EXPLOITING SYSTEMATICITY: A CONNECTIONIST MODEL OF BOOTSTRAPPING IN LANGUAGE ACQUISITION

MSc Thesis (*Afstudeerscriptie*)

written by

Hélène Tourigny
(born July 13, 1975 in Ottawa, Ontario, Canada)

under the supervision of **Dr Stefan Frank**, 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 30, 2010	Dr Stefan Frank
	Prof Dr Rens Bod
	Prof Dr Frank Veltman

Institute for Logic, Language and Computation

I would like to thank all members of my thesis committee: Frank Veltman, Rens Bod, and Stefan Frank. I am particularly grateful for the support provided to me by my supervisor, Stefan Frank, who was not only incredibly generous with his time, but also provided many insightful comments and suggestions during our many discussions. Most of all, I wish to express my appreciation for his constant encouragement and his unwavering confidence in me, which were instrumental in allowing me to complete this project.

Abstract

This thesis presents a connectionist model of syntactic bootstrapping processes in language acquisition. According to the Syntactic Bootstrapping hypothesis, children acquiring language can learn (part of) the meaning of new words based on the syntactic context in which they appear. Psycholinguistic research has shown that children can indeed use morphosyntactic cues to guide their interpretation of novel words.

This project investigates whether a connectionist network can exploit systematicity in language to acquire novel words over the course of development. The network is trained according to a semi-supervised algorithm. The model learns a sentence-interpretation task from both labelled and unlabelled data. Specifically, it learns to output a semantic representation (roughly corresponding to 'who did what to whom') for given sentences. To investigate whether syntactic bootstrapping can successfully lead to lexical development, some vocabulary items are only presented in unlabelled sentences. To correctly process these examples and learn these words, the network must infer the novel words' properties (e.g. grammatical category, animacy features) based on the context in which they appear. The network must then use its own output (i.e. its interpretation of the sentences) to train itself.

The system's ability to rely on syntactic cues for vocabulary acquisition is tested in a number of experiments. Although the network is able to acquire the language and shows very good generalization, its ability to rely on syntactic bootstrapping to learn novel words does not meet expectations.

Contents

Chapter 1

Introduction

A key aspect of natural language is its systematicity. Much of the expressive power of language lies in the fact that innumerably many sentences can be created from a finite number of words and structures. Regularities in language allow native speakers to produce and understand sentences they have never heard before. For example, a speaker of English who understands the sentence *John loves Mary* will also understand the sentence *Mary loves John*. Although systematicity is often discussed in terms of production and comprehension, regularities in language are also useful during the process of acquiring a language. One very influential theory of vocabulary acquisition, known as the Syntactic Bootstrapping hypothesis (Landau and Gleitman, 1985; Gleitman, 1990), depends on systematic correspondences between syntax and semantics.

The Syntactic Bootstrapping hypothesis holds that children acquiring language can learn (part of) the meaning of new words based on the syntactic context in which they occur. This is possible because there are strong correlations between form and meaning. For example, the number of arguments licensed by a verb is closely related to its meaning: verbs that describe one-participant actions are typically intransitive (e.g. *Bill laughed*), while verbs that describe events involving two participants typically appear in transitive frames (e.g. *Anna tickled Bill*) (Fisher et al., 2010).

Experimental research has shown that children can learn about the meaning of novel words based on the syntactic context in which these words occur. Nevertheless, little is known about the role of syntactic bootstrapping in the 'usual' course of lexical development. While it is clear that children can rely on linguistic cues to learn about the meaning of words in laboratory settings, it has yet to be demonstrated that bootstrapping can eventually lead to the acquisition of new vocabulary, i.e. that it can lead to long-term learning.

In addition to psycholinguistic experiments, computational simulations can be useful for investigating the validity of theories of language acquisition. In particular, models of syntactic bootstrapping can be helpful for determining under what conditions bootstrapping mechanisms might allow learners to acquire new vocabulary. Indeed, computational simulations have provided supporting

evidence for the use of bootstrapping processes for *comprehension* of novel words appearing in familiar syntactic frames. Computational models have been shown to make appropriate inferences about novel words presented in familiar syntactic contexts. However, these models are only presented with novel words during *testing*, i.e. after the model has acquired the input language. What has yet to be explored is whether a model can make use of inferred knowledge during *training* to acquire new vocabulary over time. For bootstrapping to be plausible as a learning strategy, it must be shown that learners can integrate knowledge inferred from linguistic context with already-existing knowledge.

In this thesis, I investigate whether a connectionist network can exploit systematicity in language to acquire novel words over the course of development. The network is trained according to a semi-supervised algorithm. The model learns a sentence-interpretation task from both labelled and unlabelled data. Specifically, it learns to output a semantic representation (roughly corresponding to 'who did what to whom') for given sentences. To investigate whether syntactic bootstrapping can successfully lead to lexical development, some vocabulary items are only presented in unlabelled sentences. When presented with an unlabelled example containing a novel word, the network must infer (part of) of the word's meaning (e.g. grammatical category, animacy features) based on the context in which it appears. The network must then use its own output (i.e. its interpretation of the sentences) to train itself.

The model also provides an opportunity to explore the issue of systematicity in connectionist systems from a new perspective. The capacity of neural networks to display the same level of systematic behaviour as humans has been heavily debated (for discussion, see, among others, Hadley (1994), Christiansen and Chater (1994), and Frank and Čerňanský (2008)). Much of the debate has focused on networks' ability to process sentences containing *known* words appearing in new *contexts*. However, as noted earlier, regularities in language can also be exploited to make inferences about *novel* words appearing in *familiar* contexts. A model of bootstrapping must display systematic behaviour in order to successfully acquire vocabulary, because it must exploit regularities in the input to make inferences about novel words. The model developed for this thesis can thus provide a new perspective on systematicity in neural networks.

The thesis is organized as follows. In chapter 2, I discuss how regularities in language can be exploited for language comprehension and language acquisition, both by children and computational models. First, I briefly discuss the issue of systematicity in connectionist models. I then give an overview of the Syntactic Bootstrapping hypothesis, and present existing computational models of syntactic bootstrapping. Chapter 3 introduces the connectionist model developed for this thesis. After presenting the language learned by the model and the training and test data, I describe the model's architecture, the semi-supervised training procedure and the criteria used to evaluate the model's performance. In chapter 4, I discuss the results of the simulations performed with the model, which test the system's ability to exploit systematic properties of the input data. Chapter 5 concludes the thesis.

Chapter 2

Background

2.1 Systematicity and connectionism

The capacity to create and understand novel sentences is an important feature of human language processing. Human beings have the ability to generalize – to discover underlying patterns in the data they are presented with, and to apply this knowledge when processing or uttering new sentences. Fodor and Pylyshyn (1988) argued that cognitive models of language should be *systematic* in this sense, and that connectionist networks cannot display systematicity without implementing a classical symbol system. The ability of neural networks to display systematic behaviour is still a matter of debate.

Studies of systematicity in connectionist models have focused on the ability (or lack thereof) of networks to display systematic behaviour when faced with a sentence processing task. What is evaluated is the model's ability to process sentences containing words that occur in new combinations or positions. For example, suppose a network has encountered the word *giraffes* only in subject position during training, and the word *apes* only in object position. The network displays systematicity if it can process a sentence like *apes see giraffes*, where the positions of the nouns are reversed.

Several researchers have investigated the ability of networks to display this kind of systematicity (see, among others, Hadley (1994), Christiansen and Chater (1994), van der Velde et al. (2004), Frank and Čerňanský (2008)), and Frank et al. (2009)). In most experiments, networks are trained on a grammar where the distribution of certain words is unnaturally restricted, so that the model's ability to process words appearing in novel positions can be tested. The problem with this approach is that it is very difficult to create a grammar that is sufficiently restrictive to allow the investigation of systematic behaviour without simultaneously requiring the system to overgeneralize.

An alternative approach is to test a model on sentences containing words that were not seen during training at all, and to ask whether the model can rely on its knowledge of syntax and semantics to interpret these utterances.

If it can process the sentences successfully, the system can be said to display systematicity.

In this thesis, I look at the issue of systematicity from this angle, by asking whether neural networks can *exploit* the regularities prevalent in language to make inferences about novel words. For example, someone who knows the verb *sleep* and hears the sentence *'Wugs sleep'* will infer that *wugs* is a noun, and refers to an *animate* entity, even if they have never encountered this word before.

This aspect of systematicity has received little attention in the debate on systematic behaviour in connectionist models, perhaps because systematicity in sentence processing is a prerequisite for inferring word meaning based on context. As we will see in the next section, the ability to rely on context to learn about the meaning of novel words is known as Syntactic Bootstrapping. In this thesis, I develop a model of bootstrapping, and explore the issue of systematicity in connectionist networks from a language acquisition perspective.

2.2 Language acquisition

2.2.1 Syntactic bootstrapping

Learning novel words is difficult. Infants learning their native language must find correlations between the utterances they hear and the real world, and learn which words correspond to which objects, concepts, relations or events. However, as observed by Quine (1960), lining up linguistic input with extralinguistic scenes is not an easy task. A number of problems arise for the learner attempting to relate linguistic input with her experience of the world. Even in simple cases, (e.g. an adult saying "This is a cat" or "Look, a cat!" while pointing at a cat), several possible interpretations are available; the child might infer that the word refers to this particular cat, to animals in general, to furriness, to white objects, etc. (Landau and Gleitman, 1985). The problem is even more complicated in the case of abstract concepts (e.g. *thought, beauty*), which have no direct instantiation in the real world.

In addition, as observed by Gleitman (1990), a given situation can typically be construed in a number of ways, thereby allowing any number of pairings between word and world. For example, verb pairs like *lead/follow* or *chase/flee* describe the same event from different perspectives. Moreover, caregivers' speech to infants is not perfectly aligned with the events observable to children, and the focus of attention of the child and adult may differ. For example, the adult might tell the child "Come take your nap" while the child's attention is directed at a cat (Gleitman and Gillette, 1995).

Given the difficulties involved in building a lexicon based solely on observation, Landau and Gleitman (1985) and Gleitman (1990) hypothesized that both extralinguistic and linguistic cues guide vocabulary acquisition, a process known as *syntactic bootstrapping*.

According to the bootstrapping hypothesis, children use linguistic cues to guide their interpretation when mapping utterances to scenes, i.e. they can use

syntax to determine what particular words refer to, and to constrain the range of possible correlations between the world and the linguistic input (Landau and Gleitman, 1985; Gleitman, 1990; Gleitman and Gillette, 1995).

Morphology and syntax can provide powerful cues to meaning. For example, function words like determiners and pronouns are highly indicative of the lexical category of surrounding words (e.g. *a blick* suggests that *blick* refers to a concrete noun, *he's blicking* suggests a verb, and the morphological marker *ish* in *a blickish cat* would indicate that it is an adjective). Children's ability to rely on such cues to interpret novel words was first demonstrated by Brown (1957), who showed that preschoolers could use morphosyntactic cues to infer the lexical category of a novel word. For example, when hearing *a sib*, they associated the novel word *sib* with a novel object, whereas when presented with *sibbing*, they associated the new word with an action.

2.2.2 Experimental evidence

Mapping utterances to scenes

Since Brown (1957), numerous other studies have shown that both adults and children can use distributional cues in language to infer (partial) meaning of novel words, and that syntax supports vocabulary acquisition. As noted by Naigles and Swensen (2007), there is now ample evidence that preschool-aged children (3- to 5-year-olds) make use of syntactic bootstrapping to learn novel words, and even toddlers appear to rely on linguistic context during comprehension.

Several experiments have focused on verb learning, investigating children's ability to rely on the syntactic frame in which a novel verb appears to make inferences about its meaning. For example, in an experiment conducted by Naigles (1998), children were shown side-by-side videos depicting characters involved in two types of actions. One action was causative, e.g. a duck making a rabbit bend over, and the other was non-causative and synchronous, e.g. a duck and a rabbit making arm circles. These scenes were paired with sentences introducing a novel verb (e.g. 'blicking') in either a transitive (e.g. "The duck is blicking the bunny") or intransitive frame (e.g. "The duck and the bunny are blicking").

Children's interpretation of the novel words were tested by asking them to "find blicking" while showing the two actions on separate screens. One video depicted the causative action, the other the non-causative action. Children who had heard the novel verb in transitive frames looked significantly longer at the scene depicting the causative action, while those who had heard intransitive sentences looked longer at the scene showing the non-causative action (Naigles, 1998). For discussion and additional references to similar work, see also Naigles and Swensen (2007); Naigles (1996); Fisher et al. (2010).

Multiple frames

In many experimental validations of the syntactic bootstrapping hypothesis, children are exposed to utterances where a novel word is presented in a single, unambiguous, context (e.g. in either a transitive or intransitive frame, as in the experiment described above). However, in many cases, exposure to a novel word in a single syntactic frame is insufficient for determining the word's meaning.

As an illustration, consider the following example from Gleitman and Gillette (1995). The sentence "John is ziking the book to Bill" suggests that *ziking* is a verb of transfer, but it is compatible with a variety of concepts, such as *give, bring, throw, explain*. Hearing this word in a variety of contexts could provide additional information about its semantics; for instance, "John is ziking that the book is boring" would indicate a mental activity. Taken together, these two utterances would then suggest that the meaning of *ziking* is analogous to that of *explain* (Gleitman and Gillette, 1995, p. 216).

Since learners must sometimes hear a novel word in a variety of syntactic structures in order to gain a full picture of its meaning and distribution, for bootstrapping to be plausible as a theory of word learning, children must be capable of keeping track of the distributional properties of words. Naigles (1996, 1998) provided experimental demonstrations of children's ability to use information from multiple frames to discover word meanings. The experiments relied on English transitivity alternations: the *causative alternation* and the *omitted object* (or *unspecified object*) alternation. These alternations are illustrated in (1) and (2) (Naigles, 1996, p. 226).

