Doktori értekezés

Perspectives on the Lexicon

Tóth Ágoston

Debreceni Egyetem
Bölcsészettudományi Kar, 2005

Perspectives on the Lexicon

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
a nyelvtudomány tudományágban

Írta: Tóth Ágoston
okleveles angol nyelv és irodalom szakos tanár,
okleveles informatika tanár

Készült a Debreceni Egyetem Nyelvtudományok doktori iskolája
(Modern nyelvészeti programja) keretében

Témavezető: Dr. Hollósy Béla

____

A doktori szigorlati bizottság:
elnök: Dr. _______________
tagok: Dr. _______________
Dr. _______________

A doktori szigorlat időpontja: 2005. _____________ __.

Az értekezés bírálói:
Dr. _______________
Dr. _______________
Dr. _______________

A bírálóbizottság:
elnök: Dr. _______________
tagok: Dr. _______________
Dr. _______________
Dr. _______________
Dr. _______________

A nyilvános vita időpontja: 200_. _____________ __.

Én, Tóth Ágoston teljes felelősségem tudatában kijelentem, hogy a benyújtott értekezés a szerzői jog nemzetközi normáinak tiszteletben tartásával készült.

____

Összefoglaló

Az értekezés célja a lexikon szerepének, működésének többszempontú vizsgálata. Ennek során először elemzem *a lexikai komponens pozícióját* a chomskyánus generatív grammatika kiválasztott alternatív elméleteiben, vizsgálva, hogy a mondattan és az alaktan között kialakuló munkamegosztás (különösen a ragozás és a képzés kérdésében) hogyan befolyásolja a nyelvészek lexikonról kialakított elképzeléseit. A folytatásban a poliszémia és homonímia kérdéskörének tömör elemzésével szemléltetem azt, hogy a *jelentéstani alapproblémák komoly buktatókat rejtenek* a lexikon tervezése során. Az elmélet mellett a gyakorlatra is hangsúlyt fektet az értekezés: megvizsgálom, hogy két jelentős nyelvi adatbázisban, a WordNet-ben és a FrameNet-ben a jelentés mely aspektusai köré szervezték a tárolt információkat, miközben a WordNet-tel kapcsolatban az elméleti oldalról korábban vizsgált jelenségeket, főként a ragozás és a szóképzés kezelésének módját is elemzem. Egy harmadik adatbázis, a MindNet bemutatása kapcsolja össze ezt a fejezetet a szójelentéssel foglalkozó fejezettel, végül Véronis és Ide jelentéstároló rendszerének bemutatása vezeti át az olvasót a dolgozat második, konnekcionista részére.

Az értekezés fontos újítása, hogy a vizsgálatba bevonja a konnekcionista, *mesterséges neurális hálózat alapú* kísérletek releváns tapasztalatait. Az 5. fejezetben látjuk, ahogy felvázolódik a nyelvi információ ábrázolásának egy lehetséges módszertana elsősorban Rumelhart és Elman kísérleteiben, majd bemutatok néhány mesterséges neurális hálózat konstrukciót, melyek a nyelvi bemenet kezelésében jelentős potenciállal rendelkeznek.

Végül ismertetem önálló kutatásomat, melyhez kifejlesztettem egy sajátos neurális hálózat szerkezetet, melyet FrameNet-szerű jelentéstani keret (frame) és keret-elem (frame element) felismerésre tanítottam be generált korpuszon. A mesterséges neurális hálózat viselkedését egy kísérletsorban vizsgáltam a betanítottól eltérő bemenet megjelenése esetén. A kísérlethez kifejlesztett „ikertérképes" bemeneti interfész képes nagyszámú szóalak automatikus megkülönböztetésére, így lehetőséget kínál a *szóalapú nyelvi bemenet* ábrázolására, melynek megoldása a neurális hálózatos kísérletek visszatérő problémája. A hálózat betanítására használt generált betanítókorpusz semmilyen (szófaji, mondattani, jelentéstani) *többletinformációt nem tartalmaz.* A visszacsatolásos hálózatokból az értekezésben ismertetett módon összeállított struktúra megfelelő általánosítási képességgel és memória-potenciállal rendelkezik a kitűzött feladat végrehajtására: mondatszekvenciák (néhány mondatos szövegrészletek) jelentéstani címkékkel történő annotálására. A kísérletsorozat keretei között a modell képes kompenzálni a hiányos bemenetet (hiányzó

mondatok és összetevők), vagy az egyéb jellegű zajt (lecserélt összetevők). A lecserélt összetevőket vizsgáló kísérlet megmutatta, hogy a manipulált bemenet felcímkézése nem csupán a kontextus érintetlenül maradt elemei alapján történt, hanem a csereként beállított összetevők (a velük betanítás során asszociált információdarabkák) is meghatározták annak sikerét. A konnekcionista kísérletekben potenciálisan megjelenő *szórványlexikont* a disszertáció egy lehetséges, a korábbiaknál ugyan jelenleg nem eredményesebb, de újszerű lexikon-közelítésként mutatja be.

Acknowledgements

I would like to thank my thesis supervisor, dr. Béla Hollósy, for his continuous support, advice, patience and encouragement, without which I could have never accomplished this task.

I express my profound gratitude to dr. Gábor Prószéky, dr. László Hunyadi, dr. Tibor Laczkó, and dr. István Boda, who read the first version of the dissertation and provided me with significant insights and invaluable comments.

Finally, I would also like to take this opportunity to thank all my family for their unwavering support and encouragement. I also truly appreciate their genuine interest, which has been a constant inspiration.

Table of Contents

1. INTRODUCTION

One of the most fundamental assumptions about language, which we all seem to share, is that we can make a distinction between linguistic idiosyncrasies and systematic linguistic processes. Many linguists have found it reasonable to work with a separate module of language that contains linguistic idiosyncrasies, with or without devices to handle various classes of regularities. We refer to this module as the *lexicon*. A goal of this thesis is to illustrate the delicate relationship between this and other aspects (or modules) of the human language system. The more general aim of this work is to offer important and useful perspectives on the lexicon.

Human languages show an intricate interplay of linguistic regularities and idiosyncrasies. It is a general approach to put idiosyncrasies into the lexicon, and delegate those phenomena that follow regular patterns to other module(s) of grammar. Actual words are not pure idiosyncrasies, however: words themselves exhibit traits of systematic processes. What we put into the lexicon (morphemes, stems, "fully-cooked" words undergoing compounding, derivation and/or inflection) and how we handle them are questions that cannot be separated from each other. **Chapter 2** of this thesis deals with this dilemma as manifested in the Chomskyan literature of Generative Grammar. We will see that the properties of the lexicon fluctuate considerably depending on the nature of the Syntax/Morphology boundary. This chapter also gives some basic insight into how morphological processes are approached in Natural Language Processing.

The lexicon should contain information about the meaning of the objects it stores, but it has always been a challenge to grasp the nature of word meaning. Since my thesis is not a contribution to the rich literature of lexical semantics, I do not discuss a full variety of relevant issues. Instead, I have selected *polysemy* and the related area of *homonymy* to illustrate the problems we face when we work with meanings. **Chapter 3** aims to present polysemy as a major natural language phenomenon that shapes the way we think about language and which should also be carefully considered in modeling language processing. Cruse (2000) identifies two opposing standpoints: monosemists opt for listing as few senses as possible and storing the regularities behind meaning "extension" in lexical rules, while polysemists argue that lexical rules only record "*potential* extensions of meaning", and therefore they encourage us to list a fuller variety of *actual* words (Cruse 2000:97). Contemplating the difference between the monosemic and the polysemic approaches is not merely linguistic theorizing: the compiler of a lexical database, or the lexicographer compiling a dictionary keeps facing decisions related to polysemy and homonymy in his or her everyday work.

While chapter 3 and the greater part of chapter 2 explore theoretical perspectives, **chapter 4** takes the more practice-oriented point of view of computational linguistics (while we also consider some aspects of compiling a traditional dictionary in lexicography). The main considerations of chapter 4 are not entirely different from those of the previous chapter, however, since we inquire into the question of *representing lexical knowledge* in real-life databases, most importantly, in WordNet and FrameNet. WordNet's lexical and semantic relations and the frame-relative frame element structure of FrameNet can be exploited by Natural Language Processing applications. This chapter also contains my analysis of how homonymy and inflectional and derivational morphology are treated in WordNet.

The second part of this thesis presents a perspective that is a novelty in lexicon-related studies. Pléh (1998) argues that symbol manipulation and connectionism are two approaches to cognitive science; while chapter 2 deals with a prominent linguistic manifestation of the symbol-manipulation approach, the final chapters take us to the connectionist side. **Chapter 5** outlines the general features of connectionism and surveys selected neural network models. Via an overview of the experiments reported in Elman (1990), Kohonen (1984), James and Miikkulainen (1995) and Mayberry and Miikkulainen (1999), the chapter illustrates possible approaches to representing linguistic input in an Artificial Neural Network.

A common feature of the models developed in the connectionist framework is an ability to take and coordinate input from a plethora of sources while finding regularities and particularities automatically during the training process (whether it is supervised or unsupervised). As a down-to-earth problem, however, all connectionist models face the difficulty of representing temporal sequences in general and *linguistic input* in particular. These issues (and many more) are discussed in the context of my neural network experiment presented in **chapter 6**. The system takes unannotated word forms and multi-word expressions making up short sentence sequences as input, and it is able to carry out FrameNet-style Frame and Frame Element recognition. In this experiment, reasonable linguistic performance is attained *without a separated and functionally transparent lexical component*: linguistic idiosyncrasies and regularities that are required to solve the task are dissolved in the neural network. Although the experiment described here tackles fundamental issues, and in general, the distributed lexis that emerges in non-localist neural networks is much less researched than the symbol-based lexicon, I believe that the theory and practice of lexicon design should consider both approaches, i.e. the symbol-based and the connectionist alternatives.

2. THE LEXICON OF GENERATIVE GRAMMARS AND MORPHOLOGY IN NATURAL LANGUAGE PROCESSING

2.1 Where is morphology?

The main goal of this chapter is to investigate the function of the lexical component and the related question of the place of morphological processes in selected works on Generative Grammar, while implementational issues (morphological analysis in Natural Language Processing) are also discussed briefly in section 2.5.

Please note that this chapter is not meant to survey the generative thought, only some of the important milestones of the generative tradition that have contributed to how linguists have been thinking about the lexicon are presented. Spencer (1991) thoroughly investigates morphological theory in the context of Generative Grammar. Spencer's book features a detailed survey of the relevant literature as well as various thematic chapters (e.g. on allomorphy, compounding and clitics). Another detailed study, thematically grouped into sections on individual morphological phenomena (clitics, agreement, passive morphology, compounds, derived nominals, derived and compound verbs) is in Hendrick (1995). Ruszkiewicz (1997) is an entire volume devoted to an elaborate discussion of the same topic, but even he has to point out that a full survey of the field is beyond the scope of his book.

This chapter is organized around the following questions:

- Is morphology an independent component in the selected models of Generative Grammar?
- Where is concatenative morphology (more precisely: derivational affixation, inflectional affixation and compounding) implemented in various versions of Generative Grammar?

Our starting point will be Chomsky (1957): he omits morphology as a separate component from his model. Then, we will see the rise of morphology as an independent component of grammar, only to be rejected in Lieber's (1992) GB-based model (the final framework discussed here), which can be treated as an implementation of the idea underlying Chomsky (1957). Based on the discussion of the treatment of morphological processes, the sections in this chapter try to show how these models delineate the notion of the *lexicon*.

Some of the works cited in this chapter are robust monographs elaborating an abundance of morphological phenomena in great detail, and I can only give a sketchy overview of them here, concentrating on selected issues, reorganizing the original structure of each source. The reader is hereby referred to the original works for more detail.

2.2 Chomsky 1957-1970

The linguistic era under scrutiny is considered a milestone in the history of syntax, and it has affected other linguistic disciplines as well, including morphology. The following summary is due to Anderson:

> In the early years of the development of a theory of generative grammar (roughly 1955 through the early 1970s), a striking difference between the research problems that characterized the emerging field and those that had occupied its predecessors was the precipitous decline of the study of morphology.
>
> (Anderson 1982:571)

2.2.1 Chomsky (1957)

Chomsky’s eye-opener 1957 work on Transformational Generative Grammar introduces the following levels of representation: Phrase structure, Transformational structure and Morphophonemics (Chomsky 1957:46). The role of the morphophonemic level is to convert the output of the first two levels into a string of phonemes.

The terminology that Chomsky (1957) uses is instructive in many ways: he refers to *morphemes* as building blocks of sentences. Let us examine the following paragraph as an illustration:

> Let us now consider various ways of describing the morphemic structure of sentences. We ask what sort of grammar is necessary to generate all the sequences of morphemes (or words) that constitute grammatical English sentences, and only these.
>
> (Chomsky 1957:18, emphasis added)

Chomsky (1957) characterizes transformations as devices that “may rearrange strings or may add or delete morphemes”. It leads us to the conclusion that transformations take care of morpho-syntactic phenomena: the need for a separate morphological component is rejected in this framework.

2.2.2 Chomsky's **Aspects**-*model*

The sophisticated model in Chomsky (1965) still assumes that grammar does not need to contain a separate morphological component.

The approach to inflectional morphological phenomena chosen by Chomsky (1965) can be characterized as paradigmatic: it makes use of grammatical paradigms of traditional grammars. These paradigms are formulated in terms of features (discrete but not necessarily binary features) describing the morphological structure of a formative (Chomsky 1965:171). Chomsky's German illustrative example, the noun *Brüder*, is described in four paradigm "dimensions". The dimensions Gender and Declensional Class are *inherent* to lexical formatives (ibid.), i.e. they are determined by the lexical entry itself. Others, including Number, are inherent to the Phrase-marker, and are associated with the entry after lexical insertion into a given phrase structure position (p. 177). The third set of features (in our example, Case) is added during transformations (pp. 171-176). Chomsky points out that the rules of agreement also "belong to the transformational component ..." (p. 174).

Being "sporadic" and only "quasi-productive", derivational processes cause more trouble. To put it simply, Chomsky's dilemma is whether we store or generate derived items. He suggests that both methods should be employed: more productive derivational processes should be generated, whereas less productive ones should be stored. He refers to "quasi-productive" processes, however, including those responsible for the following words: *horror, horrid, horrify; terror, (*terrid), terrify; candor, candid, (*candify); telegram, phonograph, gramophone* (Chomsky 1965:186). This treatment of productivity is fuzzy in the everyday sense and in the scientific meaning of "fuzziness", especially in the light of the sharp contrast in the treatment of productive and unproductive processes, but this does not seem to void Chomsky's dual solution for morphological derivation.

At the same time, Chomsky points out the drawbacks of this approach: he considers listing these items without internal structure "very unfortunate". He outlines two alternative solutions instead. His first proposal is to extend the scope of the *lexical rule* of grammar to operate within the lexicon, facilitating limited lexicon-internal computations[1]. These computations construct derived forms from simple stems by adding affixes one-by-one. They make use of features stored in lexical entries (p. 187-188); the consequences of this

[1] In this framework, rewriting rules generate strings with grammatical formatives and complex symbols. Then, it is up to the lexical rule to replace lexical formatives for complex symbols forming the terminal string from the preterminal string (Chomsky 1965:84).

procedure on the lexicon are described shortly. The second alternative is the introduction of a new level of context-sensitive rewriting rules within the lexicon. This alternative is readily rejected, however, without further explanation (p. 188).

As far as productive derivational processes are concerned, Chomsky picks *nominalization* as an unproblematic case. He suggests that nouns such as *destruction* and *refusal* should not appear in the lexicon. “Rather, *destroy* and *refuse* will be entered in the lexicon with a feature specification that determines the phonetic form they will assume (by later phonological rules) when they appear in nominalized sentences” (Chomksy 1965:184) having gone through a nominalization transformation.

It is the base of the grammar where the lexicon is situated. It is a set of *lexical entries*, “each lexical entry being a pair *(D, C)*, where *D* is a phonological distinctive feature matrix ‘spelling’ a certain lexical formative and *C* is a collection of specified syntactic features (a complex symbol)” (Chomsky 1965:84).

Formatives are defined as minimal syntactically functioning units (Chomsky 1965:5), and they are subdivided into *lexical* items ... and *grammatical* items (p. 65). Due to a later modification (the extension of the scope of the *lexical rule*), motivated by quasi-productive derivational processes, not only formatives, but more complex symbols will potentially be inserted into the phrase marker.

Consider the phrases *the boy was abundant* and *the boy elapsed* (p. 76). Should the anomaly present in these phrases be accounted for by the syntactic component, or a separate semantic component? Chomsky argues for a theory of syntax that is able to account for this kind of anomaly. In such a system, these sentences will be generated "only by relaxation of certain syntactic conditions" (p. 79). It leads us to the question of subcategorization (and selectional restrictions). Chomsky (1965) introduces *syntactic features* and rewriting rules that create and operate on *complex symbols*, "each complex symbol being a set of specified syntactic features" (p. 82), e.g. [+N, +Common]. Lexical formatives carry syntactic features (e.g. we may add [+Common], [+Human] for *boy*, Chomsky 1965:82). Syntactic features can be positively specified, negatively specified or unspecified (p. 81). Rewrite rules must match the syntactic features of the formatives they operate on[2].

[2] Chomsky abandons his former view according to which subcategorization is formulated in terms of rewriting rules, which makes subcategorization work in a strictly hierarchical manner (Chomsky 1965:79). Chomsky gives credit to Halle and adopts his device of *distinctive-feature matrices* in syntactic description (Chomsky 1965:81). It brings a different division of labor between the two parts of the *base*: subcategorization rules can be assigned to the lexical component (as syntactic redundancy rules) rather than to the categorial component (Chomsky 1965:121).

