds.

The simulation

The simulation models the spread of the rightmost tree in figure 4.7 across verbal domains. The simulation starts from a seed treebank containing 100 sentences and their associated tree structures. Both trees labelled as the leftmost trees in figure 4.7 and trees labelled like the right one are present, in proportion expressed by parameter r, given by equation 4.2

$$r = \frac{\#\text{ trees with purposive reading}}{\#\text{ trees with purposive reading} + \#\text{ trees with auxiliary reading}} \tag{4.2}$$

The trees with the purposive, non-auxiliary reading of *be going to* are restricted to a small, randomly generated set of content verbs (occupying the position of *marry* in the example trees). The noun positions are also filled randomly from a set of 100 nouns. At each iteration, the nouns and verbs are redistributed over the patterns in accordance with the lexical constraints.

The simulation tracks the relative mass of the auxiliary reading under different ratios in the seed bank and amounts of boost mass b. We expect similar behaviour of the estimators as in the previous simulation, depending on the ratio of the starting seed bank. That is, we expect the relative mass of the auxiliary analysis under the Equal Weights and DOP1 estimators to converge on 0 for high values of r and on 1 for low values of r. And we expect the relative mass under the shortest derivation and Bonnema estimators to settle on 0.5 modulo the boost mass. The interesting questions are how the convergences turn out for $r = 0.5$ and for higher values of b.

Results

Figures 4.10 and 4.11 show the averaged results of 10 simulation runs for $b = 0.0$ and for different settings of r. The figures show that the shortest derivation and Bonnema estimators behave as predicted from the results of the previous simulation.

The behaviour is different for $r = 0.5$. The shortest derivation and Bonnema estimators still behave the same way, but the DOP1 and Equal Weights estimators can now go either way. Convergence is slower, but both will eventually settle on either 0 or 1. On average, over 100 runs, both end states turned out to be equally likely for these estimators. Figures 4.8 and 4.9 show examples of single runs for this setting. Note the difference in behaviour for the DOP1 and Equal Weights estimators from that in figures 4.10 and 4.11.

Increasing the value of b has less effect than it had in the previous simulation. The only noticeable difference between $b > 0.0$ and $b = 0.0$ turned out to be the speed of convergence. Higher levels produced significantly faster convergence, but no oscillations were observed over 100 runs for each setting.

What these simulations show is that the frequency of occurence is important for the outcome of a reanalysis/levelling process. If the reanalysed, innovative subsurface structure is not able to cross a certain threshold, analogical levelling will pull the innovation into step with the rest of the language. Note that in this simulation, we have excluded semantic and other factors which may have

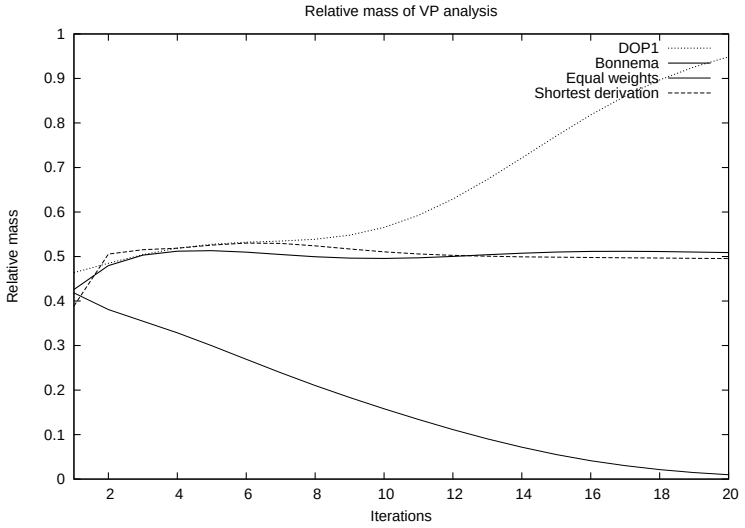


Figure 4.8: Average relative mass of single run, for $b = 0.0$, $r = 0.5$.

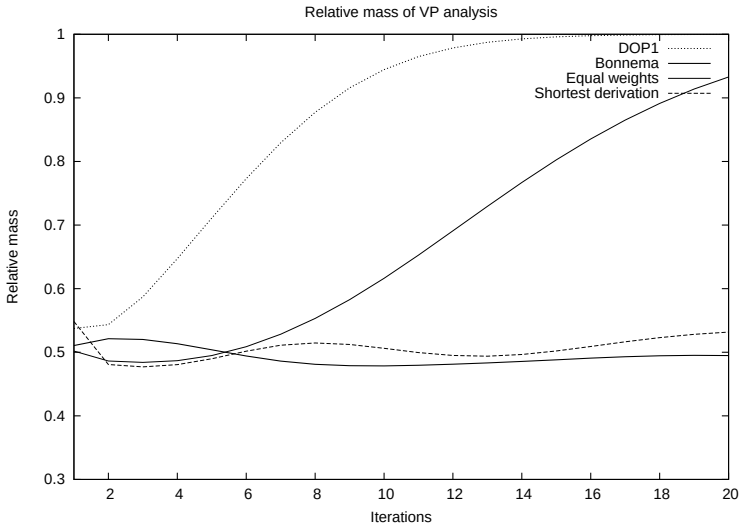


Figure 4.9: Average relative mass of single run, for $b = 0.0$, $r = 0.5$.

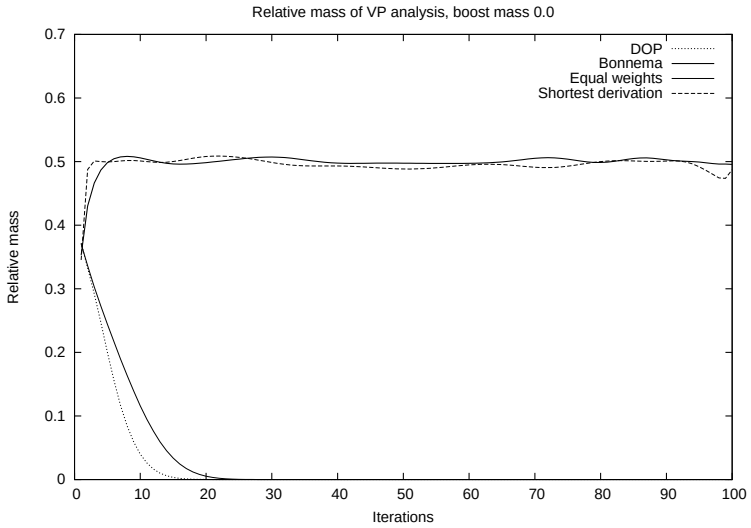


Figure 4.10: Average relative mass of auxiliary analysis over 10 runs, for $b = 0.0$, $r = 2/3$.