(1) a. The girl dropped the **ball**.
    b. The **ball** dropped.

(2) a. The **cat** was scratching the door.
    b. The **cat** was scratching.

Syntactically, these verbs are indistinguishable in the transitive frame. The difference between the two classes surfaces in the intransitive use of the verbs: in the causative alternation (1), the subject of the intransitive verb corresponds to the *object* in the transitive frame; in the omitted object alternation (2), the subject of both sentences is the same, and the affected object is unspecified. Only by comparing pairs of sentences like those in (1) and (2) can the distinction between these verbs be discovered.

These verbs differ only slightly in their lexical semantics. Causative verbs involve actions where the subject causes a change in the state or position of the object, while verbs presented in the omitted object condition involve "contacting" activities – actions with repeated contact, but no change-of-state or change-of-position. Examples of 'contact' verbs include *touch, pat, stroke*.

Naigles (1998) taught toddlers novel verbs similar to causatives like *move* (e.g. *The duck krads the* ***frog****/ The* ***frog*** *krads*) or 'contact' verbs like *pat* (e.g. *The* ***duck*** *krads the frog/ The* ***duck*** *krads*). Children were able to use information presented over multiple frames to distinguish between these verbs.

Inferring meaning in the absence of extralinguistic context

The studies mentioned above show that syntactic bootstrapping can be useful in the presence of corresponding extralinguistic input, but another aspect of the theory is that learners should be able to infer (partial) information about novel words even in the absence of perceptual cues. This is particularly important given that it would be impossible to learn language solely based on cross-situational observation.

Recent research has investigated children's ability to rely on bootstrapping strategies to infer the meaning of novel verbs even when no semantic cues are available from an extralinguistic scene. Yuan and Fisher (2009) showed that children can learn about the syntactic properties of verbs based solely on linguistic cues, while Scott and Fisher (2009) demonstrated that information about *semantic* properties of verbs can be learned from the linguistic context in which verbs appear.

Yuan and Fisher (2009) presented 2-year-olds with videos showing two women engaged in conversation. The women used a nonsense verb in a transitive (*Jane blicked the baby!*) or intransitive (*Jane blicked!*) frame. Since the videos did not show a novel action, children could rely only on syntax to learn about the verb's meaning. The experiment showed that children can indeed make use of purely linguistic cues to distinguish between transitive and intransitive actions. Children's interpretation of the verb was then tested by showing them a scene depicting either a two-participant event or a one-participant event. The experiment showed that children who had heard novel verbs in transitive frames correctly associated the verb with two-participant events, while those who had heard intransitive sentences associated the verb with a one-participant event.

This experiment from Yuan and Fisher (2009) shows that even without extralinguistic cues, children can learn about the *syntactic* properties of verbs, i.e. the number of arguments it takes. Scott and Fisher (2009) extended this work by investigating whether children can also learn about abstract *semantic* properties of verbs based on linguistic cues. More precisely, they investigated whether children can use linguistic cues to make inferences about the semantic roles a verb assigns to its arguments.

The experiment relied on the transitivity alternations discussed earlier. In the causative alternation, the subject of the intransitive variant is assigned an *Undergoer/Patient/Theme* role. That is, the subject of the intransitive sentence is the object affected by the action described by the verb. The Agent is left unspecified. This is illustrated in (3). In the omitted object alternation, the subject of the intransitive sentence is the Agent, and the object is left unspecified, as in (4) (Scott and Fisher, 2009, p. 778).

(3)	a.	Anne broke the lamp.	Causative alternation
	b.	The lamp broke.	(Undergoer subject)
(4)	a.	Anne dusted the lamp.	Omitted object alternation
	b.	Anne dusted.	(Agent subject)

In an experiment similar to that of Naigles (1998), Scott and Fisher (2009) showed that children presented with pairs of sentences like those in (3) and (4) could distinguish between the causative and non-causative meanings of verbs, even when they did not have access to a concurrent extralinguistic scene.

To sum up: there is ample evidence that children can use linguistic cues to guide their interpretation of novel words, even without a corresponding scene. In addition, children can keep track of the combinatorial privileges of verbs over multiple trials and frames to progressively refine their understanding of a word. Nevertheless, as noted by Naigles and Swensen (2007), little is known about the role of syntactic bootstrapping in the 'usual' course of lexical development. While it is clear that children can rely on linguistic cues to learn about the meaning of words in laboratory settings, it has yet to be demonstrated that bootstrapping can eventually lead to the acquisition of new vocabulary, i.e. that it can lead to long-term learning.

2.3 Related work

In this section, I give an overview of computational models of syntactic bootstrapping. As we will see, previous modelling work on bootstrapping has focused on issues related to the learnability of language. The computational models of Desai (2002, 2007); Allen (1997); Alishahi and Stevenson (2010), among others, demonstrate that learners can exploit regularities in language to discover correlations between form and meaning. Experimental and computational work support the hypothesis that linguistic context can allow the learner to infer components of the meaning of novel words.

The models reviewed in this section provide evidence for the importance of bootstrapping in comprehension, and support a usage-based approach to language acquisition, where multiple sources of information are combined to learn abstract properties of language, such as constructions or linking rules between syntax and semantics.

However, there are other aspects of syntactic bootstrapping which have yet to be investigated experimentally or computationally. In particular, few studies have explored whether knowledge obtained primarily from linguistic context – without extralinguistic input – can be integrated with previously-existing knowledge. The model developed for this thesis focuses on this issue, and thus extends or complements previous computational work on syntactic bootstrapping.

2.3.1 Desai (2002, 2007)

Desai (2002) presents a model of bootstrapping in the acquisition of a miniature language and shows that a Simple Recurrent Network trained on a sentence interpretation task can exploit syntax-semantics correspondences to infer part of the meaning of novel words. The model learns to process two types of inputs: sentence fragments (noun phrases) and simple sentences. Noun phrases consist of either a determiner and a noun (e.g. *a boy*) or two conjoined noun phrases

(e.g. *a boy and a girl*). Simples sentences contain either a transitive or intransitive verb. The grammar therefore licenses the following types of inputs, where N denotes a noun phrase, and V denotes a verb (Desai, 2002).[1]

(5) a. a boy (N)
b. a boy and a girl (NN)
c. a boy is jumping (NV)
d. a boy and a girl are jumping (NNV)
e. a boy is pushing a girl (NVN)

The network is trained on input pairs consisting of *utterances* (sequences of words) and *scenes* (semantic representations of the corresponding utterances). The task is to determine the semantics of input sentences, which entails identifying the first and second noun phrases, the verb, and whether or not the event involves causation.

The model's ability to perform syntactic bootstrapping is tested by presenting it with different sentences containing the novel word *glorp* in a variety of structures. The model assigns appropriately different representations to the semantic component representing *glorp*, depending on the structure in which it occurs. For instance, when presented with the fragment *a glorp*, the model interprets the novel word as a noun; when given the sentence *a girl is glorping*, the novel word is interpreted as an intransitive verb. The network thus exhibits the ability to infer part of the meaning of novel words based on linguistic cues.

One of the limitations of this task is that the identification of a nominal argument is only related to its linear position in the sentence, and does not depend on its *syntactic* position or semantic role. For example, in (5-d), the input *girl* is a subject and an Agent, while in (5-e), it is an object and Undergoer, but in both cases, it would receive the same semantic representation.

In addition, in this miniature language, causality is strictly related to transitivity: transitive verbs involve causation, while intransitive verbs do not. Thus, the network can fully identify the semantics of a novel verb based on a single frame; it does not need to keep track of the different structures in which a verb can appear to determine its distributional properties. Ideally, a model of bootstrapping would incorporate more refined semantics, requiring the model to attend not only to the respective order of arguments, but also to their semantic roles.

Desai (2007) presents a connectionist network that learns a slightly more complex grammar. The language is similar to that of Desai (2002), but includes a causative alternation, with some verbs licensed in both transitive and intransitive frames (e.g. *A boy broke the window*/*The window broke*). The model provides an account of Frame and Verb compliance, and is not aimed at exploring bootstrapping mechanisms. The model does demonstrate that the network is capable of keeping track of syntactic information over distinct syntactic frames. However, the model's ability to make inferences about novel words

[1] The grammar also allows noun phrases modified by an adjective encoding the referent's size, with two potential values 'large' and 'small' (e.g. *a large dog*).

is not evaluated.

Although the model of Desai (2002) does demonstrate that the network can use syntactic cues to *interpret* sentences, it does not provide a robust validation of bootstrapping as a procedure for acquiring vocabulary. In particular, it does not allow us to explore whether knowledge of novel words can be integrated with already-existing knowledge, or whether word meaning can be progressively refined through exposure to a given word in various syntactic structures.

2.3.2 Allen (1997)

Allen (1997) presents a connectionist network that exhibits bootstrapping behaviour. The model learns to assign semantic roles to arguments based on semantic and syntactic cues. In this model, nouns are encoded in the input as distributed representations of semantic features, while verbs and prepositions are given localist representations (i.e. they do not encode any semantic features). More precisely, nouns are represented by an array of 390 semantic features based on the WordNet database (Miller et al., 1990). For example, a proper name like *Peter* would be encoded as [+human, +animate, +male, -vehicle, ...].

The output of the model represents semantic features of verbs (e.g. +act, +cause) and features associated with the arguments' roles. In total, 360 features are represented in the output of the network. The model is also provided with syntactic information, in the sense that the order of a verb's arguments is also presented as part of the input. The training data were created based on caretaker speech from the CHILDES database. The trained network is tested with a grammaticality judgement task. The network is presented with grammatical and ungrammatical novel sentences, for which it must output the corresponding semantics. If the network computes semantic roles for all and only the nominal arguments in the sentence, it is deemed to have judged the sentence as being acceptable.

To demonstrate the bootstrapping behaviour of the network, Allen (1997) supplied the network with utterances containing novel words, and examined the resulting interpretation. Given the sentence *John glorped the basket to Mary*, the network attributed an Agent role to *John*, an endpoint role to *Mary*, and a theme role for *basket*. The computed pattern also reflects information inferred about the novel verb *glorp*, including features such as [+move, +hand, +pass +transfer]. Since the network was not previously trained on this word, these semantic features are primarily deduced from the syntactic construction in which the verb appears.

The network can also rely on the semantics of lexical items to make inferences about novel words. For example, when presented with *He glorped the message to Mary*, which is syntactically similar to the preceding example, the network computes the same role interpretation as before, but the semantics of the verb now include features such as [+communicate, +interact, +express].

This model provides an interesting approach to modelling language acquisition, because it allows the learner to use multiple cues (e.g. number and order of

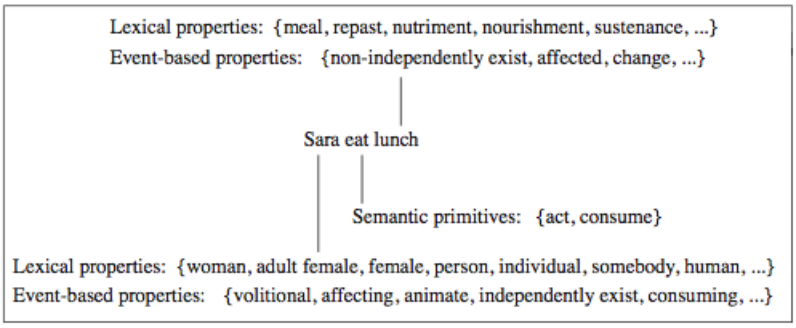

Figure 2.1: A sample verb usage: an utterance paired with the inferred semantic information. Reproduced from Alishahi and Stevenson (2010, p. 59, Figure 1)

arguments, semantic properties of nouns) to make inferences about the meaning of novel words. Indeed, although much research on syntactic bootstrapping has focused on morphosyntactic cues, it is also assumed that children also rely on conceptual cues to make inferences about novel words.

The drawback of this model is that it assumes that the child has very sophisticated and extensive semantic knowledge of both nominal arguments and events. As such, it cannot be taken as a model of bootstrapping processes occurring at an early stage of language acquisition. In addition, the grammaticality judgement task only provides a cursory evaluation method. Finally, like the model of Desai (2002), the network is only *tested* on utterances with novel words, so the model never makes use of *inferred* knowledge to learn new vocabulary.

2.3.3 Alishahi and Stevenson (2010)

Alishahi and Stevenson (2010) implement an incremental Bayesian model capable of using both syntactic and semantic information to guide its interpretation of novel verbs. This model, which is an extended version of their earlier work (Alishahi and Stevenson, 2007), learns not only linking rules governing the mappings between syntax and semantics, but also learns the semantic roles themselves. The semantic role labels are not pre-defined in this model; they are acquired from experience. In addition, the model develops knowledge of the distributional properties of individual verbs, and also generalizes this knowledge to discover different constructions.

The model is first presented with sample verb usages, which consist of an utterance paired with semantic information describing the corresponding event. The child is assumed to be capable of inferring semantic properties of events and participants, as well as being able to establish a link between the utterance and the scene. A sample verb usage, reproduced from Alishahi and Stevenson (2010), is given in Figure 2.1.

Head verb	*eat*
Number of arguments	2
Syntactic pattern	arg1 *verb* arg2
Semantic primitives of verb	{act, consume}
Lexical properties of argument 1	{woman, adult female, female, person, individual, ...}
Event-based properties of argument 1	{volitional, affecting, animate, independently exist, ...}
Lexical properties of argument 2	{meal, repast, nutriment, nourishment, sustenance, ...}
Event-based properties of argument 2	{non-independently exist, affected, change, ...}

Figure 2.2: The argument structure frame extracted from the verb usage *Sara ate lunch* in Figure 2.1. Reproduced from Alishahi and Stevenson (2010, p. 61, Figure 3)

These input pairs are generated from a hand-crafted input corpus based on statistical properties of child-directed speech. To simulate missing data and noise, two out of every five input pairs are modified by having a feature removed. One of these modified input pairs thus serves to simulate incomplete data, and the other is altered to simulate noise. This is done by replacing the missing feature with the most probable value for this feature, as predicted by the model at that point in training. This mirrors situations where the child relies on inferred knowledge to fill in missing information.