Consider the following sample entries, which store and refer to subcategorization information (Chomksy 1965:85,94):

sincerity, [+N, –Count, +Abstract]

boy, [+N, –Count, +Common, +Animate, +Human]

believe, [+V, + —— NP, + —— *that*⌒S']

persuade, [+V, + —— NP (*of*⌒Det⌒N) S']

At the level of lexical formatives, no distinction is suggested to exist between singular and plural: formatives are unspecified for this feature (p. 181). As we have seen earlier, this syntactic feature gets associated with the entry after lexical insertion into a given phrase structure position (but before transformations).

Let us now examine the above-mentioned revision to the structure of the lexicon motivated by quasi-productive morphological processes and the extension of the scope of the lexical rule. Chomsky suggests that we store stems and affixes as lexical entries (p. 187):

graph, [+Stem$_1$, ...]

horr, [+Stem$_2$, ...]

fright, [+Stem$_3$, ...]

tele⌒Stem$_1$, [F$_1$, ...]

Stem$_2$⌒*ify*, [G$_1$, ...]

Stem$_3$⌒*en*, [H$_1$, ...]

Stem$_1$ can be replaced by *graph* as well as other stems (e.g. *scope*, *phone*) that comply with contextual features F$_1$, F$_2$, etc. Lexical rules help derive items like *telegraph* from *tele-* prior to final lexical insertion. Furthermore, "[t]here may be several layers of such extension of base derivations within the lexicon, in the case of morphologically complex forms." (p. 187).

Last, but not least, Chomsky makes the following revision to the notion of lexicon:

> Of course, a lexical entry must also contain a definition, in a complete grammar, and it can be plausibly argued ... that this too consists simply of a set of features... We might, then, take a lexical entry to be simply a set of features, some syntactic, some phonological, some semantic.
>
> (Chomsky 1965:214)

Chomsky does not elaborate semantic features in more detail, leaving this territory to semanticists. As far as phonological features are concerned, Chomsky refers to existing frameworks, which are supposed to do the job well on their own.

As an important development of the years after 1965, even the seemingly problem-free case of the nominalization transformation provoked a lot of discussion. In 1970, Chomsky had to return to this question in a short, but very influential paper.

2.2.3 Chomsky's remarks on nominalization

Chomsky (1970) does not try to put together a stand-alone model of syntax. Instead, it is devoted to two types of nominalization in English: gerundive and derivational. He exemplifies the two processes with these phrases:

- *John's being eager to please*, and
- *John's eagerness to please.*

Both phrases are related in meaning to the proposition *'John is eager to please'* (Chomsky 1970:187).

Chomsky points out that gerundive nominalization seems more productive, and "the relation of meaning between the nominal and the proposition is quite regular" (p. 187). This leads to allowing the transformational component to take care of gerundive nominalization.

Chomsky argues that productivity is much more restricted in the case of derivational processes in general. In particular, derivational nominalization can be accounted for in two ways, as described below:

> We might extend the base rules to accommodate the derived nominal directly (I will refer to this as the 'lexicalist position'), thus simplifying the transformational component; or, alternatively, we might simplify the base structures, excluding these forms, and derive them by some extension of the transformational apparatus (the 'transformationalist position')
>
> (Chomsky 1970:188)

As we have seen in section 2.2.2, the *Aspects* model argues against the lexicalist position in the case of 'productive' derivational processes, including cases of nominalization (Chomsky 1965:184)[3].

[3] The following is a footnote from Chomsky's 1970 article: "The lexicalist position ... is ... rejected, incorrectly, as I now believe, in Chomsky (1965, p. 184)".

In this lexicalist framework, some transformations are still needed for taking care of complex nominals (these transformations include agent-postposing and NP-preposing, cf. Chomsky 1970:203ff). They have nothing to do with derivation proper, however; Chomsky goes on to explain that morphological processes are not hosted by the transformational component.

Although the 1970 version of the lexicon accommodates derived forms, with an option to apply redundancy rules to form new words, derivation remains somewhat underspecified in this proposal. Chomsky's position is the following: since a number of affixes may be involved in the production of any derived form, "fairly idiosyncratic morphological rules" affect the *phonological* form of the derived nominals (Chomsky 1970:190). It amounts to delegating the problem to a morpho-phonological component, perhaps producing e.g. [rɪ'fjuːzl] from refuse$_{+N}$, and [dɪ'strʌkʃn] from destroy$_{+N}$. The lexicon does not list phonological properties for affixes.

Chomsky makes it clear that *contextual features* must be stored in the lexicon for lexical entries; these features include fixed selectional and strict subcategorization features, but the lexicon contains entries that may or may not be specified with respect to lexical category membership (N, V, etc., p. 190). Chomsky's aim is to make it possible for lexical items to appear in more than one lexical category. For such a lexical item, completely fixed contextual features could be disastrous, putting successful lexical insertion and transformations at risk. "The lexical entry may specify that semantic features are in part dependent on the choice of one or another of these categorial features... Insofar as there are regularities ..., these can be expressed by redundancy rules in the lexicon" (ibid.).

As far as the connection between nominalization and contextual features are concerned, Chomsky makes the following comment:

> It must be noted that only in the *simplest* case will exactly the same contextual (and other) features be associated with an item as a verb and as a noun. In general, lexical entries involve sets of shared features, organized in complex and little understood ways, and we should expect to find the same phenomenon in the case of derived nominals, given the lexicalist hypothesis.
>
> (Chomsky 1970:201)

Examples are many. Chomsky (1970:203) points out, for instance, that since passivizability is a property of verbs, we must indicate it somehow in the lexicon thus preventing nouns from getting passivized. Finally, Chomsky offers two ways of accounting for discrepancies in general: either we can add contextual features as lexical properties (in cases of idiosyncratic phenomena), or we can add redundancy rules (in the case of regular processes).

Chomsky's lexicalism[4] triggered a mistrust in transformations, it created a vacuum "that could only be filled with an independent theory of morphology" (Webelhuth 1995:26), and it also classed up the role of the lexicon in linguistics.

2.3 Lexical Morphology

I adopt Ruszkiewicz's terminological clarification to define the word *lexical*.

> Since models of morphology based on the lexicalist position need not be lexical, a different criterion has been suggested here for differentiating between lexical and syntactic processes, namely the ordering of various processes with respect to the rule of lexical insertion. Rules which apply before the lexical insertion rule are described as lexical, those which follow it are syntactic.
> (Ruszkiewicz 1997:84)

The positive lexicalist frameworks discussed in this section comply with Ruszkiewicz's definition of *lexical*. They will be classified into two classes representing the Strong Lexicalist Hypothesis (SLH) and the Weak Lexicalist Hypothesis (WLH):

> **The weak version of the Lexicalist Hypothesis**
> All of derivational morphology is the domain of pre-lexical-insertion processes. All inflection constitutes the realm of post-lexical-insertion phenomena.
> **The strong version of the Lexicalist Hypothesis**
> Both derivational morphology and inflectional morphology constitute the domain of pre-lexical-insertion processes.
> (Ruszkiewicz 1997:61)

2.3.1 Halle's Lexicalist Hypothesis (1973)

Figure 2-1 illustrates the structure of grammar suggested by Halle (1973). He offers a unified treatment of inflectional and derivational morphology (positive and strong lexicalist approach) using word formation rules (Halle 1973:6) implemented in a separate module.

[4] Botha's (1984:136) refers to the position taken by Chomsky (1970) as the *Basic Lexicalist Position*.

The word formation module takes its primary input from a list of morphemes. The output is channeled through a filter module. The words coming from the filter are collected in a dictionary of words. The word formation component receives feedback from the dictionary as well as the phonological component, which is necessary to facilitate rules that cannot be applied to the "raw" morphemes coming from the morpheme list in the "first pass" of word formation.

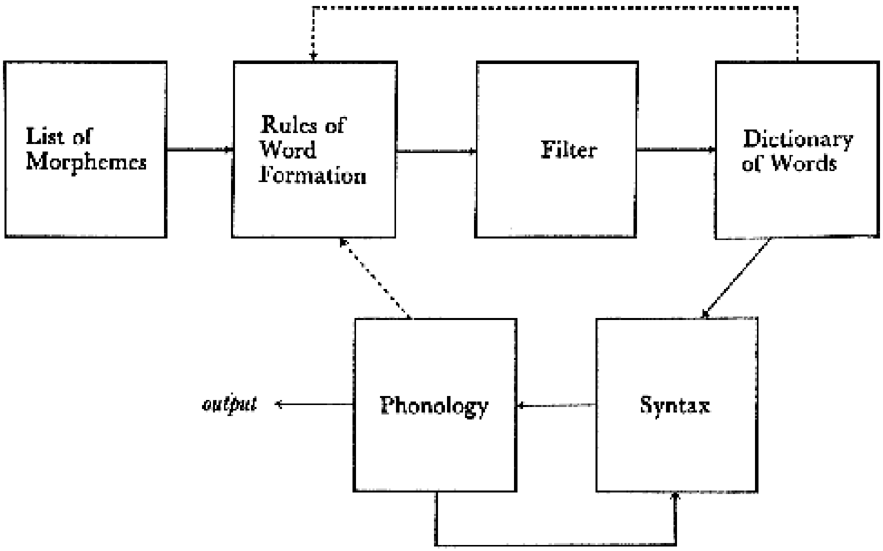


Figure 2-1 (Halle 1973:8)

The filter stores information that is attached to word-candidates coming from the word-formation component, and also doubles as an exception list that prevents nonsense words from being generated (hence the name). The exception list function is simply implemented by adding the feature [-Lexical insertion] to unwanted word forms (Halle 1973:5). It might be interesting to note that Halle also uses the exception filter to modify phonetic/phonological features of word candidates (he adduces segmental examples, so suprasegmental phenomena, such as intonation and intensity, are probably left unaffected). I would like to point out, however, that there is no way of working up general rules within the exception filter, so generalities have to be codified in the word formation component.

Halle's system utilizes feedback paths, which are shown in Figure 2-1, too. Why is it necessary to implement those two feedback circuits to the word formation module? The argument for the shorter feedback cycle (Word Formation → Filter → Dictionary → Word Formation) is the following: the morpheme list does not contain grammatical category information, but word formation rules are difficult to operate without knowing the grammatical category membership of the input. It affects inflection as well as derivation. "It has been noted above that rules of word formation must have access to the dictionary [of actual words]; i.e. that certain words presuppose the existence of other words" (Halle 1973:13). He also notes that phonetic conditions may affect word formation: this is the argument for the second (Word Formation → Filter → Dictionary → Syntax → Phonology → Word Formation) feedback cycle. For instance, deadjectival verbs with the suffix *-en* are subject to the condition that their base must be monosyllabic and end with an obstruent, e.g. *whiten* is acceptable but **greenen* is not (ibid.).

A problem arises, however, during lexical insertion. Halle proposes that the lexical insertion transformation should select items from the dictionary of actual words (Halle 1973:9; notice that figure 2-1 shows a single input channel to syntax). He realizes, however, that "the <u>case</u> which a given noun takes in a sentence is normally determined by its position in surface structure" (ibid., emphasis added), therefore case inflections cannot be applied before lexical insertion. His solution is summarized in the following way: the lexical insertion transformation inserts paradigms that contain all or selected inflected forms of a word (p. 9). Halle argues that a "perfectly general convention can then eliminate all but the one inflected form that fits syntactically into the configuration in which the word is found in surface structure" (ibid.).

This "perfectly general convention" is not elaborated in any detail, however.

In addition to unifying derivational and inflectional morphology in a single component of grammar, Halle also argues against treating derivational processes as phonological regularities: he insists that word formation[5] rules ought to have access to "different stages in a derivation", whereas the rules of phonology should be restricted to "information overtly present in the string at the point in the derivation at which the phonological rule applies" (p. 15).

He labels word formation rule application as 'passive' when compared to the more active use of syntactic and phonological rules.

We have seen that Halle (1973) implements a lexicon[6] that is divided into multiple components. Let us review and further discuss the role of these components.

The *List of Morphemes* contains roots and affixes of all types (Halle 1973:3). Let us take the following illustrative example:

> ... the entry for the English morpheme *write* must contain the information that it is a verbal root, that it is a member of the "non-Latinate" portion of the list (it is by virtue of this fact that it is allowed by the rules of word formation to combine with certain affixes

[5] Halle makes an explicit distinction between word formation and phonology (p. 15).

> and not with others), that it is among the small class of verb stems that undergo the so-called “strong” conjugation, etc.
> (Halle 1973:4)

Morphological subcategorization information will be required at a later stage (during word formation) to decide which stems and affixes can be combined into actual words. Halle suggests that this information should be stored in the form of *templates* such as these (Halle 1973:10):

[STEM + *i* + *ty*]$_{\mathrm{N}}$

[STEM + *some*]$_{\mathrm{A}}$

[*be* + STEM]$_{\mathrm{V}}$

Morphemes in the morpheme list must be marked appropriately “so that a given stem will be substitutable only in certain frames and not in others” (Halle 1973:10). On the other hand, grammatical category and syntactic subcategorization are not specified for morphemes.

As a result of the application of the templates, grammatical category will be assigned to the word coming from the word formation component. In addition to this, syntactic subcategorization and selectional restriction information must be added by the rules of word formation. In more idiosyncratic cases, where it is impossible to add subcategorization during word formation, the Filter component adds the missing pieces of information.

To sum it up, the general stages of morphological processing are the following:

1. morpheme-storage
2. word-formation
3. filter
4. dictionary
5. go back to stage 2 if necessary ⮥
6. lexical insertion

Halle’s (1973) lexicon is therefore based on the following huge stores of information: a morpheme-list; a (structured) word-list, i.e. the dictionary; and a third store of idiosyncrasies, comparable in size to the first two, namely the list of exceptions that operates the exception filter. According to Halle (1973), the list of morphemes and the exception list are static, permanently stored, while the dictionary may not exist in its entirety all the time, but can also be dynamically extended in a real-time fashion: “it is

[6] Please note that the term *lexicon* is not used in Halle (1973).

possible to suppose that a large part of the dictionary is stored in the speaker's permanent memory and that he needs to invoke the word formation component only when he hears an unfamiliar word or uses a word freely invented" (p. 16).

As we have seen in this section, lexical insertion inserts partial or whole paradigms (certain or all inflected forms of a given word, cf. Halle 1973:9). Where can we form paradigms? The best option is the dictionary. Under this assumption offered by Halle, "the dictionary must be organized into paradigms in some way and it would then no longer be equivalent to the logical product of the morpheme list, the word formation rules, and the exception filter" (p. 9).

2.3.2 Nominal Compounding in Jackendoff (1975)

Let us return for a moment to Halle's (1973) implementation of his Filter component. It is up to the Filter to classify a form as appropriate for lexical insertion or inappropriate (by assigning the [-Lexical insertion] feature to the word form). It is also hypothesized that words with productive affixes are generated as [+Lexical insertion] by default and checked against an exception list, while unproductive affixes are produced with the [-Lexical insertion] feature assigned to them by default, and enabled by checking it against some sort of inclusion list (Halle 1973:5). Still, all (appropriate and inappropriate) forms are generated, which has serious storage space requirements.

Jackendoff (1975) points out that exception filtering may not be manageable as far as compounding is concerned, since it involves connecting each noun in the lexicon to every other noun before the filtering process can take place[7] (Jackendoff 1975:655). What Jackendoff proposes instead is an approach giving "each actually occurring compound a fully specified lexical entry" (ibid.). Compound classes are defined by redundancy rules within the lexicon, so morphological and semantic relations are properly taken care of by general rules[8], and actual compounds are stored in the lexicon. Redundancy rules allow us to formulate generalities while relating items "only partially" (while a transformation, as a generative device, "cannot express partial relations", Jackendoff 1975:658). A redundancy rule that establishes a connection between a deverbal noun with the *-ion* suffix and its base form is the following (Jackendoff 1975:642):

[7] Consider a lexicon with 40,000 nouns: combining each noun with all remaining nouns amounts to storing (memorizing) slightly less than 1,600,000,000 noun combinations, and let us not forget that compounds can undergo further compounding, too.

$$\begin{bmatrix} x \\ /y + \text{ion}/ \\ +\text{N} \\ +[\text{NP}_1\text{'}s \; \_\_\_ \; \text{(P) NP}_2] \\ \text{ABSTRACT RESULT OF ACT} \\ \text{OF NP}_1 \text{ 'S } Z \text{ –ING NP}_2 \end{bmatrix} \longleftrightarrow \begin{bmatrix} w \\ /y/ \\ +\text{V} \\ +[\text{NP}_1\text{'}s \; \_\_\_ \; \text{(P) NP}_2] \\ \text{NP}_1 \, Z \, \text{NP}_1 \end{bmatrix}$$

At a later point in his paper, however, Jackendoff rejects the isomorphism between morphological structure and meaning suggested by this particular redundancy rule:

> In fact, this formulation will not do. It claims that there is a particular meaning, ABSTRACT RESULT OF ACT OF V-ING, associated with the ending *-ion.* However, several different semantic relations obtain between *-ion* nominals and their related verbs, and several nominalizing endings can express the same range of meanings.
> (Jackendoff 1975:650)

What he suggests instead is a bifurcation of redundancy rules into morphological rules (M-rules) and semantic rules (S-rules). Morphological rules can express semantic relatedness, too, while S-rules may augment that information if necessary. The need for this formulation is exemplified by the following matrix, in which each row holds nouns of the same semantic category, and each column contains morphologically similar words (Jackendoff 1975:651):

	M1	M2	M3
S1:	discussion	argument	rebuttal
S2:	congregation	government	
S3:	copulation	establishment	refusal

Jackendoff argues that the semantic relations between the forms in each row and their related verbs are the same (*discuss – discussion*, *argue – argument*, etc), although they are represented by different M-rules.