Figure 4.10 shows that the relative mass of the auxiliary analysis under the DOP1 and Equal Weights estimators converge to 0 fairly rapidly, as predicted, although the Equal Weights estimator takes slightly longer to get there, unlike the last simulation. Figure 4.11 shows the expected opposite behaviour.
helped the innovation to cross the threshold. We do not consider this a model of the development of the *be going to* construction per se, but as a simplified example of how levelling can spread an innovation across domains.

We also saw how the estimators divided into two groups in terms of their behaviour. The DOP1 and the Equal Weights estimators on the one hand and the Bonnema and shortest derivation estimators on the other. The first group exhibited the most interesting behaviour for our purposes. Since these two estimators essentially behaved in the same way, all results from here on only discuss the DOP1 estimator.

In our introduction we mentioned that eventually, quantitative models of language change might be used to evaluate the validity of language formalisms. While we do not discuss results for the shortest derivation and Bonnema estimators from here on, we feel it is premature to reject their empirical validity on the grounds of these results. The grammars we used in these simulations are of a highly artificial nature, being restricted to small and uniform subsets of

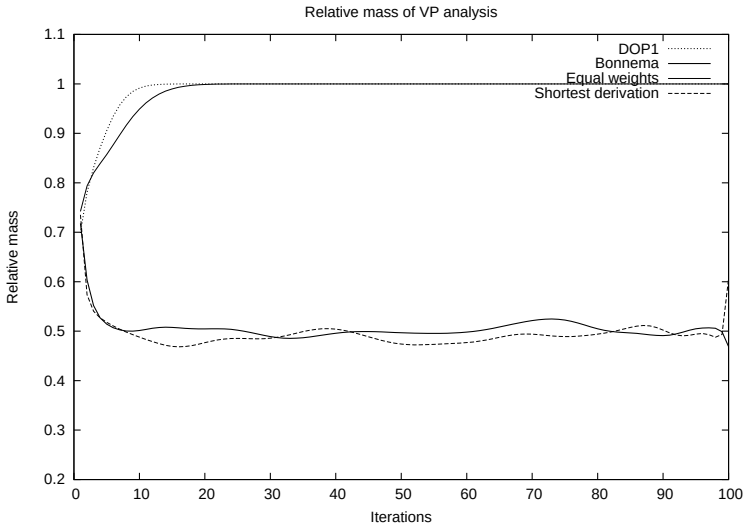


Figure 4.11: Average relative mass of auxiliary analysis over 10 runs, for $b = 0.0$, $r = 1/3$.

natural language. An evaluation of the validity of these estimators is therefore unwarranted at this stage.

4.2 Adding functional pressure

This section adds functional pressure to the simulation. We use the simulation to model the relationship between case marking and word order freeness.

4.2.1 Linguistic motivation

The historical example we study in detail in this section is the disappearance of overt case markers on nominal constituents in Dutch and English. Both Old English and Middle Dutch had a richer case system than their modern day counterparts (cf. [50] for OE and [52] for MD). Both Old English and Middle Dutch are commonly considered to be languages with an basic OV word order (cf. [50] for OE again and [51] for MD). The following examples illustrate this:[3]

(4.3) *gif hie him þæs rices uþon*
if they him the kingdom granted

(4.4) *dat ic dis macht hebbe*
that I this power have

However, these languages also had a relatively free word order system, regularly allowing extraposition of arguments of the following type (for the case for the existence of these extraposition phenomena in Old English see [53], for the case of Middle Dutch [23]).

(4.5) *þæt hit sie feaxede steorra*
that it is long-haired star

(4.6) *dat si ontmoetten ene ioncfrouwe*
that they met a lady

In Modern Dutch and Modern English from the Early Modern period onward, the extraposed structures are no longer grammatically correct sentences. While Modern Dutch settled on an OV word order, Modern English now is a VO language.[4] Both languages have also lost most of their overt case markers.

In the literature this relation is analysed and discussed in various ways. The following is exemplary of the generative approach (based on [54]).

Nominal projections need to be assigned case and verbs govern in a single direction, determined by the basic word order of the language, i.e. VO or OV. A

[3] These example sentences, as well as the others in this section are taken from [54]
[4] Section 4.3 below discusses the change of English from OV to VO in detail.

morphological case system, featuring overt case marking on nominal projections, allows complements of the verb to escape the direction in which the verb assigns case. If the morphological case system disappears, NP's no longer have a way to escape structural case and thus the word order of a language is frozen. Therefore regular extraposition constructions like in sentences (3) and (4) above are no longer allowed by the grammar.

In this model we propose a related, but more functionally-oriented view of the same phenomenon. The simulation introduced in the next section will show that it can account for the type of changes that English and Dutch have undergone. It elaborates on the simple model of the previous section by adding representations of two aspects of language: phonology and semantics. As discussed in chapter 2, phonological erosion is an important subject in grammaticalization studies. It denotes the gradual weakening of stress on morphemes, until they eventually disappear. In germanic languages the end of a word is the place where morphosyntactic features such as case are marked. Erosion of the word ending interferes with the case system.

The simulation in this section and that in the next also incorporate a semantic component. In this section we use it to model a basic prerequisite of language use: mutual understandability. Markers of syntactic case and by extension, semantic role, help the interpretation of a sentence. If this system erodes, other clues such as word order may step in to help semantic interpretation.

4.2.2 The simulation

The simulation consists in essence of a simple Generator/Interpreter module as shown in figure 4.12. The module takes as its input a semantic structure. This structure is processed by the Generator submodule, backed by a grammar. The generator returns a sentence expressing its input. This sentence is subsequently taken as input for the Interpreter submodule which outputs a reconstructed semantic structure.

This Generator/Interpreter module is integrated in a larger supermodule, shown in figure 4.13, which does two things. Firstly it generates the semantic input for the Generator/Interpreter module and secondly it checks whether the semantic output is reconstructed correctly, by testing for equality according to definition 4. The outcome of this test influences the decision to adjust the grammar.

So, put simply, we model an agent who randomly generates meanings, trans-

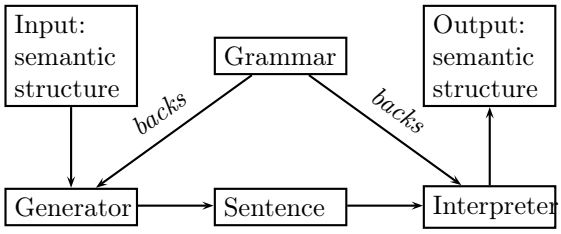


Figure 4.12: Generator/Interpreter

forms those meanings into sentences and checks whether these sentences adequately reflect the meaning, adjusting its internal grammar to optimize its language output. Before we go into the details of the adjustment, we will first discuss the Generator/Interpreter module in more depth.