The addition of noise to the model is particularly interesting for simulating bootstrapping processes, since an important aspect of the bootstrapping hypothesis holds that children exploit both linguistic and extralinguistic cues to make inferences about meaning. However, even these noisified input pairs still require that a considerable amount of information be available to the child. In particular, the child must possess knowledge about complex lexical semantic features, and must still have access to substantial observational information. Since only one feature is removed, this simulates situations where the child has access to extralinguistic input, but the full 'meaning' of the scene is not available to her. Indeed, the main shortcoming of the model is that, like Allen's (1997) model, it assumes that the child already has an extensive understanding of various semantic and conceptual properties of both words and events, and that the child always has access to extralinguistic input.

The input corpus is used to generate verb usages like the one illustrated in Figure 2.1. The model incrementally processes each verb usage to extract an *argument structure frame*, illustrated in Figure 2.2, reproduced from Alishahi and Stevenson (2010). Similar frames are then grouped together based on an unsupervised Bayesian clustering process.

Alishahi and Stevenson (2010) present a number of experiments with different language tasks, showing that the model has learned semantic roles as well as constructions, and can extend this knowledge to utterances containing novel words. After training, the model is presented with a novel verb in an ambiguous context, and asked to select the most likely interpretation amongst different frames. For example, when presented with the novel verb *blick* in a transitive

structure (e.g. *She blick her*), the model prefers an interpretation where *she* is an Agent over an interpretation where it would be an Undergoer, which shows that it has successfully interpreted the novel word.

2.3.4 General discussion

The main objective of previous simulations of bootstrapping has been to show that bootstrapping mechanisms can emerge from exposure to language, with limited prior assumptions about innate knowledge. The models discussed in this section show that linking rules can be learned from experience (Desai, 2002; Allen, 1997) and that even the semantic roles themselves need not be innately given (Alishahi and Stevenson, 2010).[2] Nevertheless, like the experimental work on child language acquisition, these models only demonstrate that bootstrapping processes can play a role in sentence comprehension. They do not illustrate that bootstrapping can work as a procedure for *learning* novel words over time.

Moreover, these models rely on training procedures where both utterances and scenes are always available to the learner. It is only during *testing* that novel words are presented to the model without concurrent extralinguistic input. Yet, the syntactic bootstrapping hypothesis holds that children can exploit linguistic cues even in the absence of an extralinguistic scene. For concrete nouns and verbs, this is perhaps not crucial, since the child could eventually be exposed to enough real-world situations to allow him to learn the words without relying on linguistic structure. However, for words with more abstract meanings, most of the learning must occur without direct tangible evidence. Indeed, syntactic bootstrapping is particularly important in cases where no extralinguistic scene is available.

Previous models treat bootstrapping as a procedure that applies after the child has acquired a significant amount of knowledge. For example, the models of Allen (1997) and Alishahi and Stevenson (2010) model procedures that come into play only after the child has acquired complex representations of individual lexical items, because words are represented as bundles of features garnered from the WordNet database. While the syntactic bootstrapping hypothesis does assume that children can build a small vocabulary from cross-situational observation alone, it does not presuppose such extensive knowledge.[3] It would be desirable to investigate whether children at an earlier stage of learning can also take advantage of syntactic cues to infer (more general) components of word meaning.

A complete account of bootstrapping must involve semi-supervised training, where the model is presented with both labelled and unlabelled examples in the learning phase. In the latter case, the model would need to rely on its inferences about the properties of novel words to progressively acquire the meaning of these words. This would simulate situations where real-world observation provides

[2] See also Morris et al. (2000) for a connectionist model that learns linking rules between syntax and semantics, as well as the abstract notion of 'subject'.

[3] Indeed, Alishahi and Stevenson (2010) do not claim that this assumption is psychologically plausible.

little or no information to the learner, as in the experiments of Yuan and Fisher (2009) and Scott and Fisher (2009). It would also allow the investigation of whether bootstrapping can allow young learners to learn new vocabulary even while they are still in the early stages of learning the syntax of their language. Finally, the training data should be sufficiently rich so that the model is required to keep track of novel words over distinct syntactic frames in order to learn the full meaning of novel words (as given by the grammar).

In the next chapter, I present a connectionist model designed to fulfill these criteria. The objective is to investigate whether syntactic bootstrapping can lead to learning new vocabulary even before the child has learned complex lexical semantics or acquired a full grasp of syntax.

Chapter 3

Methodology

Bootstrapping requires discovering and exploiting correspondences between syntax and semantics. In this thesis, this process is modelled by a neural network trained to associate "utterances" and "scenes". The network learns to generate a semantic representation for input sentences, which roughly corresponds to determining 'who did what to whom', as in Desai (2002, 2007) and Morris et al. (2000), among others. More specifically, for a given input sentence (i.e. a sequence of words), the network must identify the participants (the Agent and Undergoer), the activity or event described by the predicate (the verb), and the causality of the event. Example utterances with their corresponding scene representations are given in (1).

(1) a. *Cats sleep*
Agent: none
Undergoer: cats
Event: sleep
Causality: non-causal

b. *The kids make sand-castles*
Agent: kids
Undergoer: sand-castles
Event: make
Causality: causal

The model simulates language acquisition at a stage after the child has learned to segment speech. Inputs to the model therefore consist of sentences presented one word at a time. The main assumptions built into the model are that it processes data incrementally and sequentially, i.e. that it processes sentences word-by-word, and that it attempts to discover correspondences between syntax and semantics – mappings from 'utterances' (input sequences) to 'scenes' (targets/outputs).

No formal linguistic structures or features are provided to the model. In particular, knowledge of linking rules, lexical categories, or syntax is *not* inherent to the network, and must be learned. For simplicity, no morphosyntactic or prosodic information is represented in the data given to the model. For consistency, nouns of the language will be referred to in the plural form throughout the thesis, and verbs will be conjugated in the present tense. However, neither number nor tense features are encoded in the language.

This chapter is organized as follows. In section 3.1, I describe the miniature language learned by the model. Section 3.2 discusses the procedure used to generate training and test sentences. Section 3.3 outlines the network architecture, with special attention to the encoding of the inputs and outputs. Finally, in sections 3.4 and 3.5, I present the training procedure and describe how the network's performance is evaluated.

3.1 The language

As in previous work on syntactic bootstrapping, the model is trained on input-output pairs representing "utterances" and "scenes", respectively. The artificial grammar used to generate training and test examples must therefore specify both the syntactic forms of the language as well as their semantic correlates.

The artificial language used to train the model is designed to contain linguistic structures found in natural language, but is not intended to be a realistic representation of the linguistic input in a child's environment. The language includes both 'content' words (nouns and verbs) and 'function' words (the preposition *to*, and the determiner *the*, which optionally modifies nouns). There are two kinds of nouns: *animate* and *inanimate*, and three types of verbs: optionally transitive, strictly transitive, and strictly intransitive. Verbs are further divided into different classes, depending on the type of argument(s) they take, the semantic roles they assign to their arguments, and whether they denote a causal or non-causal activity or event.

In total, the language includes 6 verb classes, each imposing different requirements on the lexical category or semantic role of their argument(s). In what follows, I first describe the syntactic structures licensed by each verb class, then discuss the corresponding semantic properties of each class.

3.1.1 Syntax

There are 2 classes of transitivity-alternating verbs, denoted V_{move} and V_{draw}.[1] In the transitive frame, verbs of both classes obligatorily take an animate subject, but direct objects can be either animate or inanimate. In the intransitive frame, verbs of the V_{move} class license only inanimate subjects, while the V_{draw} class licenses only animate arguments.

[1] Each verb class is referred to with a subscript indicating a typical verb of the class. For example, V_{move} verbs belong to the same class as *move*.

Table 3.1: Syntactic structure of each sentence type, as defined by each verb class. Subscripts indicate restrictions (if any) imposed on verb arguments (e.g. 'NP_{anim}' indicates that the argument must be *animate*, while 'NP' indicates that no restriction is imposed on the argument).

Type		Sentence structure	Examples
1	Transitive	NP_{anim} V_{move} NP	The kids move cats/toys
	Intransitive	NP_{inanim} V_{move}	The toys move
2	Transitive	NP_{anim} V_{draw} NP	Kids draw cats/toys
	Intransitive	NP_{anim} V_{draw}	Kids draw
3		NP_{anim} V_{make} NP_{inanim}	Girls make sand-castles
4		NP_{inanim} V_{hit} NP	Frisbees hit the kids/walls
5		NP_{anim} V_{run}	The dogs run
6		NP V_{fall}	The girls/toys fall

- V_{move} class: verbs taking only animate subjects, but either animate or inanimate objects. In the intransitive frame, the subject must be *inanimate*.
  Examples: *The boys drop the balls/the cats. The balls drop.*

- V_{draw} class: verbs taking only animate subjects, but either animate or inanimate objects. In the intransitive frame, the subject must be *animate*.
  Examples: *The kids draw pictures/cats. The kids draw.*

There are 2 classes of strictly transitive verbs:

- V_{make} class: verbs taking only animate subjects and only inanimate objects.
  Example: *The girls make sand-castles.*

- V_{hit} class: verbs taking only inanimate subjects, but either animate or inanimate objects.
  Examples: *The frisbees hit the girls. The frisbees hit the walls.*

There are 2 classes of strictly intransitive verbs:

- V_{run} take only animate subjects.
  Example: *The dogs run.*

- V_{fall} take either an animate or inanimate subject.
  Examples: *The girls arrive. The toys fall.*

Table 3.1 summarizes the syntactic structures licensed by each verb class.

Sentences of the language can be generated by a formal probabilistic context-free grammar (PCFG). Table 3.2 shows the grammar used to generate training and test sentences.

3.1.2 Semantics

Verb classes also define the semantics of sentences licensed by the grammar. Verb classes determine the semantic role assigned to arguments, and the causality of the event, as explained shortly.

Only two general semantic roles are defined: Agent and Undergoer. The *Agent* role is used to encode a general concept involving the doer, causer, or source of the activity described by the verb. The *Undergoer* role is assigned to the entity affected by the action. For instance, in the sentence *'John threw the ball'*, *John* would be the Agent, and *ball* would be the Undergoer.

The transitivity-alternating verbs defined in the miniature language assign different semantic roles to their subject, much like the *causative* and *contact* verbs discussed in Chapter 2, Section 2.2. These verb classes also differ in their causality, as outlined below. Similarly, strictly intransitive verbs vary in the role assigned to the subject, and the causality of the event. Strictly transitive verbs are all causal, and assign Agent roles to their subjects and Undergoer roles to their objects. The semantics of each verb class can be summarized as follows:

- V_{move} class (causal):
  Transitive frame: Agent V_{causal} Undergoer
  Example: *The boys (Agent) bounce the balls (Undergoer)*

  Intransitive frame: Undergoer V_{causal}
  Example: *The balls (Undergoer) bounce*

- V_{draw} class (non-causal):
  Transitive frame: Agent $V_{non-causal}$ Undergoer
  Example: *The boys (Agent) draw the balls (Undergoer)*

  Intransitive frame: Agent $V_{non-causal}$
  Example: *The boys (Agent) draw*

Both classes of strictly transitive verbs are causal. They assign an Agent role to the subject, and an Undergoer role to the object.

- V_{make} class (causal):
  Agent V_{causal} Undergoer
  Example: *The girls (Agent) make sand-castles (Undergoer)*

- V_{hit} class (causal):
  Agent V_{causal} Undergoer
  Example: *The frisbees (Agent) hit the girls (Undergoer).*

Table 3.2: PCFG of the language. Variable k denotes the kind of NP (animate (*anim*) or inanimate (*inanim*)). Where the probabilities of different production rules are not equal, they are given in parentheses.[a]

S	$\rightarrow$	NP_{anim} V_{move} NP_{k}
S	$\rightarrow$	NP_{anim} V_{draw} NP_{k}
S	$\rightarrow$	NP_{anim} V_{make} NP_{inanim}
S	$\rightarrow$	NP_{inanim} V_{hit} NP_{k}
S	$\rightarrow$	NP_{k} V_{fall}
S	$\rightarrow$	NP_{anim} V_{run}
NP_{k}	$\rightarrow$	the N_{k} (0.6) | N_{k} (0.4)
N_{k}	$\rightarrow$	N_{anim} | N_{inanim}
N_{anim}	$\rightarrow$	boys | girls | kids | cats ...
N_{inanim}	$\rightarrow$	frisbees | toys | dolls | books ...
V_{move}	$\rightarrow$	move | roll | bounce ...
V_{draw}	$\rightarrow$	draw | sketch | dust ...
V_{make}	$\rightarrow$	make | build | catch ...
V_{hit}	$\rightarrow$	hit | strike | break ...
V_{run}	$\rightarrow$	run | dance | sing ...
V_{fall}	$\rightarrow$	fall | arrive | sleep ...

[a] An additional well-formedness constraint, not specified in the PCFG, is imposed on utterances: in a given sentence, the same noun is not allowed to appear in two different syntactic positions (e.g. Subject/Object). For example, there are no sentences like *the girls draw the girls*. This constraint reflects the fact that in natural language, such sentences are only grammatical if the two noun phrases refer to distinct entities (i.e. if there are two different groups of girls). Otherwise, reflexive or reciprocal pronouns would be required (at least in English), e.g. *the girls draw themselves, the girls draw each other*. Moreover, if the same noun could appear as both subject and object in a given sentence, the semantic representations would have to allow a referent to receive two semantic roles (e.g. *girls* would be both Agent and Undergoer). This would violates a robust cross-linguistic generalization: an NP cannot simultaneously fill two semantic roles.
Although this rule was omitted from the PCFG to improve readability, in the remainder of this thesis, any reference to the *grammar* or *language* includes this rule.