Noun compounds are described by using the following morphological and semantic rules (Jackendoff 1975:655):

M-rule (the morphological redundancy rule for noun compounds):

[8] Jackendoff (1975:652) adds that "phonological and syntactic conditions such as choice of boundary and existence of internal constituent structure can be spelled out in the redundancy rules".

$$\begin{bmatrix} /[_{\mathrm{N}}x]\ [_{\mathrm{N}}y]/ \\ +\mathrm{N} \end{bmatrix} \longleftrightarrow \left\{ \begin{matrix} \begin{bmatrix} /[x]/ \\ +\mathrm{N} \end{bmatrix} \\ \begin{bmatrix} /[\mathrm{y}]/ \\ +\mathrm{N} \end{bmatrix} \end{matrix} \right\}$$

S-rules:

$$\begin{bmatrix} +\mathrm{N} \\ Z \text{ THAT CARRIES } W \end{bmatrix} \longleftrightarrow \left\{ \begin{matrix} \begin{bmatrix} +\mathrm{N} \\ Z \end{bmatrix} \\ \begin{bmatrix} +\mathrm{N} \\ W \end{bmatrix} \end{matrix} \right\}$$

$$\begin{bmatrix} +\mathrm{N} \\ Z \text{ MADE OF } W \end{bmatrix} \longleftrightarrow \left\{ \begin{matrix} \begin{bmatrix} +\mathrm{N} \\ Z \end{bmatrix} \\ \begin{bmatrix} +\mathrm{N} \\ W \end{bmatrix} \end{matrix} \right\}$$

$$\begin{bmatrix} +\mathrm{N} \\ Z \text{ LIKE A } W \end{bmatrix} \longleftrightarrow \left\{ \begin{matrix} \begin{bmatrix} +\mathrm{N} \\ Z \end{bmatrix} \\ \begin{bmatrix} +\mathrm{N} \\ W \end{bmatrix} \end{matrix} \right\}$$

The lexical redundancy rules (M-rules and S-rules) limit the set of possible compounds in English, while it is up to the lexicon to list all “actually occurring compounds” (ibid.).

Actually occurring compounds may be difficult to list nevertheless, since part of the linguistic competence we model is the spontaneous creation of compounds and the ability to understand them. The following list contains examples of fully lexicalized and ad-hoc compounding[9]: *parent association, parent association committee, parent association committee chairperson, parent association committee chairperson scandal* and *parent association committee chairperson scandal video*. The last two compounds are not likely to occur in the reader’s mental lexicon, nevertheless, they are understandable. Lexical redundancy rules may explain this phenomenon. Jackendoff argues that redundancy rules are generalizations that are learned from lexical items that are already known to the speaker (p. 668). Redundancy rules make it easier to learn new items, and they also make it possible to coin (generate) new lexical items. “For example, the compound rule says that any two nouns N_1 and N_2 can be combined to form a possible compound N_1N_2. If the context is such

[9] I would like to give credit to Spencer (1991:48) for the original example “*student film society committee scandal inquiry*”, which he used to illustrate the recursive nature of compounding.

as to disambiguate N_1N_2, any speaker of English who knows N_1 and N_2 can understand N_1N_2" (ibid.).

The aim of Jackendoff's (1975) paper is to "provide a theory of the lexicon that would accommodate Chomsky's theory of the syntax of nominalization" (p. 669). He also states that the major innovation of his work is the novel use of redundancy rules as an evaluative tool. Jackendoff's idea is that *independent information content* can and must be measured so that we can compare implementations of the lexicon within a given theoretical model. This information content is tailored to match "our intuitions about the nature of generality in the lexicon" (ibid.).

Jackendoff outlines two hypotheses referred to as the impoverished-entry theory and the full-entry theory. In the former, the lexicon consists of full as well as *partial entries* that refer to fully-specified entries and the rule that produces their final form. For instance, the entry for *decision* takes the following form (p. 642):

/entry number/
derived from /entry number of *decide/*
by rule 3

Rule 3 is the derivational rule producing nouns using the *-ion* suffix in the above diagram.

The *full-entry theory* stores simple and complex words in separate, fully-specified entries containing the entry number, phonological representation, syntactic features and semantic representation. The following example is from Jackendoff (1975:642):

/decīd + ion/
+N
+[NP$_1$'*s* ___ *on* NP$_2$]
ABSTRACT RESULT OF ACT OF NP$_1$'S DECIDING NP$_2$

In this approach, redundancy rules do not routinely derive new forms: they establish a relation between entries, and play a part in the information measure for the lexicon.

Jackendoff prefers the full-entry theory. As an argument for this approach, he points out that there are 'derived' forms that lack their base-forms in (modern) English. There are no such forms as **aggress, *retribute,* or **fiss,* corresponding to *aggression, retribution, fission* (p. 645).

Jackendoff's lexical entries lack the [Lexical insertion] feature, which is utilized in Halle's (1973) theory. The explanation is the following: "... we have a lexicon containing merely a set of fully specified lexical entries (giving exactly those words that exist), plus the set of redundancy rules" (p. 645, emphasis added), thus the lexicon stores actual well-formed words only. There seems to be no difference between the storage mechanism for single-word (simple or complex) lexical items and compounds: "We ... give each actually occurring compound a fully specified lexical entry" (p. 655).

Although a proper treatment of idioms is beyond the scope of this chapter, I would like to briefly review Jackendoff's argument according to which his framework is able to accommodate them. While the words making up idioms are already in the lexicon, the meanings that idioms carry are independent of the meanings of their constituents. Jackendoff argues that it is therefore necessary to list whole idioms in the lexicon (p. 662). Note, however, that the above-the-word-level internal structure of idioms causes problems for lexical insertion. The following solution is outlined by the author:

> The lexical insertion rule will operate in the usual way, inserting the lexical entries onto deep phrase markers that conform to the syntactic structure of the lexical entries. Since the structure of the entries goes beyond the word level, the idiom must be inserted onto a complex of deep-structure nodes, in contrast to ordinary words which are inserted onto a single node.
>
> (Jackendoff 1975:662)

Jackendoff rejects a former assumption according to which the lexicon is memorized and uncreative (as opposed to syntax, which is fully creative). He explains that creativity and memorization are available in the syntactic component as well as in the lexical component, but the "normal mode" of operation is creative in the syntax and passive for the lexical component (p. 668). He adds that "[w]hen memorization of new lexical entries is taking place, the rules of either component can serve as an aid to learning" (ibid.).

The assumption according to which there is a difference in the default mode of operation of lexical and syntactic rules, which may be reversed during acquisition processes (which may include vocabulary extension as well as syntactic parameter setting), may in fact bring syntax and morphology closer.

> Is there, then, a strict formal division between phrase-structure rules and morphological redundancy rules, or between the semantic projection rules of deep structure and the semantic redundancy rules? I suggest that perhaps there is not, and that they seem so different simply because of the differences in their normal mode of operation.
>
> (Jackendoff 1975:668)

2.3.3 Verbal Compounds in Roeper and Siegel (1978)

Roeper and Siegel (1978) utilize a carefully designed set of Word Formation Rules (WFRs) augmented by lexical transformations. Lexical transformations, as described by Vergnaud (1973), affect the *contextual frames* of lexical items. Vergnaud's notion of contextual frames for verbs and adjectives is based on Chomsky (1965): it is "a frame wherein all nodes relevant for contextual features appear, as the PS rules generate them. For each node, the frame will indicate whether the verb or the adjective is positively specified for this node and, if so, what the selectional restrictions are that correspond to the node" (Vergnaud 1973:280).

This implementation of contextual frames brings a change to the structure of the lexicon: each entry must indicate whether it can undergo a certain lexical transformation. Vergnaud (1973) shows that the sequential application of lexical transformations to contextual frames is useful in derivation. Since WFRs – in Roeper and Siegel's interpretation – are supposed to shift syntactic category (cf. Roeper and Siegel 1978:202), inflection is not dealt with in this framework.

Roeper and Siegel focus on synthetic compounds[10] with one of the verbal affixes *-er*, *-ing* or *-ed*. Spencer (1991) argues that compounding "represents the interface" between syntax and morphology[11], and this is "particularly true of synthetic compounds" (p. 309).

The meaning of any element of this set of compounds, according to Roeper and Siegel, is fully compositional (therefore predictable), and is also in contrast with the meaning of what they call 'root compounds' (Roeper and Siegel 1978:206, also cf. Spencer 1991:319-324). As far as root compounds are concerned, the speaker/listener "cannot determine their semantic composition as a reflection of morphological composition"[12] (Roeper and Siegel 1978:206).

[10] A systematic description of synthetic compounds is in Spencer (1991).

[11] To support his argument, Spencer (1991) specifies the following features of compounding that resemble syntactic processes: 1) recursion (e.g. student film society committee scandal inquiry…), 2) the presence of constituent structure (e.g. [student [film society]] and [[student film] society]), and 3) relations between the elements of compounds resemble the relations between constituents (e.g. head-modifier). Compounds also exhibit word-like properties, e.g. they can lose their semantic compositionality, can undergo semantic drift, keep their morphological integrity, and their non-head elements are non-referential and typically uninflected (Spencer 1991:310-311).

[12] As far as the morphology of root compounds is concerned, they exhibit generalities, too. Williams (1981:249), extending his Righthand Head Rule to English compounds, observes that the righthand member of the compound determines the category of the whole. As described in section 2.3.4, the fourth Feature Percolation Convention in Lieber's (1981) assumes that many features of a root compound can be predicted, including category information.

The scope of Roeper and Siegel's work is shown by the following list of compounds (p. 199):

oven-clean**er**	from	clean ovens
jaw-break**er**	from	break jaws
late-bloom**er**	from	bloom late
strange-sound**ing**	from	sound strange
fast-act**ing**	from	act fast
well-buil**t**	from	buil*t* well
pan-fri**ed**	from	fri*ed* in a pan

They argue that *-ed* compounds are formed from infinitives rather than past participles. This phenomenon is taken care of by a special lexical transformation, an *adjustment* rule, formulated in this way (p. 210):

[verb] W ⇒ [[empty] + verb + ed]$_{Adj}$ W

Similar adjustment rules (Affix Rules) produce *-ing* and *-er* forms, too. Further adjustment rules handle the following phenomena:

– *-ed* compounds never incorporate direct objects: an obligatory rule (*Subcategorization Adjustment*) deletes the two subcategorization frames adjacent to the verb, namely the direct object and an adjectival or nominal complement (p. 210)[13];

– verbal compounds are produced by incorporation of a word in first sister position of the verb: the *Variable Deletion* adjustment rule is applied to avoid non-compliant cases.

The *Compound Rule,* which is treated as the main lexical transformation rule, takes the output of the adjustment rules as its input, and is formulated in this way (p. 209):

[[empty] + verb + affix][X_{+n} +word] W ⇒ [[+word] + verb + affix] W
1 2 3 4 5 4 2 3 ø 5

where W ranges over subcategorization frames and X_{+n} stands for lexical categories *N*, *A* and *Adv*. The expression *+word* is the result of the process called Subcategorization Insertion, which will be explained shortly.

Example: [[empty] + make + er] [$_N$ coffee] W ⇒ [[coffee] + make + er] W

Consider the following examples from Roeper and Siegel (1978:209):

– We think it's a shame.

– *it's-a-shame-thinker

[13] Bresnan (1982:32-37) argues that the *-ed Affix rule* and the *Subcategorization Adjustment rule* are unnecessary if we use the passive form of the verb, which is a lexicalized form in her LFG framework, instead

They point out that the compound is not permissible, because – as we have seen – sentential complements are excluded from the Compound Rule (X_{+n} stands for lexical categories *N, A, Adv* only). The authors add that "WFRs never involve phrases" (p. 209).

Since subcategorization frames do refer to phrasal categories, Roeper and Siegel must come up with a tool that converts a phrasal category into a word-level category. This device is the *Subcategorization Insertion*, which is implemented as another lexical transformation of the framework:

$[_{X''}$ empty$] \Rightarrow [_{X}$ +word$]$

thus "NP becomes N, AdjP becomes Adj, AdvP becomes Adv" (p. 211). The rule expresses a general principle which prevents phrasal categories from being introduced via the application of Word Formation Rules: "the Subcategorization Insertion rule selects a word from the lexical core and inserts it as a Noun, Adj, or Adv in the subcategorization frame" (p. 213). It also explains the fact that phrases are not incorporated into verbal compounds, a fact not accounted for by the transformational approach. Consider these examples (Roeper and Siegel 1978:213):

make [coffee] ⇒ coffee-maker

make [some good dark coffee] ⇒ *good dark coffee-maker

make [home] ⇒ homemaker

make [a home for the aged] ⇒ *home for the aged maker

Roeper and Siegel's example **beautiful day lover* also seems to fit into this framework seamlessly. What they point out as a problematic "subtlety" is the compound *?fast car lover* (ibid.). It may be more than an ad-hoc adjectival phrase but we may not want to categorize it as a compound or an idiom (in general, a lexicalized item), either. Roeper and Siegel refer to *fast car* as a *customary combination*, which results in "more acceptable" associated compounds.

The authors emphasize that the lexical account for compounding is supported by the fact that compounds permit further affixation (which is called the loop-effect). The following compounds exhibit external affixation:

– hardboiledness,

– heartrendingly,

– shopkeeperish,

while further affixes may be introduced internally, as in

of the infinitive, as a starting point of the derivation, i.e. the underlying form of *pan fried* is not *fry something in a pan* but *(something was) fried in a pan.*

- drowsiness-inducing
- widely-read
- funnyish-looking

(Roeper and Siegel 1978:214-215).

Further affixation of verbal compounds that incorporate adverbs are studied carefully by the authors. They make the point that those compounds that do not incorporate -*ly* adverbs take further affixation more freely than those that incorporate -*ly* adverbs. For instance, the following sentences are acceptable (p. 231):

- Their *well-suitedness* did not result in marriage.
- His *softspokenness* was an advantage.

The following sentences are ungrammatical:

- *Its extremely boringness was tolerated.
- *The strangely winningness of her smile baffled us.

Dubious cases are also listed (p. 232):

- ?The *carefully-consideredness* of her reply was evident.
- ?The *quickly-preparedness* of the lecture was obvious.
- ?The *rapidly-hiddenness* of these crimes was striking.

In the framework under scrutiny, the output of the Word Formation rules may or may not be stored in the *lexical core* (long-term memory) according to whether they are in common use or not (p. 200). For instance, *happiness* is listed in the lexical core while *expectedness* is not “until it is ‘invented’ in some appropriate circumstance and comes into general use” (ibid.). They argue, however, that “[w]ords with particularly frequent affixes could not all be listed in the core. For instance, the -*ly* adverbs are so numerous that it would be inefficient to remember each one.” (p. 204).

The core lexicon is bifurcated into an atomic and a complex part:

> The core lexicon must itself be divided into two parts: in one part (complex) we find the created words that have morphological structure (*happiness*). In the other part (atomic) we find words that have no morphological structure (*heart*). There is a large and interesting set of words that appear to fall between the two categories: *possible*, *happy*, etc. In Aronoff’s account, which we accept, these words must be listed as atomic words because they are not *compositional* in meaning. The output of every WFR must be a *word(s) plus an affix(es)* whose meaning is a rule-governed combination of its parts.
>
> (Roeper and Siegel 1978:200-201)

Semantic drift is described as a movement of words from the compositional section to the atomic section of the lexical core. This takes place when a new meaning is attached to the compositional form which, in that sense, loses its compositionality. For example the ‘roughly equal’ sense of *comparable* must be listed in the atomic core, while the meaning ‘able to be compared’ is compositional thus it belongs to the complex core (cf. Roeper and Siegel 1978:201).

As far as lexical entries are concerned, their basic structure is the same as the structure of the entries in Jackendoff’s (1975) (as a matter of fact, Roeper and Siegel cite Jackendoff’s examples). Pieces of semantic, phonological and syntactic information are all present (p. 201). Such a lexical entry will serve as the input for Word Formation Rules. WFRs can affect the subcategorization frames of lexical entries in the following three ways (Roeper and Siegel 1978:202):

- Frames are *inherited* from the base word used as input,
- Frames are deleted with the addition of certain affixes,
- Frames are added by redundancy rules.

The authors argue that the English prefix *re-* illustrates *frame deletion* since it does not occur with S-complements (the remaining frames are mostly inherited), as shown by the following data (p. 203):

- I started the car.
- I started to go outside.
- I restarted the car.
- *I restarted to go outside.

Frame addition is the result of the application of redundancy rules. A possible use of redundancy rules involves the addition of subcategorization frames appearing “automatically” with certain parts of speech: “[v]erbs have subjects or agent phrases; nouns may have prepositional phrases” (p. 203). The *full-listing* of full entries hypothesis Roeper and Siegel borrow from Jackendoff is weakened, however, when they state that the information added by these rules “need not be listed for every lexical item” (Roeper and Siegel 1978:203): this approach reduces redundancy but it also results in partial entries.