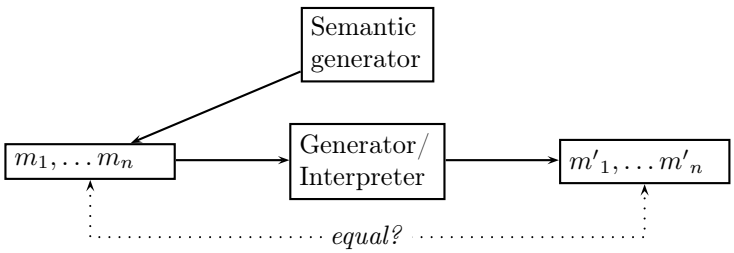


Figure 4.13: Simulation overview

The Generator/Interpreter module

The Generator submodule takes as its input a semantic structure. In chapter 3 we discussed the representation of these structures, namely dependency structures. We stressed that the conversion to syntactic structures was one-to-many. For this simulation, each semantic structure generates not only the full set of

syntactic structures as described there, but also a copy of each tree in that set, with case markings on the nouns and annotations that reflect these marking on the nominal projections.

The resultant set is ranked by probability by the grammar that backs the module. Using these probabilities as weights by normalizing them by their sum, a probability distribution is formed over the phrase structure trees that correspond to the semantic structure. From this distribution a single tree is sampled. Its yield is the output of the Generator.

The Interpreter submodule takes the opposite direction. The input sentence is parsed by the hypergraph parser described in section 3.2. The best derivation is then converted into a semantic structure according to the conversion algorithm described in section 3.3. This semantic structure is the output of the Interpreter module.

As an example of the functioning of the Generator/Interpreter module, consider the following. We start with the semantic structure `loves(john, mary)`. From this semantic structure, the entire set of corresponding phrase structure trees is generated, as shown in figure 3.8. The grammar ranks these trees by their probability and a single tree is sampled. Its yield, say it is *mary john loves*, then forms the input for the Interpreter. The Interpreter attempts to parse the sentence, if need be by using the Flip-Reverse algorithm, and returns the best derivation of the sentence. This derivation is then converted to a semantic structure, let's say that is `loves(mary, john)`. This semantic structure is then tested for equality with the original one. In this case the test fails. Depending on the outcome of the test, the grammar is adjusted.

4.2.3 Obtaining and adjusting the grammar

The simulation starts from an artificially generated treebank consisting of simple three word sentences with transitive verbs. Nominal constituents are case-marked. All possible word orders are represented in the treebank. We will study two different initial settings for this simulation. One in which all word orders appear an equal amount of times in the seed treebank and one in which there is a bias in the distribution for the SOV word order.

From the initial treebank a DOP1-grammar is extracted, with the following adjustment. For each preterminal rule with a noun as the terminal, two sets of subtrees are used in the grammar, one with morphological case and one without. The relative frequency of these sets to each other is parametrized by variable e,

for erosion.

The probability for preterminal to terminal rules that ends up in the grammar, $p'(preterminal)$ is determined by the relative frequency, $p(preterminal)$ and e, as shown in equation 4.7.

$$p'(preterminal) = \begin{cases} p(preterminal) \cdot e & \text{if case marked} \\ p(preterminal) \cdot (1 - e) & \text{else} \end{cases} \tag{4.7}$$

This equation helps incorporate the phonological pressure in the simulation. By rescaling the subtrees for nominal constituents, we build a bias into the grammar so that non case marked yields will be preferred over case marked yields in the ranking step of the Generator submodule.

As described above, the output of the Generator/Interpreter module is compared to its input. Based on this comparison, the weights of the subtrees that were used in the winning derivation of the parser are adjusted. The resultant weight of a subtree t, $w'(t)$ depends on the original weight $w(t)$ and a scaling factor a, as defined in equation 4.8.

$$w'(t) = \begin{cases} w(t) \cdot (1 + a) & \text{if equal} \\ w(t) \cdot (1 - a) & \text{else} \end{cases} \tag{4.8}$$

The probability mass of other subtrees headed by the same node as t is then recalculated so that all the subtrees headed by that node again form a probability distribution.

The result of this adjustment is that subtrees which were part of a process in the Generator/Interpreter module that resulted in an adequate sentence (i.e. one from which the original semantic structure could be reconstructed) are scaled up, while others are scaled down. The grammar tries to optimize itself for communicative purposes, while balancing this goal with economy of effort, as expressed by the preference for non-case-marked nominal constituents. This process is repeated until the equality test succeeds for 50 sentences in a row.

4.2.4 Results

When the simulation starts from a uniform distribution over the possible word orders, four distinct stable states emerge, depending on the value of e. In practice, the value of a only mattered for the speed of the simulation, not for the eventual results. All simulations discussed below ran under $a = 0.4$.

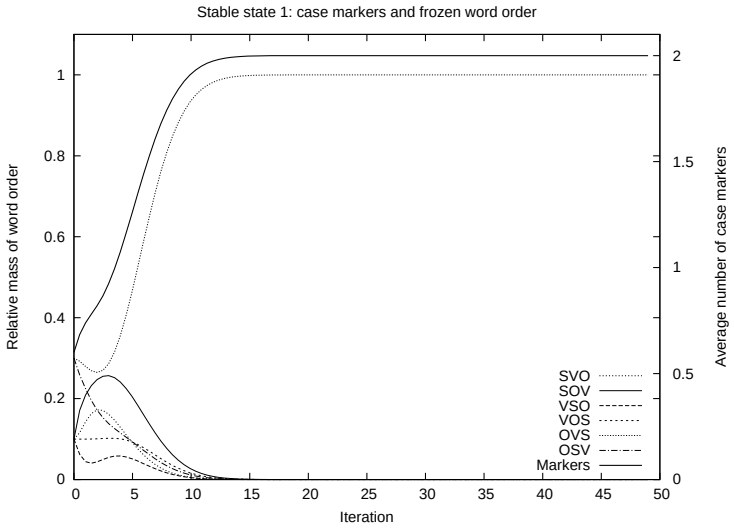


Figure 4.14: Stable state 1

Figure 4.14 shows the first stable state. Low values for e, around 0.5, mean that the penalty for using case marked nouns is low or non-existent. Under these conditions, the grammar quickly stabilizes to a state where all nouns are case marked and the word order remains completely free. Figure 4.14 shows the graph for this simulation.

For higher values of e, different end states are found. Figure 4.15 shows the graph for $e = 0.6$. Under this condition, the penalty for using case marked nouns is still light enough to allow for the eventual marking of all nouns, but by the time this state is reached, the word order has also become frozen.