Table 3.3: Semantic structure of each sentence type, as defined by each verb class.

Type		Sentence structure	Causality
1	Transitive	Agent V_{move} Undergoer The kids move balls/cats	Causal
	Intransitive	Undergoer V_{move} The balls move	Causal
2	Transitive	Agent V_{draw} Undergoer Kids draw cats/toys	Non-causal
	Intransitive	Agent V_{draw} Kids draw	Non-causal
3		Agent V_{make} Undergoer The girls build sand-castles	Causal
4		Agent V_{hit} Undergoer The frisbees hit the kids/walls	Causal
5		Agent V_{run} The dogs run	Causal
6		Undergoer V_{fall} The toys/girls fall	Non-causal

Strictly intransitive verbs differ in their causality and in the role assigned to the subject:

- V_{run} class (causal):
  Agent V_{causal}
  Example: *The dogs (Agent) run*

- V_{fall} (non-causal):
  Undergoer $V_{non-causal}$
  Example: *The girls (Undergoer) arrive*

Table 3.3 summarizes the semantic structures licensed by each verb class.

Together, the syntax and semantics fully determine the structures licensed by the grammar. Table 3.4 summarizes the syntax and semantics of the different constructions of the language.

The constructions licensed by this grammar were chosen because they parallel some of those frequently discussed in the literature on bootstrapping. Ideally, the language would include not only a larger number of lexical categories and constructions, but would also impose more realistic requirements on the arguments of particular verbs. Nevertheless, given that the model's task is to learn correspondences between form and meaning, the language must be sufficiently rich in both its syntax and semantics, without becoming overly complex. In

Table 3.4: Syntactic and semantic structure of each sentence type, as defined by each verb class. Subscripts indicate restrictions (if any) imposed on verb arguments (e.g. 'NP_{anim}' indicates that the argument must be *animate*).

Type		Sentence structure	Causality
1	Transitive	NP_{anim} (Agent) V_{move} NP (Undergoer) The kids move cats/toys	Causal
	Intransitive	NP_{anim} (Undergoer) V_{move} The toys move	Causal
2	Transitive	NP_{anim} (Agent) V_{draw} NP (Undergoer) Kids draw cats/toys	Non-causal
	Intransitive	NP_{anim} (Agent) V_{draw} NP (Undergoer) Kids draw	Non-causal
3		NP_{anim} (Agent) V_{make} NP_{inanim} (Undergoer) The girls build sand-castles	Causal
4		NP_{inanim} (Agent) V_{hit} NP (Undergoer) Frisbees hit kids/walls	Causal
6		NP_{anim} (Agent) V_{run} The dogs run	Causal
7		NP (Undergoer) V_{fall} The girls/toys fall	Non-causal

particular, the grammar is designed so that the model cannot rely *solely* on the syntactic position of words in a sentence to successfully acquire the language. For example, although most – but not all – transitive verbs are *causal*, and most intransitives are non-causal, the learner cannot simply associate (non)causality with (in)transitivity.

3.2 Data sets

In previous work on bootstrapping, models were presented with 'novel' words only during testing, to evaluate the model's ability to use linguistic context to infer the meaning of novel words. The main contribution of the present thesis is to investigate whether inferences about the meaning of words can also be used *during* the learning process, as well as *after* the language has been acquired.

3.2.1 Labelled and unlabelled data

To simulate the use of bootstrapping processes for vocabulary acquisition, a partly unsupervised training procedure is adopted, with the network trained on both *labelled* and *unlabelled* data.

Labelled data consist of input-output pairs, where the output is the correct target according to the syntax and semantics of the language, the *groundtruth*. Training the network on labelled examples is therefore referred to as 'supervised' training. The use of labelled examples corresponds to situations where the learner is exposed to both an extralinguistic scene and linguistic input. These training data therefore include an input sentence (i.e. a sequence of words) and a semantic representation of that sentence, identifying the event, its causality, and the participants in the event. However, the learner is given no direct information about which words correspond to which semantic role – the network must learn these relations over time.

Unlabelled data consist of input sentences presented to the model *without* the corresponding groundtruth. Training the network on these examples is referred to as 'self-supervised training' (or simply 'selftraining'), because the network must generate its own target, and use it to train itself. The use of unlabelled examples simulates bootstrapping in cases where the learner cannot rely on extralinguistic context to interpret an utterance. This could represent situations where there is no congruent scene, or situations where a novel word is abstract, and cannot be directly linked to the real world.

For training purposes, the only distinction between labelled and unlabelled data is that with labelled examples, the target presented to the network is fully determined by the syntactic and semantic constraints imposed on arguments, as described above, whereas with unlabelled examples, the network generates its own target.

In and of itself, the self-supervised training algorithm is not sufficient to study the use of inferences to learn novel words. This is because even if some data are unlabelled, in principle, any given word would eventually appear in

Table 3.5: Distribution of *trained/selftrained* words in (un)labelled training examples

	Trained	Selftrained
Labelled	2553	0
Unlabelled	451	2201

enough labelled examples to allow the network to discover their meaning based on extralinguistic information.

Therefore, to determine to what extent the network can learn words without the support of extralinguistic content, some words in the grammar are designated as 'selftrained', and restricted to unlabelled examples. This means that all information about these words is inferred based on the model's (evolving) knowledge of the grammar, and on the syntactic context(s) where the words appear over multiple exposures.

For example, if 'wugs' is designated as a selftrained animate noun, this word will only appear in unlabelled examples. When presented with a sentence containing this word, the network must therefore make inferences about the word's meaning based on context. Ideally, by seeing multiple instances of this word in different syntactic frames, the network will eventually build a representation of 'wugs' which is similar to that of animate nouns presented in *labelled* examples.

Words allowed in labelled examples are referred to as *trained*, but it is important to realize that these words must still be *learned* by the network. The only distinction between *trained* and *selftrained* words is that the latter can only appear in unlabelled examples, whereas *trained* words can appear in either labelled or unlabelled examples.[2]

Since any sentence containing a selftrained word must be part of the unlabelled data, training the model on sentences with selftrained words simulates situations where no extralinguistic cues are available to the learner.

Table 3.5 gives the distribution of labelled and unlabelled sentences in the training data. In brief: selftrained words are restricted to unlabelled examples, while trained words are allowed in both labelled and unlabelled examples. Labelled examples can only contain trained words.

In what follows, I describe how the training and test data are generated.

3.2.2 Training sentences

There are 54 trained words and 16 selftrained words in the language. Trained words include 8 nouns of each type (animate/inanimate), 6 verbs of each class, and the function words *the* and *to*. For each lexical (sub)category, there are 2 selftrained words, i.e. 2 selftrained animate nouns, 2 selftrained inanimate

[2]The sets of labelled and unlabelled examples are disjoint; no sentence is used for *both* supervised and unsupervised training.

Table 3.6: Word categories and examples. There are 8 trained nouns in each category (animate/inanimate) and 2 selftrained nouns in each category. For each verb class, there are 6 trained verbs and 2 selftrained verbs.

Word category	Trained word example	Selftrained word example
Animate noun	cats	wugs
Inanimate noun	toys	daxes
V_{move} verb	bounce	glorp
V_{draw} verb	sketch	blick
V_{make} verb	build	sib
V_{hit} verb	strike	krad
V_{run} verb	dance	lorp
V_{fall} verb	arrive	pilk

nouns, and 2 selftrained verbs of each class. Table 3.6 provides examples of trained and selftrained words of each category.

All sentences licensed by the grammar are initially generated, but certain sentences are then removed to create the final set of training data.

First, sentences with more than one selftrained word are removed from the training data, so in any sentence, only one word may be a selftrained word. For example, if 'wugs' is a selftrained animate noun, and 'daxes' is a selftrained inanimate noun, the sentences *cats see wugs, cats see daxes, wugs see cats, ... can* be used as input data. However, sentences such as *wugs move daxes, wugs draw daxes, daxes hit wugs, ...* are excluded from the training data, even though they are grammatical. This is done to avoid overwhelming the model with sentences where it has too little information to make inferences based only on its developing knowledge of grammar and on syntactic context.

Second, specific sentences are extracted to make it possible to generate test examples where trained words appear in highly novel combinations. More precisely, the objective is to create test sentences where adjacent words never appear together during training. For example, if the combinations *girls draw X* and *Y draw toys* are excluded from the training data, then the combination *girls draw toys* is highly novel.

The only way the network can perform well when tested on such a novel combination is by generalizing over the use of individual verbs, and relying on its knowledge of both the nouns and the syntactic positions they appear in. In most cases, neither the word nor the syntactic frame alone is sufficient to fully predict the meaning of the utterance.

Table 3.7 lists the combinations excluded from the training data. For each transitive verb class, restrictions are imposed on the arguments of one of the verbs. For example, the verb *draw* is not allowed to appear with *girls* or *toys*, so the combinations *girls draw X* and *Y draw toys* are not seen during training.

Table 3.7: Inputs removed from training set to create sentences with trained words presented in novel combinations.

Restriction		Corresponding novel test sentences
boys & *frisbees*	do not appear with *drop*	*boys drop frisbees* *frisbees drop*
girls & *toys*	do not appear with *draw*	*girls draw toys* *girls draw*
kids & *dolls*	do not appear with *build*	*kids build dolls*
cats & *rocks*	do not appear with *strike*	*rocks strike cats*

This allows the creation of the corresponding highly novel combinations *girls draw* and *girls draw toys*.[3]

After these sentences have been removed from the training data, 25% of sentences with selftrained words and 10% of the remaining *labelled* sentences are randomly extracted to serve as test examples. Finally, 15% of the remaining labelled sentences are randomly selected to be used as *unlabelled* examples. This creates a training set with sentences consisting of only *trained* words to be used for self-supervised training. The will allow a comparison between selftraining on examples with only *trained* words, and selftraining on examples that include *selftrained* words.

The procedure for generating training data is summarized below.

1. Generate all sentences licensed by the grammar.

2. Remove all sentences with more than one selftrained word.
   This results in two sets of training data:
   Set A: sentences with only trained words
   Set B: sentences where exactly one word in each sentence is a selftrained word, and the rest are trained words

3. Systematically extract specific sentences from both resulting sets (i.e. from $A \cup B$). These examples serve as test items with highly novel combinations of words; two test sets are created: one with only trained words, one with sentences composed of trained words plus one selftrained word.

4. Randomly remove 25% of examples from set B for testing. These examples (sentences with exactly one selftrained word) are used to test generalization with selftrained words.

[3]Restrictions are not applied to strictly intransitive verbs (i.e. V_{run} and V_{fall}) because there are few occurrences of these verbs in the training set, due to the fact that intransitive verbs take only one argument and therefore give rise to fewer combinations than transitive verbs.

5. Randomly remove 10% of examples from set A for testing (examples with only *trained* words). These examples serve as baseline generalization examples with only *trained* words.

6. Randomly remove 15% of examples remaining in set A to be part of unlabelled data

3.2.3 Test sentences

Test sentences can be broken down into different types, depending on their degree of novelty (relative to examples seen by the network during training), and the type of words they contain (trained/selftrained).

There are two sets of test sentences formed with only *trained* words. First, there are the sentences randomly extracted from the initial training set and reserved for testing; these serve as a baseline generalization test since they can, in principle, be very similar to labelled examples presented to the network.

Second, there are the sentences specifically excluded from training to test for generalization to highly novel combinations of words (see Table 3.7 and the discussion in section 3.2.2).

Similarly, there are two different sets of test sentences with *selftrained* words. One set simply consists of the examples randomly extracted from the training data. The other includes sentences with highly novel combinations of *trained* words combined with a selftrained word. For example, *girls draw toys* is a highly novel test item (which means that combinations *girls draw X* and *Y draw toys* were not seen during training). Thus, there are similar examples with selftrained words, e.g. *girls draw daxes* and *wugs draw toys*. These examples are not highly novel, however, because during training, the selftrained word could have occurred together with the verb. In particular, even if the combination *Y draw toys* is novel, the combinations *wugs draw* and *draw daxes* could have been seen in selftraining.

In addition to these test data, the model is also tested on entirely novel *words*, unseen during any training phase. To this end, a set of *untrained* words is also included in the grammar, and different test sets consisting of combinations of trained and untrained words can be created to compare performance on selftrained words and untrained words. This will be discussed in chapter 4.

3.3 Network Architecture and processing

The model is a simple recurrent network (SRN; Elman (1990)). Given that bootstrapping requires discovering and exploiting correspondences between syntax and semantics, the word-prediction task commonly associated with SRNs is not appropriate. Rather, the network is trained on a sentence interpretation task, which involves associating "utterances" with "scenes", as described earlier. The network architecture is shown in Figure 3.1.

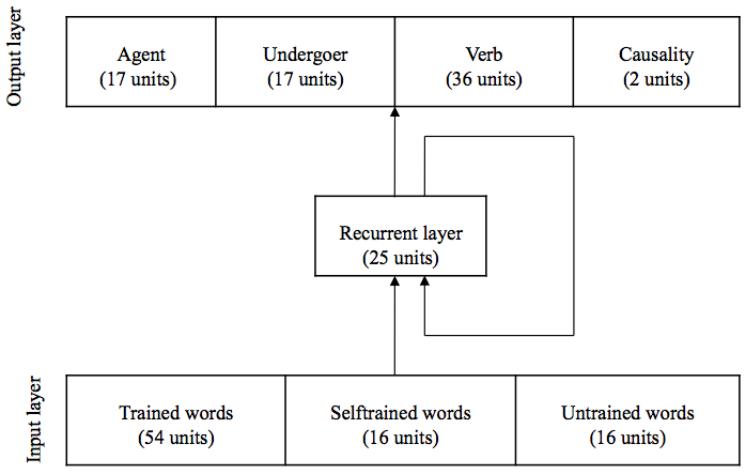

Figure 3.1: Network architecture

3.3.1 Input and output

Words are encoded in the input layer using localist representations, where each unit corresponds to an individual word. There are 86 input units in total, each corresponding to one of the 54 trained words, 16 selftrained words, and 16 untrained words that are part of the model's vocabulary. Untrained words are represented in the input layer, but since these words do not appear in any *(self)training* examples, their corresponding input units are never activated during training. The input weights for the units representing untrained words are a random permutation of the input weights corresponding to selftrained words after training.