2.3.4 Word Formation in Lieber (1981)

Lieber’s (1981) model is based on a tripartite lexicon consisting of a permanent lexicon, lexical structure and the rules of word formation. Her system belongs to the *strong* *lexicalist* family: inflection and derivation are both handled within the lexicon. Lieber's study features examples from many languages, including Latin and German. These languages have allowed Lieber to gather evidence in favor of a lexicon containing both derivation and inflection: she points out that "the sorts of stem allomorphs according to which nouns, verbs, and adjectives in various inflecting languages form their plurals, pasts, participles, etc. often form bases for rules of derivation and compounding" (Lieber 1981:3). Lieber takes Modern German *Mutter* and *Mütter* as examples (Lieber 1981:33). The existence of the compounds *Mutterfreuden* 'maternal joy' and *Mütterverschickung* 'evacuation of expectant mothers' shows that compounding takes both stem variants as input. This phenomenon is "predicted and explained" in Lieber's framework, since both stems have their own entries in the permanent lexicon. German stem allomorphy is in fact a good source of examples. The following list is from Lieber (1981:14):

*Väter*sitte	'manners of our forefathers'
*Vater*land	'fatherland'
*Geist*buldend	'formative, educational'
*Geister*seher	'visionary seer'
*Aug*apfel	'eyeball'
*Augen*arzt	'eyedoctor'

If one takes the place of compounding within the lexicon for granted (which she does, cf. Lieber 1981:16), inflection must be handled in the lexicon, too, because all the necessary stem variants must be available for compounding.

Lieber also demonstrates that stems needed for inflection in Latin may also double as input for derivation (Lieber 1981:90):

- am+or ('love') is derivation on root (from amō),
- vatābundus ('avoiding') is derivation on theme vowel stem (from vitō), and
- finditō ('hurl, sling at') is derivation on nasal infix stem (from fundō).

Let us recapitulate that the lexicon in Lieber (1981) contains three subcomponents (p. 4):

- the permanent lexicon,
- lexical structure, and
- string dependent word formation.

She explains that the <u>permanent lexicon</u> consists of "lexical entries for all unanalyzable morphemes" (ibid.), called "lexical terminal elements" (Lieber 1981:35). The entries of the permanent lexicon (lexical entries) store idiosyncratic information about lexical terminal elements, including (Lieber 1981:35-36):

– their category and conjugation or declension class
– phonological representation
– semantic representation, which is said to be “unnecessary to be dealt with within the word formation component of a grammar” (ibid.)[14]
– affix subcategorization: affixes indicate the category of the input as well as the output items: for instance *re-* takes verb inputs and produces verb output[15]. Subcategorization frames can utilize category information (N, V, A, etc.) and diacritic information, see below.
– diacritics: synchronic or diachronic features that govern the productivity of certain affixes (e.g. [±Latinate] distinguishes verbs that take *-ive* from those that do not)
– insertion frames: the frames into which lexical terminals can be inserted (e.g. *-ize* forms verbs in English with two place argument structures, as in *the riot factionalized the city*)

Being morpheme-based, this system lists stems (inflectional and derivational stem allomorphs, with no subcategorization frames) and affixes (featuring subcategorization frames) in separate entries.

Lieber’s own sample entries include the following (Lieber 1981:37):

run (stem): (phonological representation)
semantic representation: ...
category: $_V[\_]_V$
insertion frame: NP_(NP)
diacritics: [+Latinate]

The next entry shows how subcategorization appears in entries of affixes (ibid.):

-*ize* (suffix): (phonological representation)
semantic representation: causative
category/subcategorization: $]_N\_]_V$
insertion frame: NP$_{(NP)}$
diacritics: Level II

[14] It is therefore not surprising to see that Lieber’s morphological processes will not “compute” semantic content. Criticising Aronoff’s ideas, she states that “there is no more reason to believe that semantics should be a part of the formal mechanics of word formation, than there is to suppose that semantics is a part of the formal mechanics of sentence syntax (i.e. phrase structure, transformations)” (Lieber 1981:65). Later she argues that it is in fact impossible to account for lexical semantic issues within the lexicon (pp. 65-70).

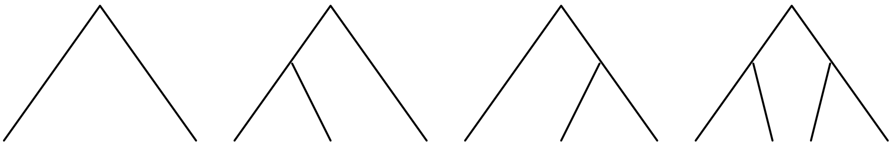
[15] This approach is supposed to account for affix ordering phenomena, too.

As far as the internal structure of the lexicon and the ordering of the entries are concerned, Lieber assumes that the lexicon is neither unordered nor alphabetically ordered (p. 38). She introduces the *category relation*, partitioning the lexicon into subsets according to universal and primitive categories. The inventory of these categories includes {N, V, A, ∅}, where ∅ stands for the category-less category (ibid.). This system might seem redundant, since lexical items contain category and conjugation/declension class information anyway, but it helps the author to set up morphological *paradigms* described by traditional grammar for Old English, Tagalog, Latin, Modern English and modern German. Moreover, category relations also allow her to relate stem variants like German *Mutter* and *Mütter*. This relation is introduced into the lexicon by the *Plural Umlaut morpholexical rule* (Lieber 1981:33). Lieber specifies the following features of morpholexical rules:

- “morpholexical rules do not relate members of a non-terminal vocabulary, but only terminal elements, i.e. members of pairs like (*Mann*, *Männer*), (*Geist*, *Geister*), which otherwise have equal status within the permanent lexicon” (p. 40);
- “it is purely arbitrary whether or not any lexical item conforms to the specifications of a given morpholexical rule, or ‘undergoes’ that rule; lexical items either conform to the relation specified by a morpholexical rule, in which case they belong to the class defined by that rule, or they do not” (p. 41).

Giving credits for Selkirk’s (1978) original idea of using context-free production rules to create a lexical structure into which the terminal symbols of the lexicon are inserted, Lieber implements a *second subcomponent* in her lexicon that she calls the lexical structure component. It contains “a single rewrite rule giving unlabeled structure”, and “insertion of morphemes into this structure subject to subcategorization restrictions, and node labeling” (Lieber 1981:48). While Selkirk uses a rule for inflectional morphology, another rule for compound formation, and multiple rules for derivational morphology, with an index on each non-affix non-terminal symbol which is either S for stem or R for root (the rules ensure that inflectional affixes are always on the outside[16], cf. Lieber 1981:44-45), Lieber’s rule uses no indexes for the symbols, which means that nodes are unlabeled in the resulting tree. She uses binary trees[17] to depict the structures her rule is supposed to produce, resulting in the following configurations (p. 47):

[16] The affix ordering Selkirk and others assume does not seem universal: Rice (1985) points out that in Slave (an Athapaskan language) derivational and inflectional prefixes are added to a verb alternately.



The unlabeled tree “skeleton” is filled with content using certain *labeling conventions*, which are the following (Lieber 1981:47):

– Convention I: a stem morpheme labels the first non-branching node dominating it[18].

– Convention II: an affix morpheme labels the first branching node dominating it.

Feature Percolation is responsible for labeling the remaining nodes. The Feature Percolation conventions are the following (Lieber 1981:49-50, 54):

– Convention I: All features of a stem morpheme including category features percolate to the first non-branching node dominating that morpheme.

– Convention II: All features of an affix morpheme including category features percolate to the first branching node dominating that morpheme.

– Convention III: If a branching node fails to obtain features by Convention II, features from the next lowest labeled node are automatically percolated up to the unlabeled branching node.

– Convention IV (Compounds): In compound words in English, features from the righthand stem are percolated up to the branching node dominating the stems.

Conventions I-III are symmetrical, i.e. features may percolate from either direction (left or right). This idea is in opposition to Williams’s claim: he assumes that his *Righthand Head Rules* (Williams 1981:248) can be applied to English syntax and morphology in a uniform fashion. This extension of the notion of syntactic heads to morphology is appealing, but morphological phenomena seem less systematic. Williams recognizes that the prefix *en*- must function as a left head (Williams 1981:249) and therefore it is a case of feature percolation from the left. Lieber (1981:58) adds the German prefixes *be*- and *ver*-, as well as a set of Vietnamese left-hand compounds, to the list of exceptions.

We have seen Roeper and Siegels’s contribution to the analysis of synthetic compounds in section 2.3.3. Lieber’s fourth Feature Percolation convention assumes that many features of a *root* compound can be predicted by rules. Notice that this convention applies to English only, and would exhibit the same problems as Williams’s theory of Right Heads when put universally. Also recall that Lieber’s entries for stems do not contain

[17] The rule should be applied recursively, so binary branching is not a limitation.

[18] As we have seen, stem morphemes lack subcategorization frames.

subcategorization frames, so root compound formation may require devices missing form Lieber (1981).

The permanent lexicon and the lexical structure component are the basis for affixation and compounding, and all “concatenative word formation processes” in general (Lieber 1981:4).

The *third subcomponent* in Lieber’s lexicon contains string dependent rules that take care of productive, but non-affixational morphological processes. Reduplication, infixing, vowel ablaut and umlaut are among the processes belonging to this class (Lieber 1981:62). She gives many examples, which are mostly taken from Tagalog and German. Lieber pinpoints the difference between affixation and the morphological processes realized by her string dependent rules in the following way: affixation is string concatenation, and does not bother with the “internal makeup” of strings, while there are processes, including Tagalog reduplication (that copies segments of the base), Tagalog infixing (reordering an affix with segments of its base) and the German umlaut, that exhibit “string dependent rules in that they must refer to segmental properties of the items to which they apply” (Lieber 1981:63).

2.4 Deconstructing Morphology: Lieber (1992)

Lieber’s (1992) approach does not recapitulate the Strong Lexicalist morphology detailed in her (1981) book. This new framework is based on the assumption that “there is no separate component of morphology in the grammar” (Lieber 1992:1), and it expels morphological processes from the lexicon.

Lieber’s reconsideration of her position is based on the examination of the following data taken from – as she puts it – the “fringes” of morphology[19]:

– phrasal compounds in English, Afrikaans, Dutch and German, e.g.
  the Charles and Di syndrome
  a pipe and slipper husband
– English possessive marking, e.g.
  $_{NP}$[a friend of mine]’s book
  $_{NP}$[a man I know]’s hat
– case marking in Warlpiri,
– adjectival possessives in Upper Sorbian,

[19] In her 1992 book, Lieber also devotes a whole chapter to non-concatenative morphology, including circumfixation, conversion, umlaut, reduplication and templatic morphology. These issues do not require “any special independent morphological principle” (Lieber 1992:195).

– verbal derivations in Tagalog, and

– *tal* nominalizations in Tamil.

A careful examination of the above phenomena has led Lieber to realize that “the rules of word formation are in fact the rules of syntax” (Lieber 1992:vii)[20].

Lieber has chosen the theory of Government and Binding as the host framework for her system. She offers a number of modifications that allow this framework to reach sublexical levels. Let us review the cornerstones of Lieber (1992).

Lieber makes use of Jackendoff’s X-bar theory with the refinements offered by Chomsky (1981, the Projection Principle) and Stowell (1981, parametrization, i.e. “each language makes a choice as to the position of the head with respect to its complements and specifiers”, and further refinements, see Lieber 1992:27-28). The basic principles of the version of the X-bar theory offered by Lieber are as follows (Lieber 1992:38):

a) $X^n \rightarrow \ldots X^{\{n\text{-}1\ \boldsymbol{or}\ \boldsymbol{n}\}} \ldots$, where *recursion* is allowed for n=0

b) Licensing Conditions[21]

i. Heads are initial/final with respect to complements.

– Theta-roles are assigned to left/right.

– Case is assigned to left/right.

ii. Heads are initial/final with respect to specifiers.

iii. Heads are initial/final with respect to modifiers.

c) Pre- or post-head modifiers may be X^{max} or X^0.

The template in a) is a variation of the basic X-bar template ($X^n \rightarrow \ldots X^{n\text{-}1} \ldots$), which captures the fact that phrases are endocentric and the head node is one level lower than the node immediately dominating it (p. 33). The modified template allows an X^0 category to dominate another X^0 category in a complex word.

[20] While she seems to find a unified morphosyntactic framework appealing and superior to previous approaches, she makes no attempt at proving that the Lexicalist Hypothesis is inadequate to account for the new data. Taking the set of phrasal compounds as an example, Ruszkiewicz pinpoints that rules of the sort $N \rightarrow XP\ N$ are not forbidden in a lexicalist framework (Ruszkiewicz 1999:259). This rule is not even incompatible with Lieber’s (1981:8) restrictions on the interaction between syntax and morphology (stating that *no syntactic rule can refer to elements of morphological structure*). Lieber’s (1992:19) position is different, however: she states that the use of “some sort of loop from syntax to morphology which feeds phrases back into the word formation component ... effectively undermines the Lexicalist Hypothesis”.

[21] The Licensing Conditions for English are the following (Lieber 1992:54):

a) Heads are initial with respect to complements.

b) Heads are final with respect to specifiers.

c) Heads are final with respect to modifiers.

Licensing Conditions should be set “just once for each language”, and “they apply both above and below word level” (Lieber 1992:39). This leads us to the next topic, i.e. the positioning of heads.

Tagalog has some right-headed word structures (five category-changing affixes are mentioned, see Lieber 1992:283), which are disallowed by the Licensing Conditions for that language[22]. Lieber comes up with the following solution: these base verbs and base nouns are predicates rather than specifiers, modifiers, or complements. The Licensing Conditions do not regulate the relative ordering of heads and predicates. Unfortunately, Lieber remains mute on the question of how predicates could be properly incorporated into her system.

Lieber makes use of transformations to account for the observation that the relative ordering produced by the Licensing Conditions does not always reflect surface structure configurations. The following are examples from Lieber (1992:53):

– a man *alone*

– a man *bruised and battered*

These phrases contain “heavy” modifiers (according to her explanation “anything with a complement should count as heavy”, Lieber 1992:53). They are generated in a pre-head position (to satisfy the Licensing Conditions) then the *Heavy NP Shift transformation* moves them to the right. Ruszkiewicz (1997:285) points out that the well-formedness of the NP ‘a *bruised and battered* man’ is correctly predicted by the fact that Heavy NP Shift is optional. However, since the phrase ‘*an *alone* man’ is ungrammatical, we must look for conditions that make Heavy NP Shift obligatory. Ruszkiewicz explains that *alone* must be moved because it is a predicative adjective.

Lieber uses the following notions of GB (the relevant definitions are in Chomsky 1986 and Baker 1988, also cited by Lieber 1992:141) to predict and account for movement at both lexical and sublexical levels: Government, ECP, Theta-GOV, Blocking Category, Barrier, L-Marking, Head Movement Constraint (HMC, see, for instance Baker 1988:53).

Lieber points out that a modification to ECP is required to prevent morphemes from moving out of the words that contain them. Thus the original version of the Empty Category Principle (a trace must be properly governed; *A* properly governs *B* if *A* Theta-governs or antecedent-governs *B*) is extended by adding a further condition: *A* properly

[22] Licensing Conditions for Tagalog (Lieber 1992:282):
a) Head initial with respect to complements.
b) Head initial with respect to modifiers.
c) Head initial with respect to specifiers.

governs *B* iff *A* Theta-governs and L-marks B, or antecedent-governs *B* (Lieber 1992:142). Since L-marking is expressed in terms of phrasal categories (X^0 L-marks a YP to which it assigns a Theta-role or Case), the revised Empty Category Principle properly constrains movements. Ruszkiewicz (1997) underlines, however, that “formulating constraints on morphological structures in terms of phrasal categories, which can never be met, undermines one of Lieber’s basic tenets, namely that syntactic and morphological structures are governed by the same general principles” (Ruszkiewicz 1997:315).

We have seen in the previous section that the tripartite system of Lieber’s (1981) lexicon consists of the permanent lexicon, lexical structure and string dependent word formation rules. Since morphological and word formation processes, as well as hierarchy building are accounted for by a somewhat revised machinery of the GB theory, the lexicon is now deprived of lexical structure and the rules of word formation. What remains is the permanent lexicon, which lists morphemes (both free and bound, Lieber 1992:22). The lexical entry of each morpheme contains:

– a phonological representation,
– a semantic representation in the form of Lexical Conceptual Structure (LCS),
– syntactic category,
– many entries contain a Predicate Argument Structure, constructed from the LCS to form “an explicit representation of hierarchical relations between the verb and its arguments” (Lieber 1992:118)[23], furthermore,
– bound morphemes have “an indication of their morphological subcategization, that is, the category of the items to which they attach” (ibid.).

Lieber’s own sample entries are reproduced here for your reference (Lieber 1992:22, the entry of *PUT* is added on the basis of Lieber (1992:118), the LCS of *-ize* is modified on the basis of Lieber 1992:119):

a. words **run** [$_{\text{V}}$ ___]
[*phonetic transcription*]
LCS: [$_{\text{EVENT}}$ GO ([$_{\text{THING}}$], [$_{\text{PATH}}$])]
PAS: x

enter [$_{\text{V}}$ ___]
[phonetic transcription]
LCS: [$_{\text{EVENT}}$ GO ([$_{\text{THING}}$], [$_{\text{PATH}}$ TO ([$_{\text{PLACE}}$ IN ([$_{\text{THING}}$])])])]

[23] As an alternative definition, the Predicate Argument Structure “will give the mapping between LCS and syntactic structure” (Lieber 1992:22)

PAS: x <y>

cat [$_{\text{N}}$ ___]
[*phonetic transcription*]
LCS: [$_{\text{THING}}$]

put [$_{\text{V}}$ ___]
[*phonetic transcription*]
LCS: [$_{\text{EVENT}}$ CAUSE ([$_{\text{THING}}$], [$_{\text{EVENT}}$ GO ([$_{\text{THING}}$], [$_{\text{PLACE}}$ AT [$_{\text{PLACE}}$]])])]
PAS: x <y, P$_{\text{loc}}$ z>

b. affixes **-ize**]$_{\text{N,A}}$ ___]$_{\text{V}}$
[*phonetic transcription*]
LCS: [$_{\text{EVENT}}$ CAUSE ([$_{\text{THING}}$], [$_{\text{EVENT}}$ BE (LCS OF BASE)])]
PAS: x

un- [___ $_{\text{A}}$[
[*phonetic transcription*]
LCS: negative

c. roots **path** [X [$_{\text{N}}$ ___]] or [[$_{\text{N}}$ ___] X]
[*phonetic transcription*]
LCS: ... *[this ellipsis is by Lieber]*

d. lexicalized words **transmission** [$_{\text{N}}$ ___]
[phonetic transcription]
LCS: [$_{\text{THING}}$ part of car]

e. lexicalized phrases and sentences **to kick the bucket**
The cat is out of the bag.
...