Figure 4.16 shows yet again a different final state. Half the nouns are case marked and the word order is mostly frozen, with SVO being used predominantly, but OSV also possible. The marking of only half the nouns in this simulation is enough to unambiguously reconstruct the semantic structure from the syntactic structure, since all sentences feature exactly two nouns and these occur only in the *agens* and *patiens* roles.

Figure 4.17 finally shows the last stable state, for high values of e. Here the penalty for using case markers has become too great to overcome. The end state is reached very slowly in this run and contains predominantly VOS, but

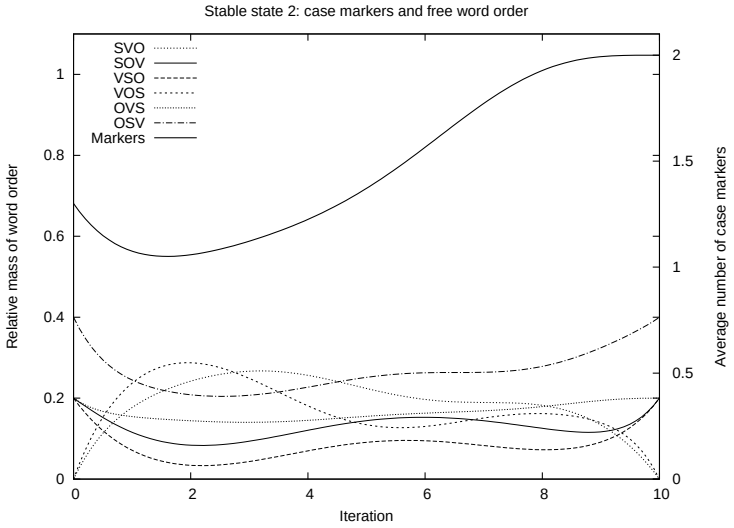


Figure 4.15: Stable state 2

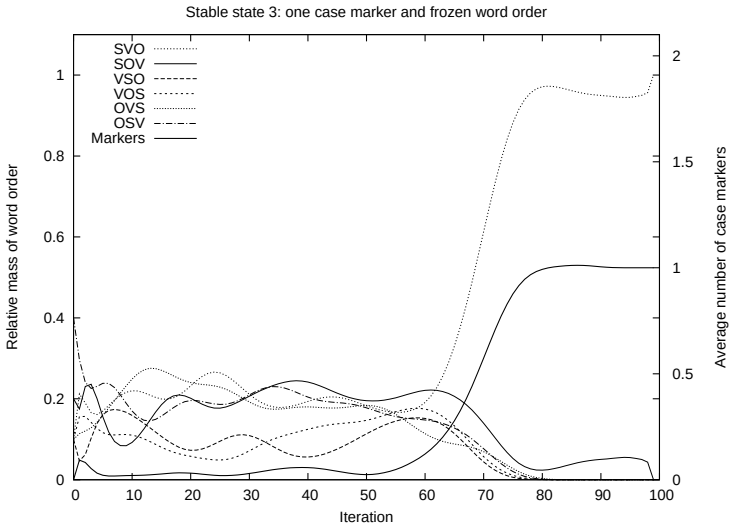


Figure 4.16: Stable state 3

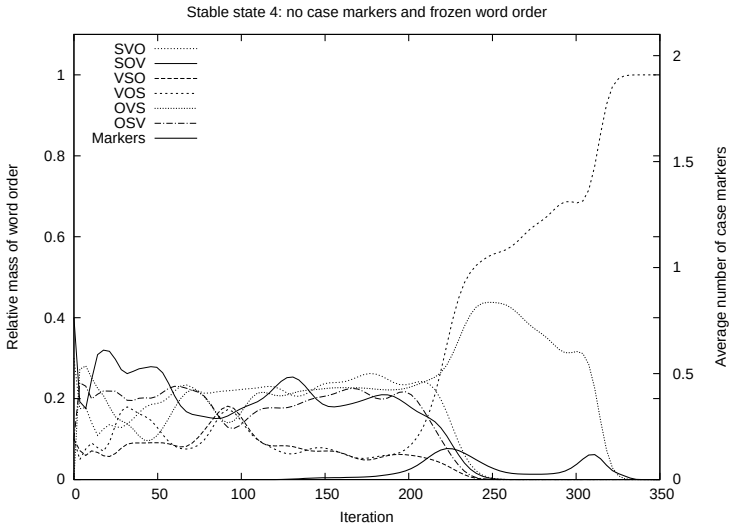


Figure 4.17: Stable state 4

also some OVS word order. Note that the specific word order reached is largely dependent on chance, other runs under the same parameters reached different end states, but showed the same pattern of slow convergence to a mostly frozen word order.

If the simulation starts from a distribution over word orders that is biased towards SOV, with a relative frequency of 0.4 appointed to this word order in the seed treebank, only the stable states with a frozen word order were encountered in the simulation runs. The convergence to SOV was also quicker than before. Figure 4.18 shows an example run, with a stable state with a frozen word order and an average of one case marker.

The results from this section show that the extension of the model with semantics can produce linguistically plausible language states. The next section adds a hetergeneous language community to the model and finally implements the full model of language transmission outlined in chapter 1 in an attempt to simulate word order changes under influence of language contact.

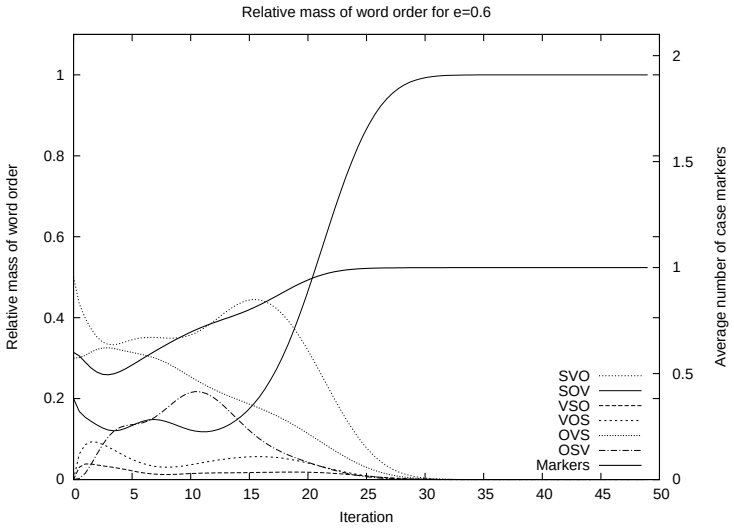


Figure 4.18: Example stable state for biased initial treebank

4.3 Adding a heterogeneous language community and communal feedback

The last simulation in this study implements the full model of language transmission, shown again here as figure 4.19. This model incorporates all of the elements that we used in the previous simulations and add some new features.