The network's output layer encodes the semantics of the sentences presented to the network, as explained in section 3.2. Output units are divided into 4 slots, each encoding a particular aspect of the event representation.

The first two slots correspond to semantic roles: the Agent and Undergoer, respectively. For each *trained* noun in the input layer, there is a corresponding 'concept' unit in the output. In addition, for each of the two semantic roles, one output unit represents the possibility that there is no argument corresponding to this role in the input sentence. For example, if the input is *cats sleep*, where *cats* is assigned the role *Undergoer*, the correct output unit for the Agent slot, as specified by the groundtruth, is the unit corresponding to 'no Agent'. The groundtruth for the Agent and Undergoer slots for *'cats sleep'* is illustrated in Figure 3.2. The symbol '$\emptyset$' denotes the 'empty' node, indicating that there is no filler for the corresponding semantic role.

Notice that inputs and outputs are *not* in a one-to-one correspondence. For instance, the function words *to* and *the* are represented in the input layer, but have no corresponding outputs. In addition, each input unit representing

Agent Slot

Animate concepts					Inanimate concepts					None
boys	girls	kids	cats	…	toys	dolls	books	balls	…	∅
										■

Undergoer Slot

Animate concepts					Inanimate concepts					None
boys	girls	kids	cats	…	toys	dolls	books	balls	…	∅
			■							

Figure 3.2: Groundtruth for the Agent and Undergoer slots for the sentence *cats sleep*; darkened boxes indicate target units.

a *trained* noun is related to two distinct output units (one for each possible semantic role). Therefore, even if a noun appears in the same position in two very similar sentences, the semantic representation corresponding to the noun need not be identical. In (2-a), for example, the 'cats' concept in the Undergoer slot would be the target, while in (2-b), the 'cats' concept in the Agent slot would be the target.

(2) a. cats sleep (cats = Undergoer)
b. cats run (cats = Agent)

The third slot represents the event description; there *is* a one-to-one correspondence between the units in this slot and the *trained* verbs in the input layer. All sentences presented to the network are complete utterances, and therefore contain a verb. Moreover, since there are no complex sentences in the grammar (i.e. no sentences with a subordinate clause), each sentence contains only one verb.

The last slot encodes the causality of the event, with the one unit activated for causal events, and the other unit activated for non-causal events.

There are a total of 72 output units. In each of the Agent and Undergoer slots, there are 17 nodes (8 representing animate concepts, 8 representing inanimate concepts, and one representing 'empty'). The verb slot contains 36 nodes, each corresponding to an individual verb, and the causality slot includes 2 nodes, one for causal events, the other for non-causal events.

To illustrate the mapping from utterances to scenes, the groundtruth for *cats sleep* and *boys read books* are shown in Figure 3.3 and Figure 3.4, respectively.

Selftrained and untrained words are not represented as individual concepts in the output layer. Therefore, when presented with selftrained or untrained words, the network must activate the output units that most closely fit the inferred 'meaning' of these words. This meaning will inevitably be related (correctly or not) to some output unit(s) corresponding to the concepts encoded

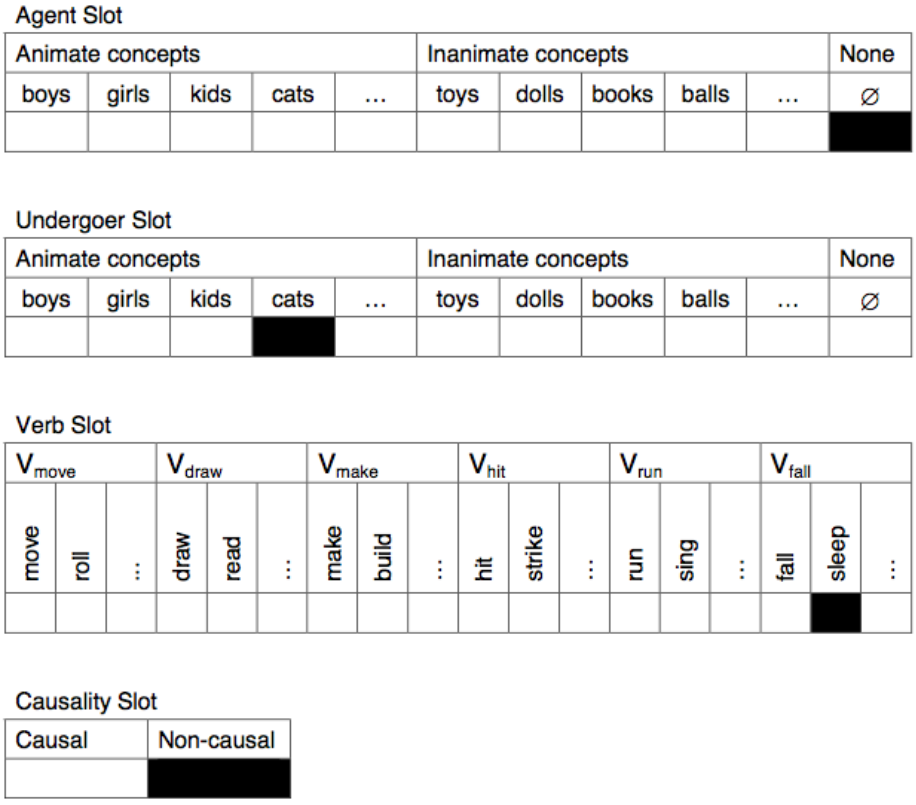

Figure 3.3: Groundtruth for the sentence *cats sleep*; darkened boxes indicate target units.

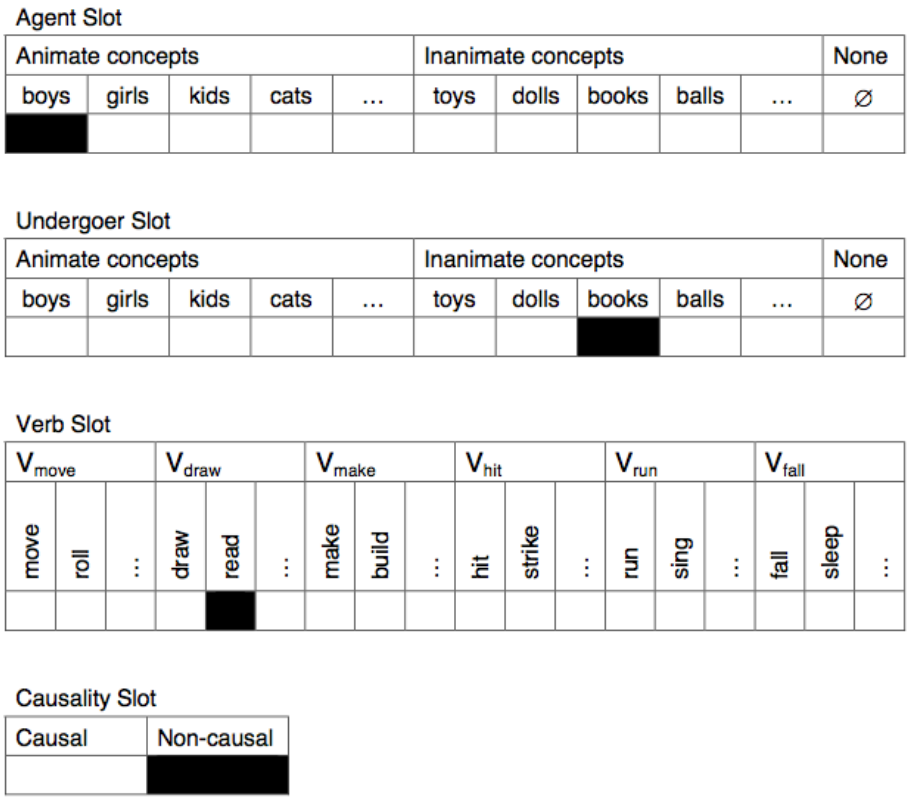

Figure 3.4: Groundtruth for the sentence *boys read books*; darkened boxes indicate target units.

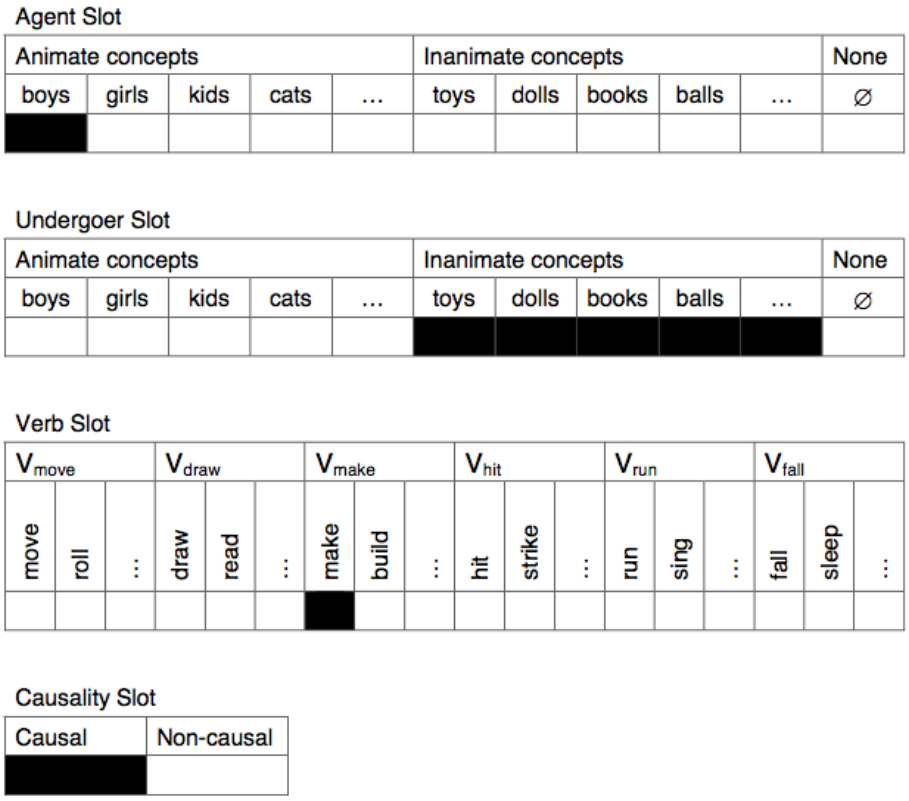

Figure 3.5: Groundtruth for the sentence *boys make daxes*; darkened boxes indicate target units.

by trained content words in the language. When the network is tested on sentences containing a selftrained or untrained word, we can therefore determine whether it has made appropriate inferences based on which output units are activated. For example, if 'daxes' is a selftrained inanimate noun presented as an Undergoer, the network would ideally activate an output unit corresponding to an inanimate trained noun in the Undergoer slot. Since selecting any of the appropriate output units would be correct, the groundtruth units in this case would range over all Undergoer outputs corresponding to inanimate nouns. This is illustrated in Figure 3.5.

3.3.2 SRN processing

Sentences are presented to the network one word at a time by activating the unit corresponding to the word currently being processed. Words are encoded using localist representations, so at each time-step t, the input to the network is encoded by the vector $a_{in}(t)$, whose length is equal to the number of input

units. The vector presented at time t contains zeros in all entries except for the entry corresponding to the input word at time t.

The values of the recurrent and output units are calculated according to

$$\begin{aligned} a_{rec}(t) &= f(W_{in}a_{in}(t) + W_{rec}a_{rec}(t-1) + b_{rec}) \\ a_{out}(t) &= f_{out}(W_{out}a_{rec}(t) + b_{out}) \end{aligned} \tag{3.1}$$

where $a_{rec}(t), a_{out}(t)$ are the activation vectors of the recurrent and output layers, respectively, and $b_{rec}(t)$, $b_{out}(t)$ are the corresponding bias vectors; W_{in}, W_{rec}, $W_{out}(t)$ are the matrices containing the input, recurrent, and output weights, respectively; f is the logistic activation function, used to compute the activations of the recurrent units:

$$a_{rec,i} = f(x_i) = \frac{1}{1 + e^{-x_i}} \tag{3.2}$$

where x_i is the activation going into unit i.

There are 25 recurrent units in the SRN. The activations of the recurrent layer are reset after each full sentence has been processed, such that each sentence has its own independent semantic representation. This is done because in the input data, the semantic representation of a given sentence is not related to the previous sentence(s); it is fully determined by the utterance itself.

Cross-entropy was used as the error function, with softmax activation applied to the output. The softmax activation function results in outputs that sum to 1, and each is non-negative, so these outputs can be interpreted as probabilities. Here, the network's task is to identify participants and events described by the input sequence, so the relevant probabilities must be taken over each *slot*, rather than over the full output. The softmax activation function is therefore applied over each slot:

$$a_{out,i} = f_{out}(x_i) = \frac{e^{x_i}}{\sum_{j \in slot_i} e^{x_j}} \tag{3.3}$$

where $slot_i$ is the set of units in the slot with unit i.

After applying this function, the sum over the output of the units in a given slot equals 1, so $a_{out,i}$ can be viewed as a probability estimate. For a given semantic role, $a_{out,i}$ can be interpreted as the network's estimated probability that i is the filler for the corresponding role. Similarly, for verbs, each output activation in the verb slot represents the network's probability estimate that i is the event described by the input sentence. Finally, each activation in the causality slot reflects the network's estimate about whether or not the event is causal.

3.4 Training

To train the network, both labelled and unlabelled examples are pooled together in a single training set, and randomly shuffled before each training epoch.[4]

[4] An epoch is a single presentation of the full set of training data.