Lexical Conceptual Structure decomposes the meaning of a word into semantic primitives e.g. CAUSE, GO, COME TO BE IN STATE (Lieber 1992:118). Following Levin and Rappaport (1986) and Rappaport and Levin (1988), Lieber argues that Theta-roles such as *agent*, *theme* and *goal* are just convenient labels for LCS argument positions rather than linguistic primitives (Lieber 1992:118)[24]. Predicate Argument Structures do not represent Theta-roles anymore. In the present framework, syntactic rules do not refer to particular Theta-roles, only to positions in PAS (cf. Lieber 1992:118).

[24] Studies, including recent ones, do not readily reject the *theme* account of (morpho)syntactic phenomena. Laczkó (2000), for instance, argues that the formation of adjectival passives, as well as the "description of the

The semantic change caused by morphosyntactic processes is reflected by Lexical Conceptual Structures. Changes in LCS will result in a subsequent change in PAS, see the entry for the suffix *-ize* (the CAUSE semantic primitive is added – while the LCS of the base is preserved – and an external argument position is projected to the PAS of the resulting verb).

Lieber makes the following note on sublexical binding:

> Whether coreference [with a sublexical element] is possible or not seems to depend on a number of factors including the nature of the sublexical elements (basically whether it is a name or not), its structural position in a word (whether it is head or not), and, most important, whether or not it occurs in a word that is derived productively.
> (Lieber 1992:129)

As far as sublexical binding is concerned, Lieber (1992:131) suggests that words with less productive[25] and fully unproductive affixes should be listed in the lexicon. The following derivational affixes belong to this set (ibid.): *-ary* (forming nouns), *-ship* and *-age*. Lieber’s aim is to prevent her indexing mechanism from accepting sentences like ****Mission**aries don’t often go **there*** (Lieber 1992:131, emphasis added to indicate undesired coindexing). Under this alternative, words like *missionary, dictatorship, orphanage* are “unanalyzable wholes with respect to syntactic rules and principles” (p. 131): no sublexical structure is assigned to such a word, therefore coindexes cannot be applied. Examples of more productive affixation, where binding below X^0 is permitted (p. 129):

– ***Bush**ians admire **him** greately.*

– *Their jam has a **fruit**y flavor, because they use so much of **it**.*

In a different context, Lieber provides us with data on the productivity of English affixes (p. 6-7), computed using the following formula: Productivity = number-of-hapax-legomena / number-of-tokens-of-the-affix. Her table lacks information, however, on the productivity of the suffixes *-ary*, *-ship*, and *-age*, which means that the above examples cannot be evaluated using the productivity figures. Furthermore, as Ruszkiewicz (1997:305) points out, “words made with affixes possessing clear-cut category-selectional and category-forming properties and attached to lexical (i.e. X^o) bases cannot be denied internal structure in a selective fashion”. We miss Lieber’s explanation of how and when

necessary but not sufficient condition on the optionality of an oblique argument of a predicate” need some kind of a thematic generalization (Laczkó 2000:109).

[25] In Lieber’s terminology, productivity of a derivational affix correlates with the possibility of an ad-hoc coinage of new words with that particular affix.

these derived forms should emerge in the lexicon. Also note that Lieber (1992) argues for storing lexicalized phrases and sentences in the lexicon (e.g. *'to kick the bucket'*, *'The cat is out of the bag'*, p. 23), but these lexicalized items remain syntactically unanalyzed[26].

Lieber's original (1981) concept of feature percolation (as introduced in the previous section) needs some modifications, too. In the 1981 model, all features of a stem or affix morpheme are allowed to percolate. In the present framework, only morphosyntactic features percolate. Diacritic features are excluded from this process at this time, the supporting evidence is from Latin, French and German morphology (Lieber 1992:80-86). Argument structures do not seem to follow the rules of percolation, either: argument structure will be *inherited* rather than percolated[27].

A notable difference between percolation and inheritance is that morphosyntactic features do not percolate across categorial lines; while argument structures may be inherited in that way. "[S]ince the argument structure of a nominalization like *destruction* is clearly related to the argument structure of the verb *destroy* from which it is derived, we have every reason to believe that the argument structure of *destroy* is inherited and subsequently acted upon by the nominalizing suffix *-tion*" (Lieber 1992:88). Lieber does not treat this question as pertinent to her analysis of morphological processes, so she only lists a collection of relevant data in passing. They include (Lieber 1992:117-119):

- the English process *-ing* affix, which has no argument structure of its own: the argument structure of its base is inherited entirely,
- the English suffix *-er*, which absorbs the external argument of its base verb,
- the English causative suffix *-ize*, which adds an argument (e.g. *"Gambling is legal."* vs. *"<u>They</u> legalized gambling."*),
- the English adjective-forming suffix *-able*, which eliminates the external argument of the base, and externalizes its internal argument (e.g. *"They washed the socks."* vs. *"The socks are washable"*), and
- the English affix *-ful* blocks the assignment of all internal arguments (e.g. *"They hoped for rain"* vs. **"They are hopeful for rain"*).

The phenomenon of morphosyntactically motivated LCS modifications and PAS inheritance is hardly dealt with by Lieber, and more thorough investigations are in place. An example of this is Laczkó (1998) presenting a study of deverbal nouns. In his analysis, a derived nominal belonging to the "ordinary process, event nominals" (p. 235) semantic

[26] The lack of compositionality of meaning does not justify this decision unless an isomorphism between meaning and form is postulated.

group (e.g. *running*, *arrival* and *examination*, p. 219), as a rule, “entirely inherits not only the LCS but also the PAS of the input verbal predicate” (p. 219). Institutionalized event nominals do so much less frequently (p. 236), while result nominals exhibit this behavior very rarely (ibid.). To complicate matters, the same noun (e.g. *examination*) can be used in all three readings, rendering it “ambiguous between a lexical entry with a PAS and another without it” (p. 235). When this inheritance does take place, certain (or all) arguments may become optional in previous analyses. Laczkó’s proposal (based on empirical data from Hungarian and English) is the following:

> ... in the case of PAS inheritance by derived nominals only one argument may be absent, “on the surface”, from the NP: the argument which is typically expressed by the possessor constituent (but I assume that even this argument is present in the structure in the form of a PRO). All the other arguments in the PAS of the nominal are as obligatory as the arguments in the PAS of the input verb.
>
> (Laczkó 1998:233)

Let us now return to the percolation of morphosyntactic features. Lieber introduces the notion of *categorial signature*. Categorial signatures are set up for syntactic categories, and they contain morphosyntactic features only. As she puts it: “[t]he categorial signature is a frame of morphosyntactic features headed by the category features [±N], [±V] that are of syntactic relevance for a particular category in a particular language” (Lieber 1992:88-89). The categorial signature proposed for English nouns is the following (Lieber 1992:89):

$$\begin{bmatrix} \text{N} \\ \pm\text{Plural} \\ \pm\text{I} \\ \pm\text{II} \end{bmatrix}$$

The features I and II are interpreted in the following way: [+I,-II] = first person, [-I,+II] = second person, [-I, -II] = third person. Unlike in German, gender is not part of the categorial signature of nouns in English, but it may be present in the LCS.

In Lieber’s framework, percolation has two distinct stages (Lieber 1992:92):

a. Head Percolation

   Morphosyntactic features are passed from a head morpheme to the node dominating the head. Head Percolation propagates the categorial signature.

[27] Argument structure may not even be “factored into” binary features at all (Lieber 1992:80,87).

b. Backup Percolation

If the node dominating the head remains unmarked for a given feature after Head Percolation, then a value for that feature is percolated from an immediately dominated nonhead branch marked for that feature. Backup Percolation propagates only values for unmarked features and is strictly local.

Head Percolation actually causes the transfer of a categorial signature from a head morpheme to the node dominating it. Backup Percolation fills in values of features that are unmarked after Head Percolation.

Lieber argues that only the following objects should have full categorial signatures (p. 112):

– stems
– bound bases
– derivational affixes

During word derivation, the features of the head morpheme will override the features of an inner morpheme, moreover, the "categorial signatures of derivational affixes (or stems) will be percolated by Head Percolation if these morphemes are heads" (p. 112).

Inflectional morphemes do not override features, all they do is fill in empty slots (unspecified values) in the categorial signature of the base form. "Inflectional word formation is therefore *additive* in a way that derivational word formation and compounding are not. A corollary of this is that while derivational affixes may or may not be heads of their words, inflectional affixes will never be heads... The features of inflectional morphemes will only be affected by Backup Percolation" (p. 112).

Ruszkiewicz (1997:299) points out that since inflection is carried out by suffixes in English, inflectional structures will be left-headed. It is a violation of the Licensing Conditions for English, unless we analyze inflectional affixes as complements.

Despite all the criticism, Lieber's enterprise of constructing a system that is sophisticated enough to account for an abundance of morphological phenomena, integrated into a widely accepted syntactic framework, is remarkable. It is interesting to see, however, how dramatic the differences are between Lieber (1981) and Lieber (1992), while the empirical data that motivated the change is labeled "exotic" by the author.

The goal of this chapter has been to illustrate that the role of the lexicon and the function of syntax have been treated as related questions in the generative tradition. Those who accept the *strong lexicalist* hypothesis (advocated by Halle 1973 and others) see morphology as completely detached from the syntactic component. The resulting lexicon is not simply a storage space for idiosyncrasies, but it has morpholexical functions and processing responsibilities. If the *weak lexicalist hypothesis* is accepted, a division of labor must be postulated between the lexicon and syntax from the point of view of morphological processes: inflection is treated by syntax, while less systematic (less productive or "quasi-productive") processes end up in the lexicon. Finally, accepting Lieber (1992) means that morphology is non-existent in the lexicalist sense, and the role of the lexicon is demoted accordingly.

2.5 Morphology and NLP

In the practice of Natural Language Processing, *syntactic parsing* has attracted compelling and significant research. Grammars developed for parsing are usually quite different from the transformational grammars that underlie the models presented in the first part of this chapter, because computational linguists are seeking devices that are *efficient*, too, while this is not an issue in theoretical linguistics. Prószéky (1989:53) points out that these two paths of seeking appropriate grammars remained completely isolated through the end of the 1970s. Unfortunately, the literature of morphosyntactic parsing does not seem to have been proliferating. Shaban (1993) introduces a GB parser that is accompanied by a word-based lexicon (which includes syntactic category, subcategorization frames and other features), but the system lacks a morphological analyzer. "The likely solution to this is to integrate PC-Kimmo" (Shaban 1993:21; note that PC-Kimmo implements two-level morphology, which is a non-transformationalist model, cf. section 2.5.2). Meyers (1994) also presents a GB parser that uses a lexicon containing independent words only (p. 75).

Kashket (1986) is one of the few exceptions. He outlines a GB parser for Warlpiri, a language with free word order. His system uses morphological case information to identify syntactic arguments in the "lexical parsing" stage (working with a lexicon that lists morphemes), then the same parser engine carries out syntactic parsing to assemble the final phase marker from the output of the first stage. Consider the following example sentence:

Ngajulu-rlu ka-rna-rla punta-rni kurdu-ku karli.
I-ERG PRES-1-3 take-NPST child-DAT boomerang
'I am taking the boomerang from the child.'
(Kashket 1986:60)

Kashket explains the process of *argument identification* in the following way:

> The dative case-marker, *ku*, selects its preceding sibling, *kurdu*, for dative case. Once co-projected, the dative case-marker may then mark its selected sibling for dative case. Because *ku* is also a case-assigner, and because *kurdu* has already been marked for dative case it may also be assigned dative case. The projected category may then be linked to dative case by *punta-rni*, which links dative arguments to the source thematic (θ) role because it has been assigned dative case.
>
> (Kashket 1986:61, emphasis added)

The author highlights that it is only the process of selection which is configuration-dependent (also noting that Warlpiri is a head-final language), but case-marking, case-assignment and argument-linking are “not directional” in Warlpiri. The corresponding lexical items are specified in the following way (Kashket 1986:61):

```
(KU       (action (assign dative))
          (action (mark dative))
          (action
                  (select (dative ((v . -) (n . +)))))
          (datum (case dative))
          (datum (percolate t)))
(KURDU    (datum (v -))
          (datum (n +)))
```

Lexical lookup involves searching for the right morpheme in the lexicon and returning the value (i.e. data and actions[28]) associated with it.

The author admits that the system cannot cope with ambiguity: morphemes must be unambiguous (p. 63). Whether this deficiency can be solved seems to remain an open question, since to my best knowledge, Kashket’s solution has not attracted significant attention in the linguistic literature.

I would like to identify three possible reasons why NLP implementations do not usually handle both morphological and syntactic phenomena. Firstly, we often design complex systems using the ‘divide et impera’ or divide and rule approach: it is easier to handle complex phenomena if we can identify sub-problems and delegate them into different modules (see Shaban’s ‘solution’ to morphological analysis as described above), especially at the level of system design, where intermodular interface problems are less apparent. Secondly, NLP has its own concerns that are irrelevant or may even run counter to the preferences of theoretical linguistics. Such a concern is the generative power of the

[28] I notice a parallel between this formulation of the lexicon and some properties of object-oriented programming, where objects have attributes and we can also define behavior (“methods”) for them.

grammar we use. Prószéky (1989:53-60) surveys the NLP-related considerations that determine the attributes of appropriate grammars from the point of view of computational linguistics. He also presents a comprehensive overview of grammars produced in the framework of NLP (pp. 63-203), whose survey is certainly beyond the scope of this chapter. Finally, the typical goal of NLP has been to serve the needs of the English language, and as far as inflection is concerned, fully listing English inflected forms is 'feasible'[29]. Morphologically richer languages and languages with free word order (see Kashket's Warlpiri parser), may require extra care from a syntactic and/or morphological point of view.

In the remaining part of this section, let me discuss the perspective of NLP on morphological processing in some more detail.

2.5.1 Morphological disambiguation

Karlsson and Karttunen (1996) make a distinction between *morphological "disambiguation"* and *morphological analysis*. Morphological disambiguation identifies the grammatical category of words using probabilistic or rule-based methods. The authors refer to the **CLAWS** (Constituent-Likelihood Automatic Word-tagging System) tagger as an example of the probabilistic type. It has a tagging accuracy of 96-97% (cf. Garside, Leech and Sampson 1987). Rule-based systems include Atro Voutilainen's **EngCG2** (English Constraint Grammar version 2) tagger. Voutilainen's (1997) system contains the following modules, which are applied to the input sequentially[30]:

- Tokenizer: identifies words and utterance boundaries.
- Morphological analyzer: all possible analyses are assigned to each word using a lexicon and a rule-based "guesser" for unknown words. On average, the system produces about 1.7-2.2 different analyses for each word (cf. Samuelsson and Voutilainen 1997).
- Disambiguator: "eliminative linguistic rules" are applied to the sentence to filter out impossible combinations.

Approximately 4000 rules are used, which are validated by a 0.7 million-word test corpus. Rules are classified into five classes according to their reliability, which makes it

[29] That is, it is not more difficult than listing all stems, which remains an issue in all languages (this phenomenon is tackled in chapters 2, 3 and 4 of this thesis).

[30] Please note that this system is behind Connexor's commercial NLP solutions (http://www.connexor.com/).

possible to control the error rate (versus remaining ambiguity). The author reports a precision level above 99% (cf. Voutilainen 1997, Samuelsson and Voutilainen 1997).

Voutilainen notes that the tagged output of his system can be used as input for syntactic parsing. Karlsson and Karttunen (1996) also note that syntactic parsers normally work with the result of morphological disambiguation or morphological analysis. While morphological disambiguation (POS tagging) is primarily used to support syntactic parsing, morphological analysis gives deeper understanding of the internal structure of words, therefore, it has more diverse linguistic uses e.g. in spelling checking, hyphenating and supporting translation (cf. Prószéky and Kis 1999:266-267).

2.5.2 Morphological analysis

In Karlsson and Karttunen's terminology, *cut-and-paste morphological analysis* means affix stripping, i.e. the process of removing affixes from a word. Although Karlsson and Karttunen do not refer to it, I would like to mention Porter's well-known affix stripping algorithm (Porter 1980), which is often referred to in the relevant literature. See section 4.1.2.1 of this thesis for a description of **Morphy**, which is a software tool that carries out affix-stripping for WordNet. Karlsson and Karttunen point out that the **Decomp** module of the MITalk text-to-speech system is a cut-and-paste implementation (it dates back to the 1960s; cf. Allen, Hunnicutt and Klatt 1987); the example of Decomp shows that morphological analysis has an important role in speech applications, too.