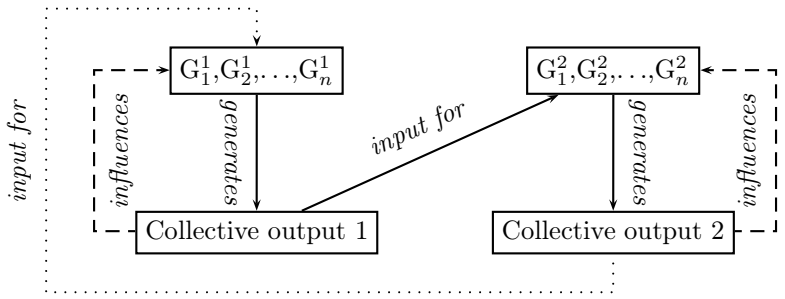


Figure 4.19: The final model of transmission introduced in chapter 2

The model in this section adds a heterogeneous language community to the simulation. In this community, agents communicate and attempt to adjust their

language output to that of their peers. Different agents do this in different ways, depending on their linguistic background and their stage in life.

4.3.1 Linguistic motivation

The historical example we take as a starting point for this section is the change in basic word order that took place in Early Middle English. As mentioned above, Old English is often analyzed as an OV language. However, in matrix clauses, the verb often occupied the second constituent position in the sentence, a phenomenon that can also be observed in modern Dutch and German, called Verb Second (V2). Between 1150 and 1350 A.D. this central feature of the English grammar changed and the basic word order became VO.

Historical linguists have pointed to the invasion of the Vikings as a cause for this shift in word order (cf. [42, 50]). In this simulation, we base ourselves on a hypothesis by Fred Weerman [54]. Weerman also identifies the invasion of the Vikings as the main cause for the change. This mass invasion brought a huge amount of second language learners to Northern England. According to his hypothesis, the fact that they were second language learners acquiring an OV-V2 language was crucial in the change.

Adult second language acquisition (L2-acquisition) differs from child first language acquisition in some important ways. Second language learners exhibit specific patterns in learning the word order of a new language. Clahsen and Muysken [17] show that L2 learners acquiring an OV-V2 langage tend to over-generalize the word order of the matrix clause to non-matrix clauses. Since the canonical matrix clause in an OV-V2 language is SVO, these learners infer that the language they are learning has a VO word order. Furthermore, this pattern occurs irrespective of the native language of the L2 learner. To clarify, from the observation that

(4.9) John loves Mary.

is a grammatical sentence. A second language learner could infer that

(4.10) Carl knows that John loves Mary.

is also grammatical. Even though the correct construction under an OV-V2 grammar would be:

(4.11) Carl knows that John Mary loves.

As discussed in the previous section, the English language of the time when the Vikings arrived had a relatively free word order. It allowed regular extrapositions of nominal constituents in both non-matrix and matrix clauses. These extraposed constructions do not conform to the basic word order pattern.

This relative freeness meant that adult native speakers of English could parse grammatically incorrect sentences like (9) with relative ease. Weerman goes one step further and posits that the frequency with which these extraposed constructions occured in the native English speakers' output could increase under the influence of the presence of second language learners.

Weerman analyses this frequency increase in the following way. Since (under the generativist assumption) the adult native speakers' language parameters have been fully set once the critical period of language acquisition is over, the extraposed constructions are added to the grammar as 'peripheral rules'. While they may not be part of the 'core' grammar, the peripheral constructions may still appear in the native speakers' language output.

From the viewpoint of child L1 learners growing up in this linguistically diffused environment, the positive evidence in the primary linguistic data for the OV-V2 word order is partly destroyed. In generative words, once the number of extraposed constructions reaches a certain critical frequency, the child may well analyze the PLD as exhibiting an SVO word order and set its parameters accordingly. When this happens, a syntactic change has taken place.

For the simulation in this section we adopt the basic idea underlying the described mechanism of the word order change, namely the interplay between L1 and L2 acquisition. We do not adopt the generativist language model, however, but make an attempt to fit the different mechanisms underlying L1 and L2 acquisition in our probabilistic data-oriented language model.

4.3.2 The simulation

When attempting to make a formal model of a language community that existed almost 1000 years ago, we come across some problems caused by a lack of evidence. How many Vikings invaded exactly? How was their linguistic interaction with the native British structured? Were the communities isolated or did they blend in well with each other? If so, to what degree? What percentage of constructions in English exhibited these extraposed nominals? How was the language of the Vikings structured?

For all these question, no exact answers are known. What we will do in

this section is propose a way to model the situation as a whole and make broad guesses toward the actual values of the parameters involved. These guesses may very well be wrong. This is not particularly important. We show that under our assumptions, plausible outcomes of the simulation may be obtained. If new data becomes available, the model may be adjusted and studied anew. What we attempt to do is show how the situation may be modelled *in principle*.

We model the linguistic community as a set of 25 agents interacting with each other. The agents are placed on a toroidal grid and each possesses a Generator and an Interpreter, as described in the previous section, backed by an internal grammar. The simulation basically consists of the same process that was used in the previous one, with a few minor changes. At initialization, each agent extracts a grammar from a previously defined treebank, containing 100 sentences along with their parse tree.

At each step of the simulation, each agent is assigned a set of 20 randomly generated semantic structures. Each agent generates 20 sentences from these structures, according to the process discussed in the previous section. All semantic structures follow the same pattern, consisting of a main predicate and an embedded predicate, e.g. `thinks(carl, loves(john, mary))`. The syntactic structures corresponding to these semantic structures therefore always contain a matrix clause and a subordinated non-matrix clause. This ensures that for OV-V2 grammars, both main word order patterns are represented and evidence for the OV nature of the language is present.

Each agent then receives the sentences from each of its neighbours on the grid (8 in total) and uses its Interpreter to attempt to reconstruct the semantic structure. If this succeeds (i.e. the reconstructed semantic structure equals the original), the agent's grammar is adjusted. Instead of the update mechanism from the previous simulation that used the best derivation, here updates proceed by *adding* the best parse from the Interpreter module to the grammar. Finally, the agent constructs a weighted treebank from a set of 100 randomly generated semantic structures. These treebanks are then used to seed a new generation of agents that inhibit the grid at the next iteration.

The description so far is merely a multi-agent extension of the previous simulation. The present simulation fully implements the model of language transmission of chapter 2, shown in figure 4.19. The heterogeneity of the community and the differences between L1 and L2 acquisition are modelled as follows.