During training, the network is provided with utterances (sequences of words) (the *input*), and the intended output (the *target*). As noted earlier, the only difference between labelled and unlabelled examples lies in how the target is created. With labelled data, the target given to the network is the *groundtruth*. With unlabelled data, the target is inferred by the network. In both cases, the network is trained (or trains itself) on a sentence and a target.

The target for a sentence is held constant during the presentation of the full sequence. As noted by Desai (2007), this makes the task very challenging, since the network must try to predict the entire scene as soon as the first word is processed. The task is also more realistic than presenting the semantic representation of a word only when that word is processed, because it means the network must learn which words are related to which aspect of the scene. It is also in line with the incremental nature of human sentence processing.

With unlabelled examples, the target is inferred based on the full sentence, because the entire context may be required to understand the full meaning of selftrained words.[5] Targets for unlabelled examples are obtained by initially presenting the utterance to the network without a target (as in testing), then 'exaggerating' the resulting output to create a target. This is done by first computing the network's activation for each slot, then finding the unit with the highest activation in each slot. Each of these 'winning' units is then set to '1', and all other units are set to '0'. This results in a vector with exactly 1 unit per slot set to '1', as in the groundtruth vectors. Each of the 'target' units corresponds to the network's 'guess' about the meaning of the given input.

No sentence contains more than one selftrained word, and in the ideal case, for each *trained* content word in the input, the target unit for the corresponding slot will be identical to the groundtruth unit for that slot. This is by no means guaranteed, however.

For selftrained words, there is no single 'correct' unit, so in the best-case scenario, the network will assign the highest activation to a content word of the same lexical (sub)category, e.g. if *wugs* is a selftrained animate noun in Agent position, the inferred target for the Agent slot will be one of the concepts corresponding to animate nouns.

The intuition behind this self-supervised training procedure is that over time, the network will see a given selftrained word in a variety of contexts; it should then be able to build on its experience with that word to make inferences about its meaning. Therefore, even if its initial 'guesses' are incorrect, as the network's knowledge of the grammar develops, it will make better inferences, and more often create targets that do not diverge (much) from the groundtruth. Given that the artificial language does not encode complex lexical semantics, only

[5] Alternatively, the target could be inferred as each word is processed, so that at each time-step, the network is presented with a new (inferred) target. However, if a sentence includes an unfamiliar word, the learner may not be able to make inferences about the word until the full sentence has been parsed. For example, hearing *the wugs* only indicates that this word is a noun, whereas *the wugs eat* suggests that it is an animate noun. To model the *acquisition* of words, it is therefore better to take the target inferred *after* the full sentence has been processed.

broad inferences can be made, e.g. the lexical category of the word, and its animacy value (for nouns) or class (for verbs).

The network was trained for 350 epochs, using the standard back-propagation algorithm (Rumelhart et al., 1986). Initial connection and bias weights were randomly generated, uniformly distributed between -0.15 and 0.15. The learning rate was 0.02.

3.5 Evaluation

To evaluate the model's performance on a given input sequence, the network's output after processing the complete sequence (which represents the network's interpretation of the entire sentence) is compared to the groundtruth. For sentences containing only trained words, the groundtruth and the target presented to the network during training are identical. The network's accuracy is the joint probability of correctly identifying each of the groundtruth units corresponding to the event representation (i.e. semantic roles, verb, causality).

Since the output units are divided into 4 slots, and there is exactly one target unit in each slot, the network's accuracy for a sentence is given by equation (3.4), where $p(\text{slot}_i)$ is the probability assigned to the correct unit in slot i, i.e. the network's activation of that unit.

$$accuracy = \prod_{i=1}^{4} p(\text{slot}_i) \tag{3.4}$$

For sentences containing selftrained words, the groundtruth is *not* identical to targets presented during training, since the training targets were generated by the network rather than based on the grammar. In addition, for a selftrained word, there is no output unit representing the concept encoded by the word. Therefore, the output activated by the network in the corresponding slot is considered correct if it is of the right class, as determined by the grammar. For instance, consider the following example, where 'daxes' is a selftrained word, classified as an inanimate noun.

(3) The boys make daxes.
The boys (Agent) make (Verb) daxes (Undergoer)

Since the Undergoer slot is filled by a selftrained word, the network's 'guess' for this slot is considered correct if it activates output units corresponding to inanimate nouns. More precisely, the network's performance for this slot is the sum of the probabilities assigned to the Undergoer units representing the inanimate nouns. For each of the *trained* words, the correct unit is the concept corresponding to the input word (e.g. 'boys', 'make'), as was illustrated in Figure 3.5, which shows the groundtruth for the sentence *boys make daxes*.

Performance is calculated using equation (3.4) as before, but with one difference: for a slot related to a selftrained word, $p(\text{slot}_i)$ is the *sum* over the

probabilities assigned to the appropriate units in slot i, i.e. the network's activations over a subset of units in that slot. In the preceding example, this would be the sum of the probabilities assigned to each inanimate concept in the Undergoer slot.

The resulting score represents the probability that the network has correctly interpreted every aspect of the full sentence, i.e. that it has accurately identified the participants and the nature of the event encoded by the input. The network's performance over a set of sentences is simply the mean of its performance on each sentence in the set.

The joint probably is a very strict performance measure, because the performance in a given slot can never be greater than the minimum probability over the slot. This means that even if the network accurately identifies the correct unit in every slot, its score for that sentence can still be very low. Consider the following example. Suppose that each incorrect output unit gets a very low probability of .01, so all the other probability mass is in the correct unit. This is very good performance, since incorrect units have such low probability. However, the accuracy measure in this case receives a low score of $(1 - .16) \times (1 - .16) \times (1 - .35) \times .99 = 0.4541$.

3.6 Summary

The model is trained in semi-supervised fashion, using labelled and unlabelled examples. Labelled examples correspond to situations where both linguistic and extralinguistic context are available to the learner. Unlabelled examples correspond to cases where there is no congruent scene, and the learner must extract all information from the utterance.

During training, the network relies on the knowledge it acquires about word meaning, syntax, and syntax-semantics correspondences. With labelled examples, the model also has access to information about the scene. With unlabelled examples, it can *only* rely on the syntactic context and what it has previously learned about words, syntax, and syntax-semantic mappings.

Some words in the grammar are designated as *selftrained* and are restricted to unlabelled examples. These words present the most interesting cases to study bootstrapping processes in the model, because information about the semantics of these words can only be inferred based on linguistic context and the evolving knowledge of the network.

Even when the model is presented with labelled examples, it must still learn the relations between syntax and semantics. In particular, the network must learn which words are mapped to which concepts, as well as how syntax relates to semantics. This is not built into the model. The semantic roles *are* predefined, however. Thus, the assumption that the child has some notion of general semantic roles such as *doer/causer* and *undergoer/affected object* is inherent to the model. These could either be innately given, or learned at an earlier stage.

In the next chapter, I present the results of simulations performed with the model.

Chapter 4

Results and Discussion

In this chapter I present results of experiments performed with the model. I first discuss the training process in section 4.1. In section 4.2, I present results related to the network's capacity to generalize and to display systematic behaviour. In section 4.3, I investigate the network's ability to rely on context to make inferences and learn vocabulary.

4.1 Training process

Three different networks were trained using the semi-supervised training procedure outlined in Chapter 3. For each network, a different set of initial random weights was generated, and different data sets were created.[1] All networks had the same architecture and were trained with a learning rate of 0.02 for 350 epochs. Training was stopped at this point because there was no longer any significant improvement in performance (for any network).

Recall from Chapter 3 that the model is trained on two types of examples: sentences containing only *trained* words, and sentences containing one *selftrained* word. In addition, training was only partly supervised: the network was presented with labelled and unlabelled data, and had to infer the target when given an unlabelled example. Trained words could appear in either labelled or unlabelled examples, while selftrained words were restricted to unlabelled examples.

The networks' performance on training and test data is shown in Figures 4.1-4.3. Each graph shows the network's accuracy on three different data sets: labelled training data, test examples consisting of only trained words, and test examples containing a selftrained word.

In all cases, performance on test sentences with only trained words is nearly equal to performance on labelled training data, which shows that the network can learn the grammar and the *trained* lexical items. For sentences containing selftrained words, performance initially improves, suggesting that

[1] The data sets differed only with respect to the items randomly extracted for testing, or, in the case of sentences with only trained words, for use as unlabelled examples.

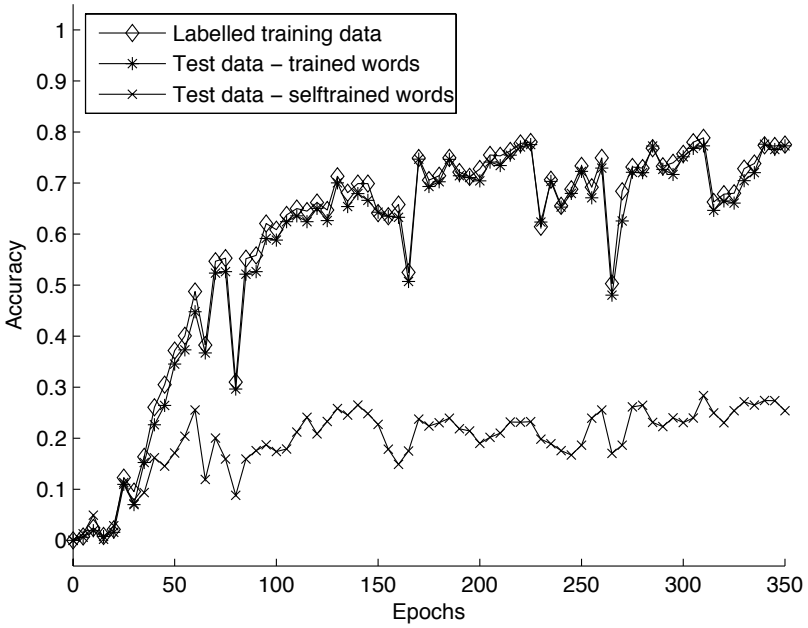

Figure 4.1: Accuracy of network 1 on training and test data.

self-supervised training does have a positive effect on vocabulary acquisition. However, given that performance on these data remains far below that of examples with only trained words, the model cannot be said to successfully acquire selftrained words.

As illustrated by the graph in Figure 4.4, performance on *unlabelled* training examples with only trained words is nearly identical to performance on *labelled* training data.[2] Given this, in the remainder of this chapter, I will not discuss these data further.

To evaluate the model, I take each network at epoch 350. All further results discussed in this chapter are based on the performance of these networks.

4.2 Generalization

The model's ability to generalize and display systematic behaviour is evaluated by testing its performance on sentences containing only *trained* words. Recall that there are two sets of test sentences with trained words. First, there are the sentences randomly extracted from the initial training set and reserved for

[2]Only the graph for network 1 is plotted, but the graphs for networks 2 and 3 show a similar pattern.

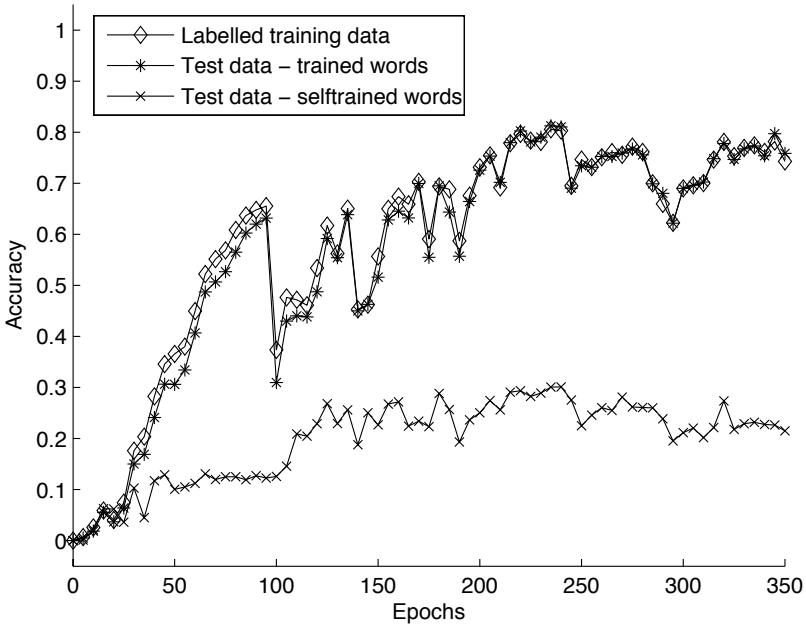

Figure 4.2: Accuracy of network 2 on training and test data.

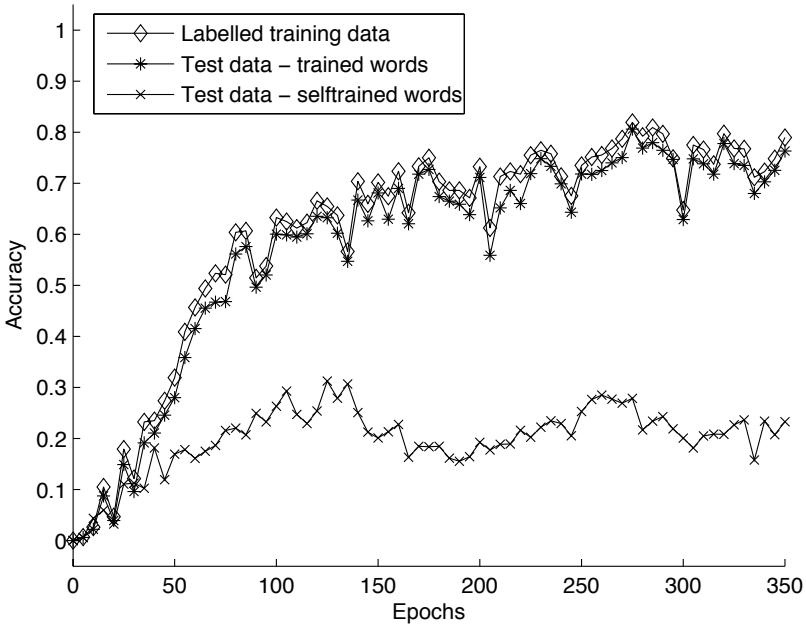

Figure 4.3: Accuracy of network 3 on training and test data.