Karlsson and Karttunen pinpoint that *finite-state networks* can be used to analyze and also to *generate* word forms. The most successful model that uses finite-state networks is probably Kimmo Koskenniemi's *two-level morphology* (cf. Koskenniemi 1983)[31]. In this model, finite-state automata are used to align surface and lexical level symbols that correspond to single phonemes. Karttunen (2001) points out that rules are applied in parallel, and there are no intermediate levels of derivation, only a surface level and an underlying lexical level. Symbols are not rewritten to other symbols; instead, two-level rules relate symbols in a one-to-one fashion on the surface level and the lexical level. The length of the surface and lexical level representations should match ("zero" symbols can be used as padding devices to compensate for length mismatches). It is possible to use two-

[31] Richie (1992) shows that two-level models can generate regular languages, which also means that their generative strength may not be enough to handle a full variety of natural language phenomena; but note that finite-state automata (that exhibit the same generative strength) have been popular in computational linguistics nevertheless.

level rules to produce surface forms from underlying lexical forms, or vice versa. Karttunen (2001) also points out that the application of all matching two-level rules to surface or lexical level forms results in massive “overanalysis”, which is remedied by the use of the lexicon as a filter in each step of the analysis. Karttunen specifies the following main properties for two-level morphology:

- Rules are symbol-to-symbol constraints that are applied in parallel, not sequentially like rewrite rules.
- The constraints can refer to the lexical context, to the surface context, or to both contexts at the same time.
- Lexical lookup and morphological analysis are performed in tandem.

(Karttunen 2001)

Karttunen also pinpoints that the inviolable nature of two-level rules is a limiting factor, and “perhaps we will see in the future a new finite-state formalism with weighted and violable two-level constraints” (Karttunen 2001).

As far as actual software implementations of two-level morphology are concerned, **PC-KIMMO (**which is apparently named after Kimmo Koskenniemi) is the best-known one (cf. Antworth 1990). A PC-KIMMO morphology of Turkish, which uses 22 phonetic rules and about 23000 words in its lexicon, is presented in Oflazer (1993). A free (GPL-licensed) implementation of two-level morphology is available as a Debian Linux package (*mmorph*). Karttunen (2001) points out that two-level morphology has been exploited in numerous NLP systems, including the *Alvey project* (Black et al. 1987), the *CLE Core Language Engine* (Carter 1995), and the *ALEP Natural Language Engineering Platform* (Pulman 1991).

It must be noted that *all acceptable parses* are computed during the two-level analysis process, that is the mismatches between surface and lexical forms are described in all possible ways, which leads to ambiguity that can only be resolved later. Koskenniemi (1990) introduces a system in which the output of the two-level morphological analyzer contains “analyzed words with all interpretations (and all possible syntactic functions)” (p. 229), which is disambiguated by a “local disambiguator” (which contains restrictions on compounds, removes duplicate analyses and also contains weighting methods to identify the most probable parses) and a finite-state syntax (ibid.).

2.5.3 Humor and HomorESK: Morphosyntactic analysis in language technology

The dominatingly isolating nature of the English language does not encourage linguists to pay too much attention to agglutinative or fusional natural language phenomena, but morphologically more complex languages, including Hungarian, may require morphological (pre)processing. NLP systems for these languages may use an implementation of two-level morphology, or any other model that is powerful enough to handle morphological processes. Prószéky (1994) introduces a morphological analyzer called **Humor** (High-speed Unification Morphology), which can be used as a morphosyntactic analyzer, too (cf. Prószéky 1996, see below). When used as a morphological analyzer, the system can process agglutinative and other highly inflectional languages (e.g. Hungarian, German). The author reports that the system can generate 2.000.000.000 well-formed Hungarian word forms using a lexicon of about 90.000 *stems*, while the "*suffix* dictionaries contain all the inflectional suffixes and the productive derivational morphemes of present-day Hungarian" (Prószéky 1994:214, emphasis added). Prószéky and Kis (1999) point out that the Humor 99 system handles *affix arrays* (e.g. complex endings) as atomic strings[32]. Their example is the English *-ers'* affix array, which contains derivational and inflectional suffixes. All base variants (allomorphs) should be pre-listed (or pre-generated during a learning phase), e.g. *wolf / wolv*, *happy / happi*. Concatenation of stems and affixes, including the concatenation of multiple stems and multiple affixes, should be licensed by paradigm descriptions (e.g. if *green*:{Deriv=Abstr, Deg=Comp, Deg=Super}, *-er:*{Deg=Comp}, *-ness:*{Deriv=Abstr} then the forms *greener* and *greenness* are licensed; cf. p. 263). Concatenation licensing is also done by checking the unifiability of feature sets associated with the forms in the lexicon. E.g. *green* and *-er* as defined below are unifiable; therefore, they are licensed to undergo concatenation (the following descriptions are from Prószéky and Kis 1999:264):

green: [Cat=Nom, Lex=Base, Subcat=Adj, Deriv=Abstr, Deg={Comp, Super}]

-er: [Cat=Nom, Subcat={Adj,Adv}, Deg=Comp]

Prószéky (1996) argue that *Humor* can be extended to carry out syntactic analysis, too (this system is called **HumorESK,** which stands for "Humor Enhanced with Syntactic Knowledge"). Having completed the morphological analysis phase, the system switches to

[32] Prószéky and Kis (1999) also argue that handling affix arrays as atomic strings has psycholinguistic relevance.

a lexicon that stores words[33] containing meta-letters that stand for grammatical categories, and starts to analyze the sentence. Observe the author's example below (pp. 1123-1124):

Input sentence:	*The dog sings.*
Morphological analysis:	*The*[DET] *dog*[N] *sings*[V]+[3SG] .[END]
Category codes only:	DET N V 3SG END
Input for syntactic analysis:	dnvxe

The letters in the word "dnvxe" correspond to the category labels produced for the words of the original sentence in the morphological analysis phase. The lexicon that Humor uses during syntactic analysis is a precompiled grammar for the given language (p. 1124), prepared in a way that each possible sentence arrangement corresponds to a single word in the lexicon. This grammar contains "level 1" and "level 2" rules. Level 1 rules help to form phrases, e.g. the string *dn* (DET+N) is accepted by a level 1 rule, and the result is *dn*[m], where *m* is a new symbol that labels an NP. In the next step, a level 2 rule accepts *mvxe* producing *mvxe*[S]. No backtracking is possible, so category symbols (e.g. *m* in the above example) are meta-letters on *higher* levels (p. 1124). Prószéky (1996) points out that it is possible to fine-tune the system by using feature-unifiability checking for which we can associate *features* with the meta-letters.

To facilitate the HumorESK analysis of a language, a preparatory tool must be used to create the *largest regular subset* of a context-free language that describes the given language. This regular subset stays within the *string completion limit* (SCL), which imposes an upper limit on "the number of branches in the longest path from a non-accepting state to an accepting one" (p. 1124). The result is a finite-state automaton (ibid.). By adding two more conditions (the *length of the output string* is limited and the *maximum number of passing the same branch* is also specified), we can extract finite descriptions from this FSA (p. 1124). The result is stored in the lexicon as a set of pseudo-words (containing category meta-letters) corresponding to sentences. The above procedure limits the generative power of the grammar, but in the context of a language technology application, where we do not carry out "in vitro" modeling of linguistic competence but "in vivo" language processing, that is we work with authentic language samples that have been affected by performance issues anyway, imposing such limitations should not cause major difficulties.

Modeling the morphosyntactic behavior of languages, which has been our primary focus in the present chapter, is but one issue that is relevant to lexicon design. The next chapter introduces the perspective of *representing meaning* in a lexicon.

[33] These pseudo-words correspond to entire sentences.

3. SENSE DELINEATION AND THE POLYSEMY - HOMONYMY DISTINCTION

3.1 Introduction

Any kind of linguistic project that involves the enumeration and delineation of word senses (including the compilation of dictionaries and the development of lexicons for NLP applications) must account for many issues related to word meaning, including *polysemy*[34] and *homonymy*. This chapter concentrates on theoretical considerations, even when we take the perspective of NLP and traditional dictionary design in section 3.6, but practical issues are not neglected, either: chapter 4 contains an analysis of how homonymy appears in WordNet, as well as a brief survey of MindNet, which claims compatibility with some of the ideas presented here.

3.2 Polysemy versus homonymy

The polysemy-homonymy distinction is clear and unproblematic for the first sight. **Homonyms** are unrelated words that share the same spoken and written form[35], while a word that has two or more different, but related meanings is **polysemous**. Polysemy is exemplified by the word *bulb*, which can refer to “the root of a plant”, as well as “an electric lamp”. The similarity of their shape leads to relatedness in meaning, therefore these two senses are said to be connected to the same lexeme, which is polysemous.

Well-known examples for homonymy are money *bank*$_1$ and river *bank*$_2$. Some linguists, including Verspoor (1997) disagree with this straightforward categorization, pointing out that the “financial institution” sense is related to the “riverbank” sense since it was the riverbank where bankers were available: “going to the financial institution meant going to the edge of the river, hence to the *bank*” (Verspoor 1997:215). Lyons (1995) points out,

[34] More complete reviews on polysemy include Kilgarriff (1992) offering an NLP approach and Pethő (2001).

[35] Lyons (1995) makes a distinction between absolute and partial homonymy. *Absolute homonyms* satisfy all of the following conditions (Lyons 1995:55):

– they will be unrelated in meaning;
– all their forms will be identical;
– the identical forms will be grammatically equivalent.

In cases of *partial homonymy*, one or two of the above conditions are satisfied in addition to the identity of at least one form (ibid).

however, that these two senses of *bank* are etymologically unrelated: *bank*$_1$ is a 15th century Italian borrowing, while *bank*$_2$ originates from a Scandinavian word[36] (Lyons 1995:28).

The *bank* example shows that making the polysemy - homonymy distinction involves diachronic considerations. I would like to suggest, however, that the dependence on the facts about the history of the language should be aligned with the observation that speakers of a language are more or less unaware of the etymology of words, which also means that diachronically motivated polysemy-homonymy decisions lose their psycholinguistic relevance. On the other hand, when the history of the language is rejected as a clue, distinguishing polysemy and homonymy may turn out to be more than challenging.

Lyons (1977) argues that we can exclude either polysemy or homonymy from our descriptions. If homonymy were excluded, the lexicon would have to be fairly underspecified for meaning to accommodate "remote" uses of any given form. In such a system, no lexical relations could easily be expressed in practice. If polysemy were excluded, different meanings would be assigned to different lexical entries. Can you, however, give a full description of all the possible uses of a form? Can you enumerate all senses of a lexical entry?

While the above questions are open-ended, enumeration of senses *in printed dictionaries* is a proven tradition and may well be a necessity in practice. Discrete senses have always been difficult to find, and lexicographers have long been aware of this problem. Since the dictionary-writing tradition requires linguists to come up with entries enumerating different uses of the keyword, they have to decide whether a tiny difference in usage pattern constitutes a different sense or not. In the compilation of a dictionary entry, "*lumping* is considering two slightly different patterns of usage as a single meaning", and "*splitting* is … dividing or separating them into different meanings" (Kilgarriff 1997:9). Whether lexicographers lump or split senses is a matter of tradition, editorial policy and subjective decisions.

3.3 Sense delineation in Cruse (1986)

The smallest unit in Cruse's (1986) lexicon is a *lexical form*, which is an inflectional paradigm (this view is compatible with what the strong lexicalist tradition suggests, see for instance Halle 1973). At the next level, a *lexical unit* combines a lexical form with a single

[36] To further complicate matters, this Scandinavian form is related to the German source of the Italian "banca", which is the source of English *bank*$_1$ (Lyons 1995:28).

sense (Cruse 1986:52). He argues that a lexical form can be associated with an unlimited number of senses, but these senses are not equal in status: *established senses* are more frequent, and "are presumably represented differently in the mind's lexicon" (p. 68). Cruse adds that the more central, established senses should be defined in a dictionary (p. 79). He emphasizes that even established senses exhibit different *grades of centrality*, furthermore the *primary lexical units* "become operative in minimal, or neutral, contexts" (p. 79). The *role of the context* is different for pre-established and not-established senses: with pre-established senses, the context is a filter that facilitates the selection process; with not-established senses, the context triggers the right set of procedures generating the sense in question (pp. 68-69).

At the topmost level, the lexicon contains *lexemes*, which are families of lexical units (p. 49). This approach makes it possible to create structure based on relatedness of sense.

Sense-spectra are "amoeba-like" features of the lexicon, with the potential of growing along many dimensions. It is not possible to find a common superordinate for them (pp. 71-72). The author admits, however, that it is difficult to work with sense-spectra, and they do not "enter into any recognised lexical relations" (p. 73) as a whole.

Cruse is one of the most prominent advocates of context-based sense manipulation processes. The following is from Cruse (1986):

> There are fundamental ways in which the effective semantic contribution of a word form may vary under the influence of different contexts. First, a single sense can be modified in an unlimited number of ways by different contexts, each context emphasising certain semantic traits, and obscuring or suppressing others… This effect of a context on an included lexical unit will be termed modulation; the variation within a sense caused by modulation is largely continuous and fluid in nature. The second manner of semantic variation concerns the activation by different contexts of different senses associated with ambiguous word forms.
> (Cruse 1986:52)

In Cruse (1986, 2000), the context is a stimulus that can "generate" senses. Other linguists have also postulated sense-generating processes. Most notably, Pustejovsky's Generative Lexicon (Pustejovsky 1995) is able to account for given cases of systematic polysemy[37]: it can generate forms that are systematically related in meaning. Let me finally

[37] Systematic polysemy occurs when "the same relationship holds between the senses for two or more polysemous words" (Kilgarriff and Gazdar 1995:2, also see section 3.4 of the present work). Pethő (2001)

point out that Cruse (1986) refrains from connecting modulation, sense establishment and other processes to the concepts of homonymy and polysemy (which is made explicit by Cruse 1986:80). As the next section shows, however, Cruse (2000) finds homonymy and polysemy worth incorporating into a description of lexical semantics.

3.4 Lexical Semantics in Cruse (2000)

Cruse (2000) argues that *ambiguous words* have multiple senses that exhibit the phenomenon he calls *antagonism* (see below), but even those words that do not show antagonism may have multiple *discrete readings* that can be detected via various tests.

Consider the following example (Cruse 2000:108):

(1) *We finally reached the bank.*

Cruse (2000) points out that we cannot focus our attention on both the “financial institution” and the “riverbank” senses of *bank* at the same time, only one of them can be active at a time. This phenomenon is called *antagonism* (p. 108). He adds that “the speaker will have one reading in mind, and the hearer will be expected to recover that reading on the basis of contextual clues: the choice cannot normally be left open” (ibid.).

Cruse (2000) suggests the following procedures for the examination of the discreteness of readings.

The *identity test* (which is based on the *identity constraint*) is applicable to sentences that evoke the meaning of a word more than once through anaphoric back-references (p. 106). The identity constraint makes it difficult for such a back-reference to assume a reading that is different from the preceding readings of the given word. Cruse offers the following example (ibid.):

(2) *Mary is wearing a light coat; so is Jane.*

He points out that this sentence cannot normally be used to express a situation in which Mary’s coat is lightweight and Jane’s is light colored, or vice versa.

Independent truth conditions for sentences with multiple readings indicate discreteness, too. “A good test of this is whether a context can be imagined in which a Yes/No question containing the relevant word can be answered truthfully with both *Yes* and *No*” (Cruse 2000:107). Consider the example in (3):

(3) Are you wearing a light coat?

points out that this is called *regular polysemy* in Apresjan (1973), where a parallel is drawn between the mechanisms of regular polysemy and that of derivational morphology.

A person wearing a light-colored, heavyweight coat can truthfully answer *yes* and/or *no* (p. 107), which is made possible by the independent truth conditions associated with the discrete readings of the word *light.*

The presence of multiple readings is also indicated by the existence of *independent sense relations* for the word. Consider, for instance, the antonyms for the two readings of the adjective *light* mentioned above. One of the readings has the opposite *dark*, while the other reading can be contrasted with *heavy* (p. 107).

Finally, the discreteness of various readings is also shown by the phenomenon that Cruse calls *autonomy*: when a reading becomes anomalous in a certain context, *autonomous readings will still remain available*. His example is the following:

(4) *I prefer dogs to bitches.*

In this sentence, the "canine species" reading of *dog* is unavailable, but a more specific meaning, "male of canine species" is acceptable (p. 107).

Discrete readings detected by the above tests do not necessarily cause ambiguity, but antagonistic readings are ambiguous and they show the highest degree of discreteness (p. 108). In Cruse (2000), antagonistic readings constitute distinct *senses*.

In Cruse's classification, readings that do not form distinct senses may fall into the categories of facets, perspectives, subsenses or sense spectra.

Facets are discrete readings of a 'gestalt'. When a word has multiple facets, their combined reading is the default reading of that word (p. 116). Cruse's example is the "text" and "tome" readings of *book*. Tests to detect facets include the discreteness tests (see above). An interesting new test checks for the presence of *independent metaphorical extension*: for example, the metaphor in *a book of matches* is only related to the "tome" facet (p. 115).

Perspectives are not antagonistic, not autonomous readings that show less discreteness than facets do. The author names the following four major perspectives:

- Seeing something as a whole consisting of parts (e.g. the perspective of a veterinarian on a *horse* is likely to involve the body parts of the horse)
- Seeing something as a kind, in contrast with other kinds (e.g. a zoologist will view a horse as strikingly different from deer and zebras)
- Seeing something as having a certain function (e.g. a jockey's or a tribesman's perspective on *horse*)
- Seeing something from the point of view of its origins (e.g. a builder's view on a *house* [as associated with, for instance, groundwork and bedding])

(Cruse 2000:118)

Subsenses display “a lower level of both discreteness and antagonism” than “full” senses do (p. 119). Cruse points out that the word *knife* has subsenses corresponding to *penknife, table knife, pruning knife*, and one of these subsenses is the default reading, depending on the context, with an option to use the general sense when required by the context (ibid.).

The phenomenon of *sense spectrum* is exemplified by several readings of the word *mouth* in Cruse (2000). The author points out that some ‘distant’ readings of *mouth* cause zeugma when they are coordinated (e.g. *?The poisoned chocolate slipped into the Contessa’s mouth just as her yacht entered that of the river*, Cruse 2000:120). Other, less distant readings on the same sense spectrum may pass the coordination test without punning: *The mouth of the cave resembles that of a bottle* (ibid.). Cruse emphasizes that the readings that are positioned along a sense spectrum are domain-specific local senses, which are similar to subsenses (but, unlike the readings of a sense spectrum, subsenses are not points along a *semantic continuum*, p. 119).