At initialization, 1/5 of the population, connected on the grid, is assigned Viking status. This entails two things. (1) The treebank they extract their

grammar from contains only sentences with SVO word order. This is not meant to reflect their native language, but the knowledge they have of English. (2) in updating their grammar, Vikings extract the matrix clause from the best parse tree from their Interpreter and add that to their grammar, reflecting the observations from Clahsen and Muyskens about the nature of L2 acquisition.

The other 4/5 of the population are Brits. Their assigned treebank at initialization contains both OV-V2 sentences and sentences with extraposition constructions. The ratio between these two is governed by the leakage parameter L. Sentences with the OV-V2 pattern have relative frequency of $(1 - L)$. Sentences with the VO pattern have relative frequency of L in the treebank.

When updating their grammar, agents with Brit status add the entire tree from the Interpreter module, reflecting their ability to add 'peripheral rules' [54].

Child language acquisition proceeds in the way described in chapter 3, by extracting a DOP1 grammar from the ancestor's weighted treebank.

Throughout the simulations, the relative frequency of OV and VO sentenes in the entire population's linguistic output is tracked.

4.3.3 Results

The value for L, denoting the ratio of leakages, to canonically constructed sentences, turned out to be crucial for the results of the simulation. Figure 4.20 shows the average over 10 runs of the simulation for $L = 0$. As can be seen, the population fairly rapidly settles on a 50/50 state, with both word order types equally likely to occur. Single runs for $L = 0$ followed the same pattern.

More interesting are the results for higher values of L. Figures 4.21, 4.22, 4.23 show the average results for these settings. They show that higher values for L produce situations where the VO word order dominates the OV word order. The higher the level for L, the greater the relative frequency of VO.

Two things are of note here. (1) Setting L to 0.5 may be considered implausible at best. If the frequency of leaked constructions was so high in English, it might have changed word order without external pressure at all. We consider $L = 0.1$ and $L = 0.25$ the only plausible settings that we discuss here. (2) In none of the runs we performed the OV word pattern completely vanished. Even the $L = 0.5$ runs did not produce this outcome. The best we could produce were convergence levels around 0.8.

Evidently, this does not concur with the actual historical development. Dur-

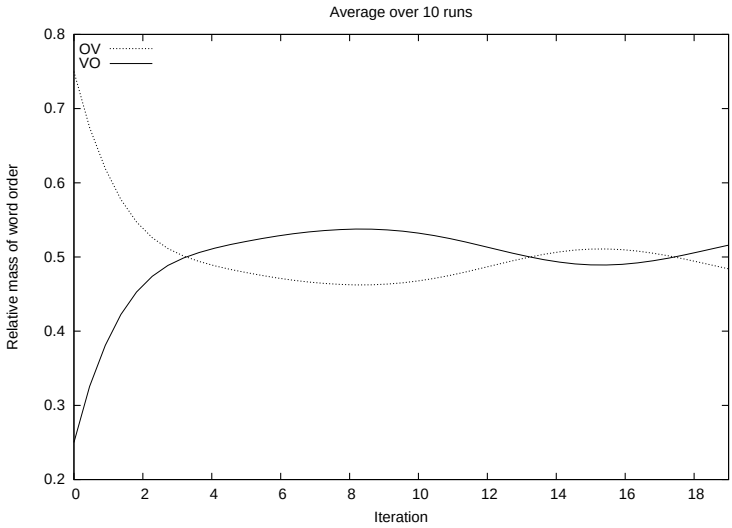


Figure 4.20: Relative frequency of word order for $L = 0$

ing the 200 years of word order change, the extrapositions also disappeared and word order was frozen to VO. In generative, non-probabilistic theories of the word order change, this fact does not pose much of a problem (cf. [54, 50]). It seems, however, that it does pose a problem for a probabilistic model.

But this does not necessarily mean that the simulation has failed. Concurrent with the word order change, another major change to the English morphosyntactic system took place: the loss of case markers [50]. As we showed in the previous section, our model can in principle account for word order freezing together with loss of case, if the pressure of phonological erosion is high enough. A combination of these two results may lead to the following conjecture.

The correct and full acquisition of case inflection is notoriously difficult for adult L2 learners (cf. [20]). The presence of a large portion of the population that does not produce case inflection consistently may have put more pressure on the phonological erosion. As we saw in the results from the previous simulation, high rates of erosion could lead to a completely frozen word order, even in a probabilistic model. In situations where all word orders are equally likely, this leads to different end states. In situations where one word order is already more likely to be observed than others, the effect was invariably an end state with a

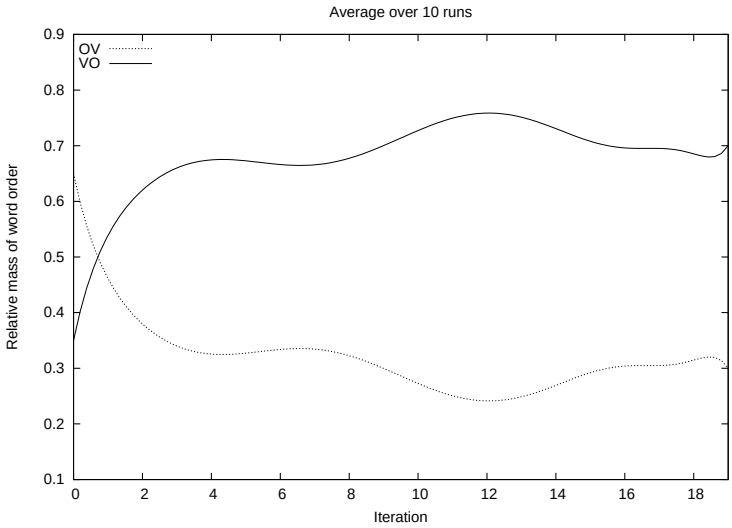


Figure 4.21: Relative frequency of word order for $L = .1$

completely frozen word order.

The solution to explaining the word order change in English, while basing oneself on a probabilistic language model, may therefore lie in a combination of these approaches. The simulation in this section showed that our model could explain a relative dominance of the SVO word order in the linguistic output of the community. The combination with increased phonological erosion may have been enough to completely freeze the English word order to SVO.