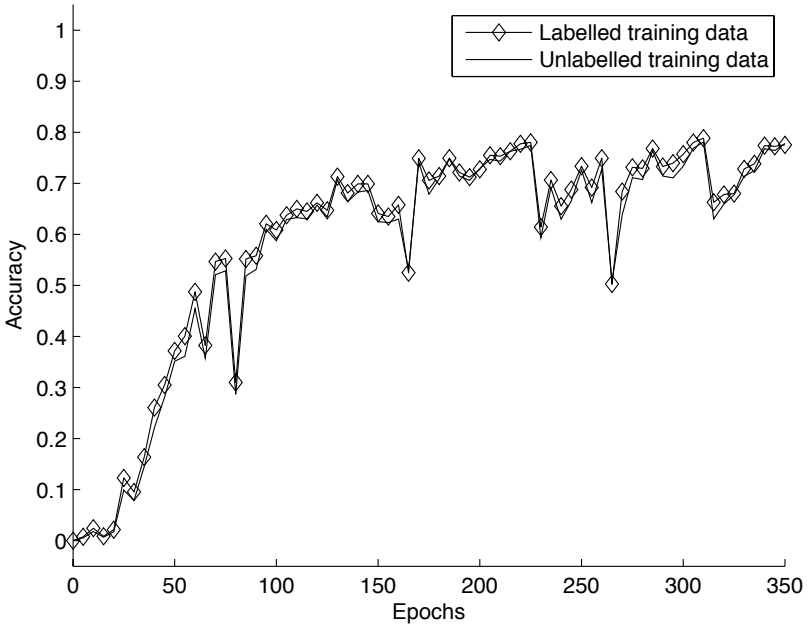

Figure 4.4: Accuracy of network 1 on labelled training examples and on unlabelled training examples consisting of only trained words.

Table 4.1: Test sentences featuring highly novel combinations of words ('Systematicity test sentences')

boys drop frisbees
frisbees drop
girls draw toys
girls draw
kids build dolls
rocks strike cats

testing. Second, there are the sentences where words appear in highly novel combinations, because specific combinations were excluded from the training data. Two types of examples can therefore be compared: sentences that were randomly selected and withheld from training data ('Generalization test sentences'), and sentences where words appear in highly novel combinations, as described in Chapter 3. These sentences, referred to as 'Systematicity test sentences', are listed in Table 4.1.

For each network and each systematicity test sentence, an 'equivalent' generalization test sentence is selected from the set of test sentences consisting of only *trained* words. Equivalent sentences contain the same verb and have the same syntactic structure as a systematicity sentence. For instance, since *frisbees drop* is a systematicity test example, *balls drop* is an equivalent generalization sentence. This allows for a paired comparison between generalization and systematicity examples.

Results show that the model's ability to generalize is very good; however, its accuracy on systematicity examples is significantly lower than on generalization examples (0.7722 accuracy on generalization examples, 0.3794 on systematicity examples, $t_{17} = -3.67; p = 0.0019$). Thus, the model cannot be said to fully display systematic behaviour. However, as noted in Section 3.5, the performance measure used here is very strict, and a score of 0.3794 is quite good, even though it is much lower than the generalization accuracy.[3]

4.3 Syntactic bootstrapping

4.3.1 Experimental set-up

To investigate the model's ability to rely on syntactic bootstrapping, the context where selftrained words appear must be taken into account. Words can appear

[3]Generalization and Systematicity examples cannot be perfectly matched. For any sentence in the Systematicity set, the corresponding Generalization sentence has the same structure and the same verb, but only one of the noun phrases can be matched. Given this, conclusions based on these results are not very reliable.

in either ambiguous or unambiguous contexts, as determined by the *trained* words in the sentence and the structures licensed by the grammar.

Unambiguous contexts are defined as those where the learner does not need to rely on its knowledge of the *selftrained* words to correctly identify the semantics of an utterance. This occurs when the *trained* words provide sufficient information to fully interpret the utterance. For example, suppose the learner is presented with the example *wugs draw*. Since *draw* is a trained verb, the learner can, in principle, rely on knowledge of this verb to infer that *wugs* must be an animate Agent, and can therefore correctly determine the semantics of the full sentence. In that sense, knowledge of *wugs* itself is not required, and the example is considered unambiguous.[4]

Conversely, ambiguous contexts are defined as those where the learner *must* rely on knowledge of the selftrained word to correctly interpret an utterance. For instance, consider the sentence *kids draw daxes*. Since verbs of the V_{draw} class license both animate and inanimate direct objects, the model cannot rely solely on knowledge of the trained words *kids* and *draw* to correctly interpret the sentence. More precisely, it can only identify the animacy feature of the Undergoer *daxes* if it has learned that this word refers to an inanimate noun, after having seen this word appear in unambiguous contexts over the course of training. If the model successfully identifies the semantics of such 'ambiguous' sentences, it must have used bootstrapping processes to progressively *learn* the meaning of selftrained words.

For selftrained nouns, identifying ambiguous and unambiguous contexts is relatively straightforward, because this only depends on whether the verb licenses both animate and inanimate arguments in a given position. For example, subjects of V_{run} are obligatorily animate, so this context is unambiguous. Subjects of V_{fall}, however, can be either animate or inanimate, so this context is ambiguous. Direct objects of V_{make} can only be inanimate, resulting in an unambiguous context. More generally, unambiguous contexts are those where a restriction is imposed on an argument. Ambiguous contexts for nouns are those where no restriction is imposed on the NP.

For selftrained verbs, no distinction is made between ambiguous and unambiguous contexts, simply because most contexts are ambiguous. Other than selftrained verbs of the V_{hit} class, all selftrained verbs appearing in transitive frames are ambiguous. This is because all verb classes other than V_{hit} license animate subjects, which means that a verb appearing in the structure NP_{anim} V NP has an ambiguous interpretation. Selftrained V_{hit} verbs are not ambiguous because they are the only verbs licensed in the frame NP_{inanim} V NP. Similarly, all verbs appearing in intransitive frames have an ambiguous interpretation because verbs taking inanimate subjects can belong either to the V_{move} or V_{fall} class, while those taking animate subjects can be of the V_{draw} or V_{run} class.

The model's ability to rely on bootstrapping processes during *interpretation*

[4]The underlying assumption, of course, is that the model has correctly learned the meaning of *trained* words, and has acquired the syntax and semantics of the language. Of course, although the model's performance on generalization examples is very good, this assumption is not entirely valid.

can be tested by presenting it with examples containing *untrained* words, rather than selftrained words. This is similar to the psycholinguistic experiments discussed in Section 2.2, in that the model is presented with a word it has never seen before, and can *only* rely on context to infer the word's meaning.

To evaluate whether the model can rely on bootstrapping processes for *acquisition* of vocabulary, what is evaluated is whether the model has succeeded in combining information from multiple frames to learn the meanings of selftrained words. Therefore, the model's performance on sentences with selftrained words is evaluated against the groundtruth. For example, in the sentence *kids draw daxes*, if the model identifies the Undergoer as *animate*, this is incorrect. As illustrated by the discussion on verbs, the model must frequently rely on information from multiple frames to successfully acquire vocabulary, which results in a very difficult task.

In what follows, I compare the model's performance on sentences containing selftrained words with 'matched' sentences containing *untrained* words. There are two selftrained words of each class, and two untrained words of each class, so there is a one-to-one correspondence between these two sets of words. More specifically, for each selftrained word, there is a unique untrained word which has the same semantic features (i.e. animacy feature in the case of nouns, verb class in the case of verbs).

Two sentences are defined as being matched if they differ only in one word in the sentence, and this 'critical' word is a selftrained or untrained word of the same lexical class. For example, *kids draw daxes* is a sentence with a selftrained word (*daxes*). If *zups* is the untrained word corresponding to *daxes*, then the matched untrained sentence is *kids draw zups*. Since there is a one-to-one mapping between selftrained and untrained words, for each selftrained sentence, there is a *unique* matched untrained sentence. This means that any set of sentences with selftrained words can be compared to a matched set of sentences with untrained words. The statistical significance of the difference between selftrained and untrained sentences can therefore be established using paired t-tests.

4.3.2 Verbs

The model's performance on sentences with selftrained and untrained verbs of each class is shown in Table 4.2 and Figure 4.5.

The results show that the model has not successfully acquired selftrained verbs, since its performance is below 0.1 for all verbs other than V_{move} and V_{hit}, and accuracy on these is only 0.1172 and 0.1805, respectively. Results on untrained verbs are similar, except that accuracy on untrained V_{draw} verbs is significantly higher than on selftrained verbs of this class (0.1540 and 0.0058 accuracy on untrained and selftrained verbs, respectively, $p \approx 0$). In addition, overall performance on untrained verbs is significantly better than on selftrained verbs (0.1225 and 0.0815 accuracy on all untrained and selftrained verbs, respectively, $p = 0.0057$).

This suggests not only that selftraining does not lead to the acquisition of

Table 4.2: Results on sentences with selftrained and untrained verbs, separated according to the class of the verb. Columns 2 and 3 ('Selftrained' and 'Untrained') give the mean accuracy on each data set, averaged over all networks and verbs in each group. The abbreviation 'dfs' refers to 'degrees of freedom', and gives the number of data points minus one.

Set	Selftrained	Untrained	t-value	dfs	p-value
V_{move}	0.1172	0.1228	-0.2460	161	0.8060
V_{draw}	0.0058	0.1540	-5.7355	154	0.0000
V_{make}	0.0773	0.0112	2.1465	46	0.0371
V_{hit}	0.1805	0.1542	0.5826	80	0.5618
V_{fall}	0.0000	0.0155	-1.4555	15	0.1661
V_{run}	0.0000	0.0450	-2.3160	6	0.0598
All verbs	0.0815	0.1225	-2.7774	467	0.0057

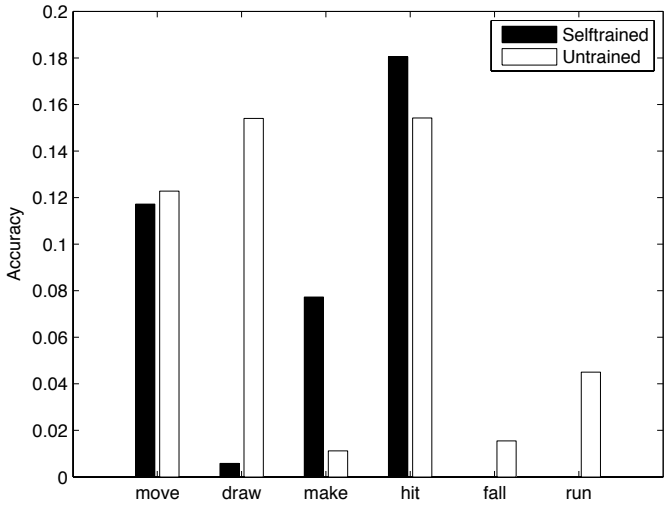

Figure 4.5: Mean accuracy on sentences with selftrained and untrained verbs, separated according to verb class. Results are averaged over all networks and verbs in each group.

Table 4.3: Results on sentences where a selftrained or untrained noun appears in an ambiguous context. Columns 2 and 3 ('Selftrained' and 'Untrained') give the mean accuracy on each data set, averaged over all networks and nouns in each group.

Set	Selftrained	Untrained	t-value	dfs	p-value
Animate nouns	0.0028	0.1747	-7.2406	143	0.0000
Inanimate nouns	0.8496	0.4561	11.4218	144	0.0000
All nouns	0.4276	0.3159	4.181	288	0.0000

verbs, but also that it has a *negative* effect in the case of V_{draw} verbs. The only case where performance on selftrained verbs is significantly higher than on untrained verbs is for the V_{make} class, but the difference is extremely small, and accuracy in both cases is very low (0.0773 and 0.0112 accuracy on selftrained and untrained verbs, respectively). Based on the results for sentences with selftrained verbs, it is clear that the model has failed to rely on bootstrapping processes to acquire new verbs.

Given that performance on untrained verbs is also very poor, it is likely that the model is not capable of relying on contextual information to make adequate inferences about the verbs. As noted earlier, most verbs occur in ambiguous contexts, suggesting that the model fails to learn verbs because it cannot incorporate information from multiple frames during selftraining.[5] Indeed, the highest accuracy score for both selftrained and untrained verbs is with those of the V_{hit} class (0.1542 and 0.1805 accuracy on untrained and selftrained verbs, respectively), whose interpretation is unambiguous.

To determine whether ambiguity is indeed the main cause of the model's failure to acquire novel verbs, further research is needed. Another possible explanation is that verbs are more complex than nouns, in the sense that they impose both syntactic and semantic requirements on their arguments. To successfully learn these verbs, the model must therefore attend not only to the syntactic contexts in which they occur, but also to the causality of the event, and the semantic features of the surrounding nouns.

4.3.3 Nouns

Table 4.3 and Figure 4.6 gives the model's performance on sentences where the critical word is a noun appearing in an ambiguous contexts. Results for sentences with selftrained and untrained nouns appearing in unambiguous contexts are shown in Table 4.4 and Figure 4.7.

These results show that selftraining can lead to the acquisition of some lexical items, since the model's performance on sentences with selftrained inanimate

[5] However, as shown by the results on generalization presented in section 4.2, the model can keep track of information presented over multiple frames when trained in supervised fashion.