Cruse (2000:120-123) lists three ways in which the context can influence the meaning of a lexical item. The context may facilitate a *selection* process: existing readings or established senses are selectively activated and suppressed. When the established senses do not fit into the context, the listener is supposed to look for a matching meaning extension, possibly metaphorical or metonymical, “because of a tacit assumption that speakers are usually trying to convey an intelligible message” (p. 120). The meaning that is found is *coerced* by the context (ibid.). Finally, meanings can be *modulated* by the context in various other ways. Consider the following sentences (taken from Cruse 2000:121):

(5) Our maths ***teacher*** is on maternity leave.

(6) The ***coffee*** burnt my tongue.

Cruse argues that both sentences contain hyponymic enrichment, which adds meaning to the semantic content of the lexical item in bold. In (5), the teacher’s gender is added; in (6), the high temperature of the coffee is implied.

Cruse (2000) points out that the *relatedness of senses and readings* is continuous in nature, and this continuum includes “clear cases” of homonymy (he refers to the *bank* example, p. 109), as well as various forms of polysemy[38].

[38] Cruse explicitly denies the existence of a sharp distinction between polysemy and homonymy (cf. Cruse 2000:109).

A major subclass of polysemy involves readings that are specializations/generalizations of one another: this is called *linear polysemy* (p. 110-111). Linear polysemy has the following cases:

- Autohyponymy: a default general sense is accompanied by a more specific, "contextually restricted" sense. Cruse's example is the word *dog* in the meanings "member of canine race" and "male member of canine race" (p. 110).
- Automeronymy: it is similar to autohyponymy, but "the more specific reading denotes subpart rather than subtype" (p. 111).
- Autosuperordination: a default specific reading is accompanied by a contextually restricted, more general reading (e.g. the word *man* referring to "the human race" in certain contexts, p. 111).
- Autoholonymy: the default reading denotes a part of an entity, but this entity can also be expressed – under certain circumstances – by the same word.

Non-linear polysemy includes the following cases:

- Metaphor: "figurative usage based on resemblance" e.g. *You've put me in an awkward **position*** (p. 112, also see section 3.5).
- Metonymy: "figurative use based on association", e.g. *Jane married a large **bank account*** (p. 112).
- Miscellaneous, e.g. the "non-calendric" reading of *month* in a situation when it expresses duration rather than a month starting on the first day of the given month and ending on the last day of the same month (p. 112-113).

When the relationship between readings is generalizable across a range of lexical items, which should also be "partly predictable on semantic grounds", we talk about *systematic polysemy* (p. 113). Cruse argues that metaphorical meaning modification is the least systematic, metonymical processes are the most systematic, and linear polysemy can be systematic, too (ibid.).

3.5 The prototype approach and the basis for Polysemy in Cognitive Linguistics

To illustrate the difference between the classical theory of categorization and an alternative, non-Aristotelian approach, let us examine two possible ways of accounting for color terms. The *structuralist* approach was laid out by Bloomfield:

> Physicists view the color-spectrum as a continuous scale of light-waves of different lengths, ranging from 40 to 72 hundred-thousandths of a millimetre, but languages mark

> off different parts of this scale quite arbitrarily and without precise limits, in the meanings of such color-names as violet, blue, green, yellow, orange, red, and the color-names of different languages do not embrace the same gradations.
> (Bloomfield 1933:140)

According to Bloomfield, color categorization is *arbitrary*, and it is tempting to draw the conclusion that similar phenomena, including temperature, speed and length, are also arbitrary in nature.

Berlin and Kay (1969) proved that color categories have a structure that is universal across languages. They examined almost one hundred languages, and concluded that "although different languages encode in their vocabularies different numbers of basic color categories, a total universal inventory of exactly eleven basic color categories exists from which the eleven or fewer basic color terms of any language are always drawn" (Berlin and Kay 1969:2). They call the colors that correspond to these basic color terms *focal colors* (in English, they are *black*, *white*, *red*, *yellow*, *green*, *blue*, *brown*, *grey*, *orange*, *purple* and *pink*). On the basis of their cross-linguistic vocabulary study, the authors also argue that some of these colors are more basic than others.

The research conducted by Berlin and Kay pinpoints that human conceptualization, which is based on human perception, cannot be ignored in linguistic description. Taylor (1995:16-20) points out that we can approach the relationship between language and the conceptual system in two ways. The first option is modular in the sense that it is based on Chomsky's idea of the *autonomous* language faculty, which is "viewed as a computational device …, which determines a person's grammatical competence" (p. 16). Accounting for the data on color categorization involves the "blurring of the distinction between the purely linguistic and non-linguistic components of language knowledge" (p. 18). Taylor makes a reference to Chomsky (1986:18) who states that blurring this boundary does not cause theoretical or practical conflicts, but this option is not discussed in more detail. Throughout his book, Taylor argues for the existence of our second option: "I shall take the reverse position, i.e. that no distinction needs to be drawn between linguistic and non-linguistic knowledge. The facts of colour categorization as manifested in the meanings of colour terms are at once both facts about human cognition *and* about human language" (Taylor 1995:18, emphasis original)[39].

[39] Taylor explains that on this view (as compared to the first option relying on an autonomous language faculty), "a clean division between linguistic and non-linguistic faculties, between linguistic facts and non-linguistic facts … may ultimately prove to be both unrealistic and misleading" (Taylor 1995:18).

Berlin and Kay's research is interesting from a different point of view, too: it affects the way we think about how categories can be defined. Taylor (1995:23) characterizes traditional (Aristotelian) categorization in the following way:

a) categories are defined in terms of a conjunction of necessary and sufficient features (e.g. "man is a two-footed animal" = if X is a man then X is two-footed AND X is an animal)
b) features are binary (either true or false)
c) categories have clear boundaries (as Taylor points it out, it follows from *a* and *b*,)
d) all members of a category have equal status (again, it follows from *a* and *b*)

Taylor highlights that some linguists, mostly phonologists and semanticists, introduced further assumptions about the features used in the Aristotelian model of categories (Taylor 1995:25-37):

a) features are primitive, i.e. they cannot be decomposed into more primitive features, which also means that one cannot build a hierarchy of features,
b) features are universal: from a practical point of view, linguists have to come up with a universal feature inventory, and every human language will pick out its own set of features.

As Smith (1991:296) underlines, the *prototype approach* does not necessarily reject featural representations, it rejects only the claim that categories have *necessary* features that collectively are sufficient to define them. Instead, we have exemplars in mind for our categories that help us "decide whether something else is a member of the category by comparing it with that prototype" (ibid.). I would like to suggest the term *prototypical exemplification*: having rejected the use of defining features in this framework, we need to come up with "defining" examples instead that are able to instantiate prototypical categories. Taylor points out that the examples must be selected carefully so that we

a) maximize the number of attributes shared by the members of the category; and
b) minimize the number of attributes shared with members of other categories

(Taylor 1995:51).

I would like to mention two (related) phenomena that accompany the prototype effect. Firstly, a classical category facilitates two degrees of membership only (member versus non-member), while prototype categories exhibit a non-binary membership function (Taylor 1995:54). It corresponds to the phenomenon called <u>degree of membership</u>, which I connect to prototypical exemplification (as described above) in the following way: the example(s) instantiating a prototypical category has or have the highest degree of membership in that particular category.

Secondly, prototype categories have no clear boundaries. Elements with a low degree of membership are on the periphery of a category, and of course, certain elements may belong to multiple prototype categories at the same time with different degrees of membership. The bottom line is that prototype categories are fuzzy in nature. Degree of membership and fuzziness are rather neglected phenomena, probably because they are difficult to handle and systematize.

The following overview of the characteristics of prototypical categories is due to Geeraerts:

a) Prototypical categories cannot be defined by means of a single set of criterial (necessary and sufficient) attributes.
b) Prototypical categories exhibit a family resemblance structure, or more generally speaking, their semantic structure takes the form of a set of clustered and overlapping meanings (which may be related by similarity or by other associative links, such as metonymy). Because this clustered set is often built up round a central meaning, the term ‘radial set’ is often used for this kind of polysemic structure.
c) Prototypical categories exhibit degrees of category membership; not every member is equally representative for a category.
d) Prototypical categories are blurred at the edges.

(Geeraerts 1994:3385)

Geeraterts has also come up with a matrix that sheds further light on the nature of the above properties:

	Nonequality (differences in structural weight)	*Nonrigidity* (flexibility and vagueness)
Extensional characterization	degree of representativity	absence of clear boundaries
Intensional characterization	clustering of overlapping senses	absence of necessary-and-sufficient definition

(Geeraerts 1994:3386)

Geeraerts points out, however, that researchers are still “faced with the task of further clarifying the relationship between the various characteristics of prototypicality” (Geeraerts 1994:3386).

A possible reason for the lack of fully developed theories of prototypicality can be found in Smith (1991):

> … the concept of a prototype is itself prototypical rather than criterial. It has fuzzy boundaries and is characterized by family resemblances and by typical rather than necessary and sufficient attributes. Therefore, it is not at all clear how one would set out to formalize prototypical representations…
>
> (Smith 1991:296)

Although we must take these warnings seriously, researchers have found prototypes extremely useful, efficient and convenient at many levels of linguistic description.

Pethő (2001)[40] points out three sources of polysemy in Lakoff's (1987) model[41]:

a) Generality of human cognitive categories: the prototype effect makes the borderline between "single meaning" and polysemy fuzzy (Pethő 2001:188).

b) Family resemblance: "For example, a prototypical category *A* based on family resemblance has a set of members *a, b, c d,* of which *a* is similar to *b, b* is similar to *a* and *c,* … but *a* is not at all similar to *d* and there is no single member of *A* that is similar to all other members. In such a scenario, one will likely be tempted to judge the word w_A that corresponds to the category *A* to be polysemous, but one will have difficulty deciding exactly what distinct meanings to attribute to it" (p. 189).

c) General cognitive operations including conceptual metaphor and metonymy, which are major sources of polysemy (Pethő 2001:189-190, cf. section 3.4 on non-linear polysemy).

In what follows, I would like to elaborate this third option in some more detail. Kövecses et al.'s (1996) definition for the term *conceptual metaphor* is the following:

> Conceptual metaphors bring into correspondence two domains of knowledge. One is typically a well-delineated, familiar physical domain and the other a less well-delineated, less familiar abstract domain.
>
> (Kövecses et al. 1996:18)[42]

[40] He gives a review of the literature published after 1987 on polysemy. 1987 is the publication date of Paul Deane's unpublished dissertation on Polysemy, which is used as a point of departure in Pethő's paper.

[41] The role of metaphor in linguistics has been addressed before Lakoff, too. For example, Botha (1968) states that creative metaphors are beyond the scope of linguistics proper, since they are not rule-governed but rule-changing in nature (Botha 1968:200). Established metaphors, on the other hand, should appear in the lexicon together with the conventional senses (Botha 1968:201).

[42] Also see Kövecses (2002) for an explanation of key ideas on metaphor.

Pethő's definition echoes the main points made by Kövecses et al. (1996): "Conceptual metaphor is a general cognitive strategy which involves the conceptualization of abstract or less familiar phenomena by recourse to something more concrete or more familiar" (Pethő 2001:189). Pethő uses the conceptual metaphor AN OBJECT IS A HUMAN BODY to exemplify his point.

> If this metaphor is applied to an object that has parts protruding from it so that they resemble appendages, these can be referred to as *arms* and *legs*. Thus one can talk about an *arm of an ocean* or *legs of a table*. These are, obviously, lexicalized uses of the words *arm* and *leg* that are motivated by this conceptual metaphor.
> (Pethő 2001:190)

Pethő argues that the above account of polysemy "is non-explanatory and motivational", because it cannot predict the meaning that a specific word develops, but specifies the motivation behind the coinage of a new synonym (p. 190). Pethő reasons that *metonymy* is a possible source of *systematic polysemy* (cf. sections 3.3 and 3.4), and metaphorical extension leads to *non-systematic polysemy* ("the relationship is particular to a single word", ibid.)[43]. In the case of non-systematic polysemy, you cannot give rules that could account for the different meanings; therefore, you must enter them in the lexicon (Pethő 2001:178-179). His example is a set of meanings related to the word *glass.*

Langacker (1987) points out that there are *domains* in the conceptual system, and any piece of language should be described along the lines of the domains that are involved. Basic domains include the concepts of TIME and SPACE (cf. Cruse 2000:141). Taylor (1995) argues that domains are useful in the detection and description of polysemy: "if different uses of a lexical item require, for their explication, reference to two [or more, alternative][44] domains, or two different sets of domains", it is likely that we are dealing with polysemy (Taylor 1995:100).

3.6 Insights from lexicon-makers

3.6.1 Kilgarriff's notes on Polysemy

Kilgarriff (1992) analyzes polysemy in terms of four "neighbours" (Kilgarriff 1992:71-81).

[43] Pethő's position is very similar to Cruse's opinion on systematic polysemy (cf. Cruse 2000:113-114).

[44] This modification is based on Taylor (1995), too.

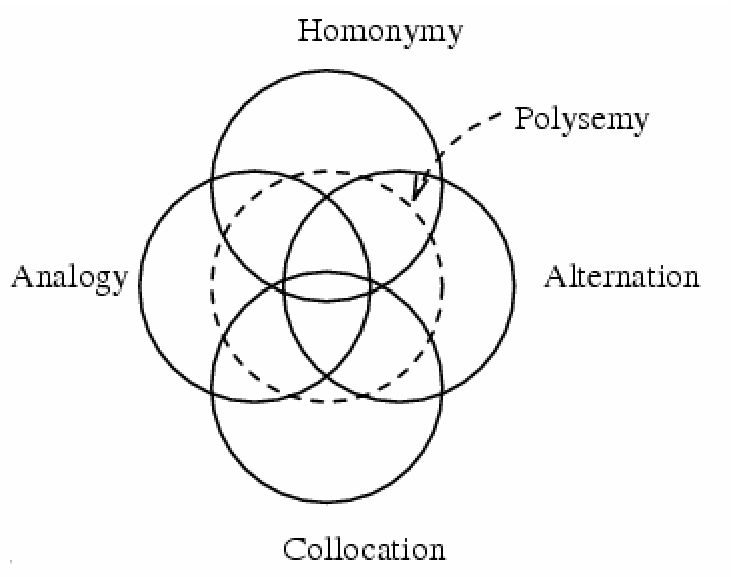


Figure 3-1 “Polysemy and its neighbours” (reproduced from Kilgarriff 1992:72)

Kilgarriff makes the following comments on the natural language phenomena that appear in the above diagram (the included explanations are due to Kilgarriff, too):

1. Homonymy. … Usage-types are expressed through being listed. They are represented in the lexicon as distinct one-word entries.
2. Alternation. A system of rules indicates how a non-basic usage-type may be inferred from a basic one.
3. Collocations. For a usage-type which co-occurs only with a limited range of words, all the collocations are listed. Thus the adjectival sense of *frontal* which means `direct and obvious' (COBUILD) seems to occur only with *attack* and *assault*. The two collocations can be stored in the lexicon. …
4. Analogy. …

   Two words, *x* and *y*, have a similar meaning in their primary senses, x_1 and y_1 , and *x* has a familiar secondary sense, x_2 . Then if *y* is used in the sort of context where x_2 is often used, *y* will be interpreted as the novel y_2 , relating to y_1 in the same way that x_2 relates to x_1 .

(Kilgarriff 1992:72-73)

Kilgarriff adds that with Alternations and Analogy, the “usage-type” is only available as an inference, and in the latter case, the “inference cannot be made on the basis of facts and rules in the lexicon alone” (p. 72).

Kilgarriff's model offers four dimensions along which polysemous words are potentially formed and interpreted. The result is at the "crossroads" between the four neighbors. As far as the polysemy vs. homonymy distinction is concerned, Verspoor makes the following remark about Kilgarriff's model:

> there cannot be clear-cut tests for identifying polysemy due to its multi-faceted nature. Homonymy is not orthogonal to polysemy, but rather an endpoint of one of the dimensions along which polysemy can be described (fully predictable sense variation – unpredictable sense variation).
> (Verspoor 1997:218).

One may wonder where non-systematic polysemy should be placed in this model. Kilgarriff and Gazdar (1995) make a comment that clarifies their commitment: "it is unlikely that a distinction between 'irregular polysemy'[45] and 'homonymy' would serve any purpose in a synchronic description of the lexicon" (Kilgarriff and Gazdar 1995:2).

3.6.2 The limitations of Sense Enumerative Lexicons

Pustejovsky (1995) points out that conventional lexicon design, which is based on sense enumeration, is inadequate in many respects. First, it cannot account for the *Creative Use of Words*, the process of how "words assume new senses in novel contexts" (Pustejovsky 1995:39). Examples for creative use include various readings of the adjective *good*. Compare, for instance (p. 43):

(7) Mary finally bought a good umbrella.

(8) After two weeks on the road, John was looking for a good meal.

The definition of *good* in (7) is "to function well", while it means "tasty" in (8). Pustejovsky points out that sense enumeration would involve the creation of separate entries for both (and many more) uses. "As an alternative, one might simply keep the meaning of *good* vague enough to cover all the cases mentioned above. Then, world knowledge or pragmatic effects could further specify the manner in which something is good…" (p. 43).