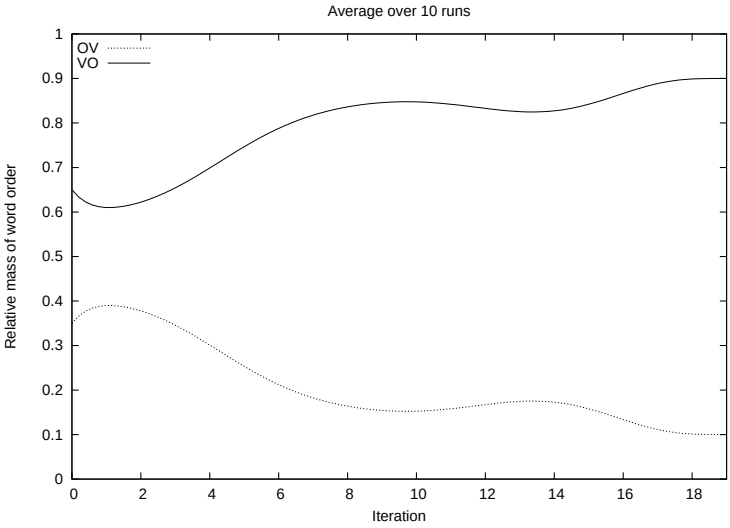


Figure 4.22: Relative frequency of word order for $L = .25$

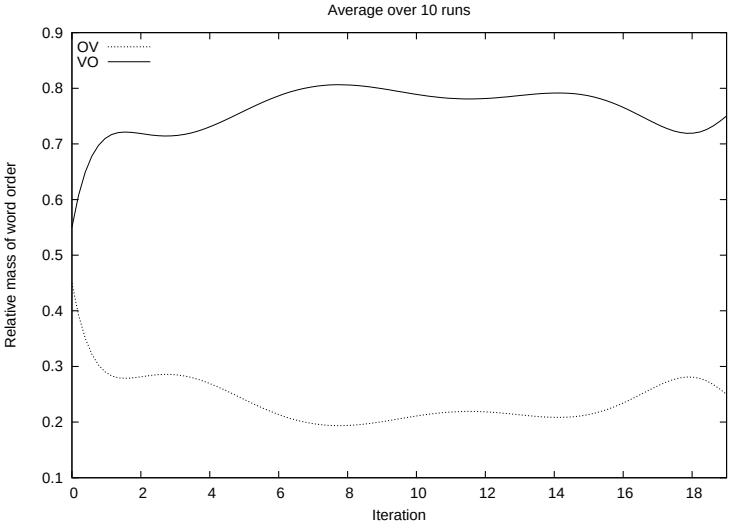


Figure 4.23: Relative frequency of word order for $L = .5$

Chapter 5

Conclusion

In chapter 1 we stated two goals for this thesis. (1) Develop and implement an iterated learning model of syntactic change that can be applied to a wide variety of historical phenomena. (2) Apply the model to case studies from historical linguistics and report on the results.

From a survey of the literature discussed in chapter 2, we concluded that our model should include the following factors, that were found to be important to syntactic change. Central to the model is the iterated application of language learning mechanisms.

Insights gained from a survey of grammaticalization theory pointed to the importance of frequency effects and analogy/reanalysis in syntactic change. These ideas led us to adopt Data-Oriented Parsing as our language model, because its probabilistic, all-subtrees approach enabled us to account for both these factors. Through DOP we were able to account for both instances of language learning. Child language acquisition is modelled as the extraction of a grammar from a treebank. Adult modification is modelled in several different ways, each pertaining to a modification of the subtree probabilities in the DOP-grammar.

In chapter 4 we developed the simulations that we used to study several linguistic phenomena. First we showed that a simple version of our model was able to adequately simulate reanalysis and analogical levelling. These results are important for the following reason. As we discussed in chapter 2, the grammaticalization approach regards reanalysis and analogy as singularly important mechanisms of language change. We only showed results for two simple linguistic examples, but the fact that we were able to obtain linguistically plausible results from these simulations means that we were successful in developing a

model that can simulate these mechanisms. Because of their centrality to grammaticalization theory, this indicates that extensions to our most basic model can be used to simulate a wide variety of phenomena in historical linguistics.

The last two simulations we discussed in chapter 4 showed examples of this. We modelled two complex phenomena using extended versions of our model, (1) the relation between word order freeness and case marking and (2) a major syntactic change in the English language. The results from these simulations showed two things. First, they showed that our model can produce realistic outcomes for the simulations. In the case of the word order freezing model the resultant stable states reflected naturally occurring language systems. Second, the results from the simulations can be used to evaluate existing theories in historical linguistics. In the simulation of word order change, our results indicated that the effects of language contact in isolation may not be enough to account for the full shift in English from a free OV-V2 language to a fixed order VO language.

Results like these should be strong motivations for linguists to develop quantitative models of language and language change. Implementing a language model forces formal exactness and devising and studying simulations may provide valuable insights that can otherwise be hard to come by. One of the goals of this thesis was to provide a contribution to the development of such formal models of syntactic change. This thesis has contributed to the integration into a single model of language transmission of a number of the factors at play in language change, namely acquisition, community dynamics, language contact and the interplay between different modalities of language, specifically syntax, semantics and phonology. We hope to have shown that it is possible to develop a relatively simple model of language change that can produce interesting results.

Further research based on this model can take a number of directions. Obvious improvements can be made to the language acquisition mechanism to make it more realistic, for example by employing Bod's [7] Unsupervised Data-Oriented Parsing model. In order to model full grammaticalization clines, more elaborate representations of semantics, morphology and phonology could be added. It would be interesting to see what results a combination of these two extensions would yield. Further research can be done by performing case studies other than the ones performed here.

Bibliography

[1] H. Andersen. Abductive and deductive change. *Language*, 48:765–769, 1973.

[2] M.D. Atkinson, J.-R. Sack, N. Santoro, and T. Strothotte. Min-Max heaps and Generalized Priority Queues. *Communications of the ACM*, 29(10), 1986.

[3] Rens Bod. Data Oriented Parsing. In *Proceedings COLING-91*, 1991.

[4] Rens Bod. The problem of computing the most probable tree in data-oriented parsing and stochastic tree grammars. In *Proceedings of the Seventh Conference of the European Chapter of the ACL*, Dublin, Ireland, March 1995.

[5] Rens Bod. Parsing with the shortest derivation. In *Proceedings COLING-2000, Saarbruecken, Germany*, 2001.

[6] Rens Bod. Unsupervised Parsing with U-DOP. In *Proceedings of the 10th conference on computational natural language learning*, pages 85–92. Association for Computational Linguistics, 2006.

[7] Rens Bod. From exemplar to grammar: a probabilistic analogy-based model of language learning. *Cognitive Science*, 33(4), 2009.

[8] Rens Bod, Remko Scha, and Khalil Sima'an. *Data-Oriented Parsing*. CSLI Publication, 2003.

[9] R. Bonnema and R. Scha. Reconsidering the probability model for dop. In Rens Bod, Remko Scha, and Khalil Sima'an, editors, *Data-Oriented Parsing*. CSLI Publications, 2003.

[10] E.J. Briscoe. Grammatical acquisition and linguistic selection. In E.J. Briscoe, editor, *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press, Cambridge, 2002.

[11] Joan Bybee. From usage to grammar: The mind's response to repetition. *Language*, 82(4):711–733, 2006.