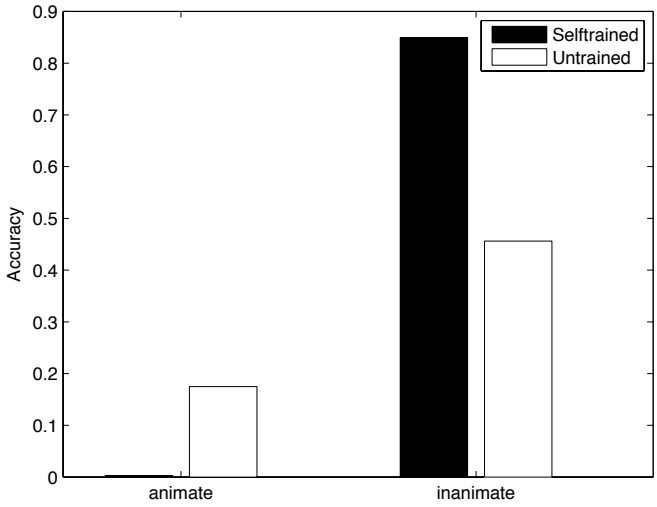

Figure 4.6: Mean accuracy on sentences where a selftrained or untrained noun appears in an ambiguous context. Results are averaged over all networks and nouns in each group.

Table 4.4: Results on sentences where a selftrained or untrained noun appears in an unambiguous context. Columns 2 and 3 ('Selftrained' and 'Untrained') give the mean accuracy on each data set, averaged over all networks and nouns in each group.

Set	Selftrained	Untrained	t-value	dfs	p-value
Animate nouns	0.0003	0.1175	-4.7421	121	0.0000
Inanimate nouns	0.6588	0.4510	5.0201	101	0.0000
All nouns	0.3002	0.2694	1.2073	223	0.2286

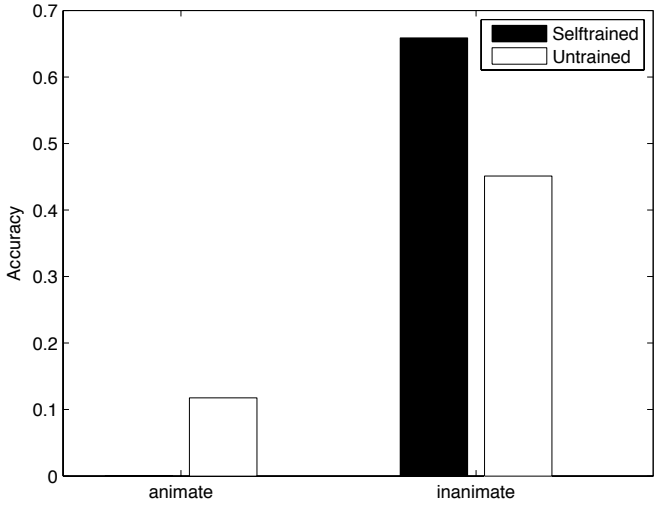

Figure 4.7: Mean accuracy on sentences where a selftrained or untrained noun appears in an unambiguous context. Results are averaged over all networks and nouns in each group.

nouns is very good. In addition, in both ambiguous and unambiguous contexts, performance on sentences with selftrained inanimate nouns is significantly higher than on matched sentences with untrained inanimates. In ambiguous contexts, mean accuracy for sentences with selftrained inanimate nouns is 0.8496, compared with 0.4561 for matched sentences with untrained nouns ($p \approx 0$); in unambiguous contexts, performance on sentences with selftrained inanimate nouns is lower than in ambiguous contexts, but still significantly higher than on matched sentences with untrained inanimate nouns (0.6588 and 0.4510 for selftrained and untrained nouns, respectively, $p \approx 0$).

Nevertheless, performance on sentences with selftrained or untrained *animate* nouns is very poor. As in the case of verbs, this suggests that the model does not make correct inferences, and is not capable of integrating information from multiple frames during selftraining. In ambiguous contexts, low accuracy is not unexpected when the critical word is an *untrained* noun, because the learner simply does not have sufficient information to infer all the required information, but performance on *selftrained* nouns in any context should be good, if the model has acquired these vocabulary items. This is clearly not the case. However, it is only when performance on untrained sentences is good that accuracy on sentences with selftrained words is higher than on corresponding sentences with untrained words. In particular, performance on sentences with untrained inanimate nouns is 0.4561 and 0.4510 in ambiguous and unambigu-

ous sentences, respectively. Given that the evaluation measure is very strict, these are both good scores. This suggests that when the learner can make good inferences about novel words based on context alone, bootstrapping can lead to vocabulary acquisition.

The difference in performance on animate versus inanimate nouns is most likely due to the fact that inanimates have a more restricted distribution. Indeed, inanimates rarely appear as subjects, and except when they appear as subjects of V_{hit} verbs, they are assigned an Undergoer role. Their semantic representations are therefore less varied than those of animate nouns, which often appear as subjects, but also as direct objects. In addition, the role assigned to animate nouns is not easily predictable because they can be Agents or Undergoers.

It is puzzling that performance on sentences with critical words appearing in ambiguous contexts is better than on sentences where they are in unambiguous contexts, even though the difference is not very large, especially for untrained words. One possible explanation is that this is an effect of the *trained* words in the utterances. For example, one unambiguous context is the object position of V_{make} verbs, which license only inanimate objects. Since no other transitive verb class imposes animacy restrictions on direct objects, it could be that the model expects animate nouns to be licensed in this context as well, and therefore incorrectly interprets sentences with verbs of this class.

Moreover, the subject position of transitive verbs is typically unambiguous, because several verbs license the structure NP_{anim} V NP. In fact, all transitive verbs other than V_{hit} appear in this configuration, so it may well be that the learner has difficulty selecting the correct verb. If the model assigns a low probability to the verb because of this uncertainty, the resulting performance score will be very low. This could explain why performance in unambiguous contexts is poor. In addition, since the subject position of most transitive verbs is filled by animate nouns, and accuracy on these is also low, it is likely that there is a correlation between the model's performance on sentences where the critical word is animate, and sentences where the critical word appears in an unambiguous context. Further research is needed to determine precisely which factors affect the model's performance in these cases, and to obtain a clearer understanding of the connection between them.

Chapter 5

Conclusion

In this thesis, I investigated whether a connectionist network could exploit systematicity in language to acquire novel words over the course of development. The model was trained to output a semantic representation (roughly corresponding to 'who did what to whom') for given sentences.

To investigate whether syntactic bootstrapping could successfully lead to lexical development, a semi-supervised training algorithm was used, with the model presented with both labelled and unlabelled data. Crucially, some vocabulary items were presented only in unlabelled sentences. In these cases, the network needed to infer (part of) of the word's meaning (e.g. grammatical category, animacy features) based on the surrounding context. The network then used its own output (i.e. its interpretation of the words and sentences) to train itself. The objective was to investigate whether the model could make use of inferred knowledge during *training* to acquire new vocabulary over time.

The model was able to successfully acquire the language when trained in *supervised* fashion. Performance on test sentences containing words presented in *labelled* examples was excellent, as was the model's ability to generalize. This indicates that the model was able to learn the syntax, semantics, and form-meaning correspondences of the language.

Nonetheless, the model did not succeed in learning novel words presented only in *unlabelled* examples. In most cases, performance on test sentences with words learned only through self-supervised training ('selftraining') was very poor. There was only one exception to this generalization: the model's interpretation of utterances containing inanimate nouns was very good. Although it is not entirely clear why the model successfully learned these words and not others, this may have been due to the fact that the distribution of inanimate nouns was more restricted than that of other lexical items.

For this project I experimented with different vocabulary sizes, training parameters (learning rate, percentage of examples withheld from training) and network sizes. Results (not reported in the thesis) were qualitatively similar for different variable settings, suggesting that the model's failure to acquire novel words is due to other factors, such as the distributional properties of the artificial

language, or the nature of the selftraining algorithm.

However, with respect to the selftraining algorithm, the fact that the model did acquire *some* lexical items suggests that the procedure was successful. Moreover, the model was trained not only on unlabelled examples with *novel* words, but also on unlabelled examples composed only of words that also appeared in *labelled* examples (*trained* words). The model's accuracy on these unlabelled examples with only trained words was nearly identical to accuracy on labelled training examples, as illustrated in Figure 4.4. This suggests that selftraining was indeed successful, at least when supported by supervised training. This is important because bootstrapping is based on the notion that children can rely on *both* linguistic and extralinguistic cues to acquire vocabulary.

The most likely explanation for the model's failure to acquire words from selftraining is that it was unable to correctly infer the meaning of novel words based on context. For bootstrapping to lead to long-term learning, a number of prerequisites must be met. First, the learner must learn the syntax and semantics of the language, as well as the correspondences between form and meaning. Given that the network performed well on examples consisting only of words learned from labelled examples, this was achieved. Second, the learner must be able to use linguistic context to infer the (partial) meaning of a novel word appearing in a familiar syntactic frame.

To investigate whether the model could succeed at this task, it was tested on two types of examples: sentences containing words seen during selftraining (*selftrained* words), and words never seen during training (*untrained* words). In most cases, the model did not accurately interpret sentences with *untrained* words, even when these were presented in unambiguous contexts, where, in principle, enough linguistic cues were available to allow the learner to make correct inferences about novel words.

It is particularly interesting to note that in the case of inanimate nouns, which the model *was* able to learn, performance on sentences with untrained words was good. This suggests that when the learner *could* correctly infer the meaning of a novel word, it *was* capable of integrating this inferred knowledge with already-existing knowledge.

Experimental research on children's ability to rely on bootstrapping processes has shown that children can indeed rely on syntactic cues to infer the meaning of novel words. Thus, if indeed this is sufficient for word learning, bootstrapping processes are likely to play an important role not only for the *interpretation* of novel words, but also for vocabulary *acquisition*. Nevertheless, since the network failed to acquire most selftrained words, only tentative conclusions about language acquisition can be drawn based on this model.

Further research is needed to determine precisely which factors lead to the model's success in acquiring some selftrained words, and its failure to acquire others. Future work could include simulations with different language models and different input languages. The main contribution of this thesis is the development of the selftraining procedure. It would be particularly interesting to investigate this use of this method with input data having statistical properties resembling those found in child-directed speech.

Bibliography

Alishahi, Afra, and Suzanne Stevenson. 2007. A computational usage-based model for learning general properties of semantic roles. In *Proceedings of the 2nd European Cognitive Science Conference*, 425–430.

Alishahi, Afra, and Suzanne Stevenson. 2010. A computational model of learning semantic roles from child-directed language. *Language and Cognitive Processes* 25:50–93.

Allen, Joseph. 1997. Probabilistic constraints in acquisition. In *Proceedings of the GALA*, volume 97, 300–305.

Brown, Roger W. 1957. Linguistic determinism and the part of speech. *Journal of Abnormal Psychology* 55:1–5.

Christiansen, Morten H., and Nick Chater. 1994. Generalization and connectionist language learning. *Mind and Language* 9:273–287.

Desai, Rutvik. 2002. Bootstrapping in miniature language acquisition. *Cognitive Systems Research* 3:15–23.

Desai, Rutvik. 2007. A model of frame and verb compliance in language acquisition. *Neurocomputing* 70:2273–2287.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science* 14:179–211.

Fisher, Cynthia, Yael Gertner, Rose M. Scott, and Sylvia Yuan. 2010. Syntactic bootstrapping. *Wiley Interdisciplinary Reviews: Cognitive Science* 1:143–149.

Fodor, Jerry A., and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition* 28:3–71.

Frank, Stefan L., Willem F.G. Haselager, and Iris van Rooij. 2009. Connectionist semantic systematicity. *Cognition* 110:358–379.

Frank, Stefan L., and Michal Čerňanský. 2008. Generalization and systematicity in echo state networks. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, ed. K. McRae B.C. Love and V.M. Sloutsky, 733–738. Austin, TX: Cognitive Science Society.

Gleitman, Lila. 1990. The structural sources of verb meanings. *Language Acquisition* 1:3–55.

Gleitman, Lila R., and Jane Gillette. 1995. The role of syntax in verb learning. In *The handbook of child language*, ed. P. Fletcher and B. MacWhinney. Oxford: Blackwell.

Hadley, Robert F. 1994. Systematicity in connectionist language learning. *Mind and Language* 9:247–272.

Landau, Barbara, and Lila. R. Gleitman. 1985. *Language and experience: Evidence from the blind child*. Cambridge, MA: Harvard University Press.

Miller, G.A., R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. WordNet: an on-line lexical database. *International journal of lexicography* 3:235–312.

Morris, William C., Garrison W. Cottrell, and Jeffrey Elman. 2000. A connectionist simulation of the empirical acquisition of grammatical relations. In *Hybrid neural systems*, ed. Stefan Wermter and Ron Sun, 175–193. Berlin: Springer-Verlag.

Naigles, Letitia R. 1996. The use of multiple frames in verb learning via syntactic bootstrapping. *Cognition* 58:221–251.

Naigles, Letitia R. 1998. Developmental changes in the use of structure in verb learning: Evidence from preferential looking. *Advances in infancy research* 12:298–318.

Naigles, Letitia R., and Lauren D. Swensen. 2007. Syntactic supports for word learning. In *The handbook of language development*, ed. E. Hoff and M. Shatz, 212–231. New York: Blackwell.

Quine, W.V.O. 1960. *Word and object*. Cambridge, MA: The MIT Press.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986. Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition, (vol. 1)*, ed. D.E. Rumelhart and J.L. McClelland. Cambridge, MA: MIT Press.

Scott, Rose M., and Cynthia Fisher. 2009. Two-year-olds use distributional cues to interpret transitivity-alternating verbs. *Language and Cognitive Processes* 24:777–803.

van der Velde, F., Gwendid T. van der Voort van der Kleij, and Marc De Kamps. 2004. Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science* 16:21–46.

Yuan, Sylvia, and Cynthia Fisher. 2009. "Really? She Blicked the Baby?". *Psychological Science* 20:619–626.