[12] Joan Bybee and Sandra Thompson. Three frequency effects in syntax. In *Berkeley Linguistics Society 23: General Session and Parasession on Pragmatics and Grammatical Structure*, pages 65–85, Berkeley, 1997. Berkeley Linguistics Society.

[13] Lyle Campbell. *Historical Linguistics*. Edinburgh University Press, 2004.

[14] David J. Chalmers, Robert M. French, and Douglas R. Hofstadter. High-Level Perception, Representation, and Analogy: A Critique of Artificial Intelligence Methodology. Technical Report CRCC-TR-49, Center for Research in Concepts and Cognition, Indiana University, 1991.

[15] Noam Chomsky. *Knowledge of language: its nature, origins and use*. Praeger, New York, 1986.

[16] Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, Massachussets, 1995.

[17] Harald Clahsen and Pieter Muysken. The availability of universal grammar to adult and child learners; a study of the acquisition of german word order. *Second Language Research*, 2:93–119, 1986.

[18] Olga Fischer. Grannaticalisation: unidirectional, non-reversible? In O. Fischer, A. Rosenbach, and D. Stein, editors, *Pathways of Change*, pages 149–169. Benjamins, 2000.

[19] Olga Fischer. *Morphosyntactic Change,* Functional and Formal Perspectives. Oxford Surveys in Syntax and Morphology. Oxford University Press, Oxford, 2007.

[20] Susan Gass and Larry Selinker. *Second language acquisition*. Lawrence Erlbaum Associates, 2001.

[21] Dedre Gentger and Arthur Markman. Defining Structural Similarity. *Journal of Cognitive Science*, 6:1–20, 2005.

[22] Dedre Gentner. Analogy. In *The MIT encyclopedia of the cognitive sciences*, pages 17–20. MIT Press, Cambridge, 1999.

[23] Marinel Gerritsen. *Syntaktische verandering in kontrolezinnen*. Foris, 1987.

[24] Marinel Gerritsen and Dieter Stein, editors. *Internal and External Factors in Syntactic Change*. Mouton de Gruyter, 1992.

[25] Adele Goldberg. *Constructions at Work: the nature of generalization in language*. Oxford University Press, 2006.

[26] Joshua Goodman. Efficient parsing of DOP with PCFG-reductions. In Rens Bod, Remko Scha, and Khalil Sima'an, editors, *Data-oriented parsing*. CSLI publications, 2003.

[27] A. Harris and L. Campbell. *Historical Syntax in a Cross-Linguistic Perspective*. Cambridge University Press, Cambridge, 1995.

[28] Paul Hopper and E. Traugott. *Grammaticalization*. Cambridge University Press, Cambridge, 2003.

[29] Liang Huang and David Chiang. Better k-best Parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies*, Vancouver, B.C., 2005.

[30] Otto Jespersen. *Negation in English and other languages*. A.F. Høst, Copenhagen, 1917.

[31] Mark Johnson. The dop estimation method is biased and inconsistent. *Computational Linguistics*, 28(1):71–76, 2002.

[32] T. Kasami. An efficient recognition and syntax analysis system for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA, 1965.

[33] Simon Kirby. Learning, bottlenecks and infinity: a working model of the evolution of syntactic communication. In *Proceedings of the AISB'99 symposium on imitation in animals and artifacts*, pages 55–63, 1999.

[34] Simon Kirby, Mike Dowman, and Thomas Griffiths. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12):5241–5245, 2007.

[35] Dan Klein and Christopher Manning. Parsing and hypergraphs. In *Proceedings of the Seventh International Workshop on Parsing Technologies*, Beijing, China, 2005.

[36] William Labov. *Principles of Linguistic Change: Internal Factors*. Blackwell, Oxford, 1994.

[37] William Labov. *Principles of Linguistic Change: Social Factors*. Blackwell, Oxford, 2001.

[38] Erez Lieberman, Jean-Baptiste Michel, Joe Jackson, Tina Tang, and Martin Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449:713–716, 2007.

[39] David Lightfoot. *Principles of Diachronic Syntax*. Cambridge University Press, Cambridge, 1979.

[40] David Lightfoot. The history of noun phrase movement. In C.L. Baker and J. McCarthy, editors, *The Logical Problem of Language Acquisition*, pages 86–119. MIT Press, Cambridge, Massachussetts, 1981.

[41] David Lightfoot. *The Development of Language,* Acquisition, Change and Evolution. Blackwell Publishers, Malden, Massachusetts, 1999.

[42] April McMahon. *Understanding language change*. Cambridge University Press, 1994.

[43] Daniel Nettle. Using Social Impact Theory to simulate language change. *Lingua*, 108:95–117, 1999.

[44] Jerzy Kuryłowicz. Zur Vorgeschichte des germanischen Verbalsystems. In *Beiträge zur Sprachwissenschaft, Volkskunde und Literaturforschung: Wolfgang Steintz zum 60. Geburtstag*, pages 242–247. Akademie-Verlag, Berlin, 1965.

[45] Andrew Radford. *Transformational grammar*. Cambridge University Press, 1988.

[46] Remko Scha. Taaltheorie en taaltechnologie; competence en performance. In R. de Kort and G.L.J. Leerdam, editors, *Computertoepassing in de Neerlandistiek*, pages 7–22, 1990.

[47] Khalil Sima'an. Computational complexity of probabilistic disambiguation by means of tree grammars. In *Proceedings of the 14th Computational Linguistics Conference (COLING'96)*, Copenhagen, Denmark, August 1996. Association for Computer Linguistics.

[48] Khalil Sima'an. Computational complexity of disambiguation under DOP1. In *Data-oriented parsing*. CSLI publications, 2003.

[49] Luc Steels. Language learning and language contact. In *Proceeding of the workshop on Empirical Approaches to Language Acquisition*, pages 11–24, 1997.

[50] Carola Trips. *From OV to VO in Early Middle English*. John Benjamins Publishing, 2002.

[51] Evert van den Berg. Het middelnederlands als sov taal. *De Nieuwe Taalgids*, pages 3–60, 1980.

[52] Nicoline van der Sijs. *De geschiedenis van het Nederlands*. Bert Bakker, 2005.

[53] Ans van Kemenade. *Syntactic Case and Morphological Case in the History of English*. Foris, 1987.

[54] Fred Weerman. The diachronic consequences of first and second language acquisition: the change from ov to vo. *Linguistics*, 31:903–931, 1993.

[55] Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *Proceedings of the first international conference on human language technology research*, pages 1–5. Association for Computational Linguistics, 2001.

[56] D.H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189–208, 1967.

[57] Andreas Zollmann and Khalil Sima'an. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics*, 10(2/3):367–388, 2005.