A sense enumerative lexicon also fails to accommodate the phenomenon referred to as the *Permeability of Word Senses*, which is described in the following way: "Word senses

[45] "Irregular polysemy" is used as a synonym for what Pethő calls "non-systematic polysemy" (but the term is used somewhat ironically: in this paragraph, the authors argue against the name "regular polysemy" used in Apresjan 1973)

are not atomic definitions but overlap and make reference to other senses of the word" (p. 39). Compare the following sentences:

(9) Mary cooked a meal.

(10) Mary cooked the carrots.

Pustejovsky points out that *cook* in (9) implies both *creating a meal* and *change-of-state*, but only this latter reading is implied in sentence (10) (p. 47). Pustejovsky argues that overlaps of core and peripheral meaning components cannot be described in a "flat, linear enumeration-based organization of dictionary entries" (p. 48).

Finally, the *Expression of Multiple Syntactic Forms* is also hindered in enumerative models (pp. 50-54), although this statement is meant to refer to "maximally enumerative" implementations in which different syntactic interpretations, such as factive vs. non-factive use, are encoded as separate lexical entries.

3.6.3 Verspoor's notes on NLP lexicon design

Traditional dictionaries are *for human use*, and the needs of the human dictionary user do not necessarily coincide with the needs of an NLP system. What do we expect from an NLP lexicon?[46] According to Verspoor (1997), it depends on the task for which the system is designed. She identifies the following NLP tasks and their "lexical needs" (pp. 207-214):

– POS tagging (she calls it "shallow parsing"): part of speech taggers achieve remarkable accuracy relying on statistical co-occurrence information (derived from a corpus), but enhancing their accuracy requires the introduction of "non-probabilistic rules which define highly specific contexts in which a certain part of speech is more likely…" (p. 207). The implementation of this supporting rule-set relies on a lexicon that contains, for instance, idioms, naming expressions and compounds (p. 208).

– Syntactic parsing ("deep parsing"): the needs of *stochastic* parsers are similar to those of POS taggers, while "theoretically-based" (*rule-based*) methods may require subcategorization information (to identify phrases, for instance) and features such as case, gender and tense.

– Information Retrieval (IR): the "most linguistically naïve" branch of IR only needs a wordlist, which is used to index documents; this index can then be searched for the words of the query string to identify relevant sources. To eliminate irrelevant responses, word sense disambiguation (WSD) must be carried out. Let me point out that WSD has

[46] Also cf. Prószéky (1997).

its own pitfalls (cf. Ide and Véronis 1998 reporting varying levels of WSD accuracy including cases of worse-than-chance performance and also describing experiments involving manual WSD, which turns out to be inaccurate and problematic, too), but Information Retrieval seems well-suited for non-traditional approaches to storing lexical information that may effectively eliminate WSD (cf. sections 4.3 and 4.4 of this thesis).

- Machine Translation (MT): Verspoor argues that WSD causes less problem for MT than for Information Retrieval, since “translation equivalents which are associated with the same set of senses in both source and target language, i.e. which are ambiguous in the same ways in both languages, do not need to be disambiguated for the purposes of translation” (p. 211). I doubt, however, that these ‘cross-linguistically unambiguous’ lexical coincidences help a lot in the translation process: this phenomenon is most likely to affect entries with a low polysemy figure, which are easier to disambiguate anyway, while our most common words exhibit a rather high polysemy figure (cf. Mihalcea and Moldovan 2001:454). In general, Verspoor seems to present MT as an NLP process involving the replacement of source language words by target language “equivalents” (p. 211), and isolates MT from the more ‘demanding’ tasks of language understanding and language generation (see below). It is probably a “naïve” approach to an NLP field that has attracted so much attention and funding and has produced a lot of disappointment.
- Natural Language Understanding (NLU; for text summarization, information extraction, natural language interfacing): Verspoor argues for the implementation of semantic relations (cf. section 4.1) in the lexicon of these NLP systems: “[l]exical structure needs in particular to reflect semantic relations such as hyperonymy/hyponymy (i.e. the hierarchical relations between words[47]) and synonymy/antonymy in order to capture generalizations about the ways in which similar words can be used” (pp. 211-212).
- Natural Language Generation (NLG): Verspoor points out that NLG involves *text planning* and *linguistic realization*. “The solution for linguistic realization will need to rely heavily on a rich lexicon of syntactic and semantic information, governed by constraints on the combination of words into sentences and discourse coherence factors” (p. 213). She also argues that in an NLG system, polysemy should be handled appropriately, “since the range of meanings a word can take on in specific syntactic

[47] Hyperonymy (also known as hypernymy) and hyponymy may be used to create *is-a* hierarchies of concepts (e.g. *vertebrate–mammal–horse*), cf. Pedersen, Patwardhan and Michelizzi (2004).

environments is directly relevant to the problem of forming a grammatically correct and easily understandable sentence using that word" (ibid.).

Verspoor (1997) argues that a rich lexicon with fine sense granularity should be used for NLU and especially for NLG, which leads her to the question of *polysemy*. She points out that traditional dictionary design hinges on "the discreteness of the meaning expressed in a usage of a word – where there is ambiguity, only one sense of a word can be active at any one time" (Verspoor 1997:219)[48], and these distinct senses correspond to distinct entries in the lexicon[49] (ibid.). She goes on to explain that this view is incompatible with the idea of underspecified representations and emphasizes a potential clash with Cruse's (1986) sense-modulation theory. Using Cruse's (2000) terminology, Verspoor (1997) seems to argue that traditional lexicon design concentrates on *antagonistic readings* (cf. section 3.4) while non-antagonistic *discrete readings* and meaning variations triggered by the context are suppressed, and traditional lexicon design "demands the impossible task of enumerating in advance all the senses which might be associated with a lexical form" (p. 219). Verspoor points out that the use of a finite set of pre-fabricated choices "from which the NLP system can choose the most appropriate" has remained a practice nevertheless (Verspoor 1997:220).

The next chapter of this thesis introduces real-world lexical databases (WordNet and FrameNet) that have already had a major impact on the design of the lexical component of NLP projects (see Mihalcea 2004 for a huge list of WordNet-related works, for instance). These databases are based on the sense-enumeration idea. MindNet, a third solution, offers an exciting novel approach to relational lexical databases, and it aims to avoid some of the problems related to sense-enumeration.

[48] Verspoor also argues that traditional lexicon design is an example of the "maximized homonymy" approach.

[49] The term *lexicon* is used because Verspoor argues that this approach appears both in lexicography and linguistics (Verspoor 1997:219).

4. WORDNET, FRAMENET AND REPRESENTING LEXICAL KNOWLEDGE IN A LEXICAL DATABASE

Although WordNet and FrameNet are two lexicon-building enterprises that offer very different content, they are similar in many respects. Firstly, the possible uses and target applications are similar: lexicographical projects, NLP systems and language teaching applications may profit a lot from them. Secondly, both of them can be exploited as dictionaries as well as thesauri. Finally, they are likely to influence the lexical component of future NLP applications, since they offer a formidable amount of reliable lexical information which is freely available for the research community. They introduce semantic relations (WordNet uses a set of sense-relations and FrameNet stores frame-relations), which are exciting non-traditional tools in the practice of representing meaning in a dictionary or an NLP lexicon, although these relations do not emerge as the right tools for representing polysemy (or the polysemy-homonymy distinction). A less influential, commercial project (MindNet) is also discussed in this chapter, which foreshadows some new, exciting possibilities, not unlike Véronis and Ide's (1990) connectionist model, which is introduced in the final section.

4.1 WordNet

WordNet[50] (WN) is a large, publicly available electronic dictionary, thesaurus and semantic network of English, which has the potential of serving Natural Language Processing systems well due to is size and sophistication. The compilers' original idea was to "identify the most important lexical nodes by character strings and to explore the patterns of semantic relations among them" (Miller 1998:xvii). A plethora of information has become input including the Brown Corpus, various thesauruses and wordlists, and the database has been growing steadily. More than 100,000 word forms are listed in the database in a structure that facilitates lookup on the basis of semantic similarities. In addition to *synonymy*, which is the main organizing *lexical* relation in the database, a host of *semantic* relations[51] are incorporated. Let us take one of the nominal senses of "book" as an example: it is co-listed with the synonym *volume*, and it is possible to locate hypernyms (e.g.

[50] For a comprehensive introduction to WordNet, see Fellbaum (1998). The official homepage of the project is at http://www.cogsci.princeton.edu/~wn/ as of December 31, 2004.

[51] In WordNet, relations that link *words* are called 'lexical relations', while the relations that link *synonym sets* are referred to as 'semantic relations' (cf. Fellbaum 1998:9).

product), hyponyms (e.g. *album*, *journal*), holonyms ("book is a part of …", not available for this sense), meronyms (e.g. *binding, cover*) and coordinate terms (e.g. *inspiration* and *deliverable*) for the synonym set containing this sense. The system of semantic relations stored for verbs are equally elaborate. Adjectives and adverbs are also accounted for.

WordNet has been used for many purposes in the NLP literature, which is well illustrated by Fellbaum (1998): in addition to some papers discussing the design and contents of WordNet, this volume includes works on various WN-related research topics (e.g. informational retrieval and semantic concordances). Researchers working on Word Sense Disambiguation (WSD), which is a key problem area in NLP today, have been especially enthusiastic about WN. Papers discussing WordNet-related WSD include Agirre et al. (2000), Banerjee and Pedersen (2003), Basili, DellaRocca and Pazienza (1997), Basu et al. (2001), Dorr and Jones (1996), Fellbaum, Grabowski and Landes (1995), Fellbaum et al. (2001), Karov and Edelman (1996), Kwong (2001), Li, Szpakowicz and Matwin (1995), Lin (1997), Mihalcea and Moldovan (1998, 1999, 2000, 2001b), Narayanan and Bhattacharyya (2002), Nastase and Szpakowicz (2001), Pedersen and Bruce (1997), Voorhees (1993) and Wiebe, O'Hara and Bruce (1998).

Word Sense Disambiguation is directly connected to the natural language phenomena (polysemy and homonymy) discussed in chapter 3. The highly polysemous nature of human word stocks has always posed a great problem for computational systems, and no universal solution has been found: most of the research on WSD is heavily task-dependent (for an excellent general introduction and review, see Ide and Véronis 1998, and also see Kilgarriff 1992). In WordNet, a particular word is very often co-listed in a number of synonym groups. The unusually high number of co-listed entries makes disambiguation even more difficult in WordNet-based systems, so attempts have been made to compile more "compact" versions WN, as discussed in the next section.

4.1.1 Sense distinctions in WordNet

Sense is represented by lexical and semantic relations in WordNet. Synonymy is the principal device: the compilers captured the senses of words (and multi-word strings) by assigning them to synonym sets (synsets). In 1989, the compilers started to add explanatory glosses (illustrative examples), too, to help users and themselves to keep "all the different word senses distinct" (Miller 1998:xx). Many synsets are exemplified by full sentences. Glosses and example sentences serve as instances of *context* in which the given sense can be interpreted. As we have seen, context has a crucial role in sense specification in Cruse (1986) and other works.

In WordNet, even tiny sense variations are kept distinct, and the database is probably as fine-grained as possible. Seagull (2000) argues that fine WN sense distinctions often reflect *regular polysemy* created by processes that affect multiple words (Seagull 2000:1). Mihalcea and Moldovan (2001) point out that it is not uncommon that WN "word senses are so close together that a distinction is hard to be made even for humans" (Mihalcea and Moldovan 2001:454). They have also computed *polysemy values* for WN version 1.6. According to them, the average polysemy figure (senses/words) is 1.39 (the figure for verbs is the highest at 2.13). They emphasize, however, that our most frequent words are highly polysemous. This is the most probable reason why the SemCor corpus, which is tagged using WordNet senses, exhibits an average polysemy of 6.55 (Mihalcea and Moldovan 2001:454). I would like to point out, however, that the use of the term "polysemy figure" may not be appropriate, since WordNet does not make a distinction between polysemous senses and instances of homonymy. WN is an example of the *maximized homonymy* approach (cf. section 3.2).

Palmer (1998) argues that computational lexicons in general, and WordNet in particular, should only enumerate senses that can be identified later by differences in argument structure and/or selectional restrictions (Palmer 1998:7). She also argues that WN contains distinctions that are based on *world knowledge*. Her analysis of the senses of the verb *lose* is an eye opener. Sense #1 has the gloss "is fail to keep or to maintain; cease to have, either physically or in an abstract sense" (the example is "She lost her purse when she left it unattended on her seat"). Sense #5 is glossed as "miss from one's possessions; lose sight of" (exemplified by the sentence "I've lost my glasses again!"). There is nothing in the database indicating that sense #1 and sense #5 are related, but the difference is difficult to detect even for the human reader. Palmer gives the following explanation for this: "these two WordNet senses are not distinguished because of anything to do with the verb arguments (an animate agent and a solid object possessed by the agent in both cases), but rather are distinguished by possible future events – namely the likelihood of the object being found" (Palmer 1998:7).

There have been many attempts at compacting ("lumping") senses, making WN coarser. *Automatic processes* use heuristics, which may have little theoretical or psychological relevance, but work well in an NLP environment. EzWordNet implements the following heuristics (Mihalcea and Moldovan 2001:455-457):

A. If S1 and S2 are two synsets containing at least two words, and if S1 and S2 contain the same words, then S1 and S2 can be collapsed into one single synset S12.

B. If S1 and S2 are two synsets with at least K words in common, then S1 and S2 can be collapsed into one single synset S12.

C. If S1 and S2 are two synsets with the same hypernym, and if S1 and S2 contain the same words, then S1 and S2 can be collapsed into one single synset S12.

D. If S1 and S2 are two synsets representing two senses of a given word, and if S1 and S2 have the same antonym, then S1 and S2 can be collapsed into one single synset S12.

E. If S1 and S2 are two synsets representing two senses of a given adverb, and if S1 and S2 have the same root adjective, then S1 and S2 can be collapsed into one single synset S12.

In addition to these rules, very rarely occurring synsets are simply dropped below a certain threshold. Depending on this threshold and the value of K (for rule B), Mihalcea and Moldovan (2001) lowered the average polysemy figure from 1.34 to 1.29 in their first experiment and to 1.24 in a second experiment. The polysemy value for the SemCor corpus decreased from 6.55 to 4.89 and 4.00, respectively. The use of the above heuristics introduces error, which was assessed using the SemCor corpus: 2.16% of the senses was missing in the first experiment, and 5.6% in the second.

I would like to point out that EzWordNet's heuristics do not make it possible to "lump" senses #1 and #5 of *lose*, which would help us avoid the problems described by (Palmer 1998) as discussed above. Since no synonyms are given for either sense, the first two methods are not applicable. The antonym of sense #1 is *keep*, while sense #5 has the antonym *find*, so in this case, the antonym rule does not allow us to compact the senses. We can clearly see, however, that WordNet antonyms are useful for the human user in delineating senses or identifying different readings. According to Cruse (2000), the presence of potential *independent sense relations* for a word indicates discrete readings (albeit not necessarily ambiguous senses, cf. section 3.4 of this thesis).

Chen and Chang (1998) describe another heuristic approach designed for automatic sense clustering that can work on a dictionary and a thesaurus (thereby connecting them while also producing a coarser sense division). The algorithm they propose involves POS tagging the dictionary definitions, the removal of function words, and computing the similarity between the remaining "skeleton" of the definition and all relevant thesaurus classes (Chen and Chang 1998:71). They implement the algorithm using the Longman Dictionary of Contemporary English (Proctor 1978) as the dictionary component and the

Longman Lexicon of Contemporary English (McArthur 1992) as a hierarchical thesaurus. They also consider WordNet as an option (for clustering, i.e. as a thesaurus), but they reject it since WN "synsets are too fine-grained from the WSD perspective" (Chen and Chang 1998:65).

Pedersen, Patwardhan and Michelizzi (2004) report on a set of tools that implement semantic similarity and relatedness measures for WordNet. On the one hand, their system offers six *similarity* measures, which work primarily with information contained in *is-a* hierarchies. According to the authors, version 2.0 of WordNet contains 554 verb hierarchies (accommodating 13500 concepts) and 9 noun hierarchies consisting of 80000 concepts. *Is-a* distances can only be measured within hierarchies, but they have introduced two hypothetical root nodes that subsume all noun concepts into a single noun hierarchy and all verb concepts into a unified verb hierarchy (Pedersen, Patwardhan and Michelizzi 2004:1). On the other hand, the authors also offer three *relatedness* measures, which work with additional WordNet relations, such as *has-part* (relating, for instance *wheel* and *car*), or *is-made-of* (connecting *snow* and *water*, for example). The similarity and relatedness measures exploited in this work have been developed and published by Leacock and Chodorow (1998), Jiang and Conrath (1997), Resnik (1995), Lin (1998), Hirst and St-Onge (1998), Wu and Palmer (1994), Banerjee and Pedersen (2002) and Patwardhan (2003).

Manual compaction of WordNet senses is exemplified by Seagull (2000). He has selected 194 words (verbs and nouns) from version 1.6 of WN, and carried out a well-planned, but extremely laborious manual procedure. His aim has been to come up with a minimal set of senses for the selected words using:

- semantic reevaluation (Seagull 2000:7): different aspects of the same event are sometimes listed in WN as distinct senses: these senses are collapsed together;
- decomposing compositional items (Seagull 2000:3);
- finding "inferable meanings" (Seagull 2000:9), inferable from Pragmatics and real-world knowledge; and finally,
- metonymy is eliminated.

When you apply Seagull's compaction database to WN 1.6, you get a result that is supposedly more suitable for WSD and other NLP tasks. The relations among senses that make compaction possible are not stored in the compaction database, however, which means that meaning overlaps are repressed rather than represented.

Seagull's compaction enterprise is instructive in many ways. We must remember, however, that "splitting" and "lumping" senses in dictionaries depend on editorial policy and tradition, and we can also argue that lumping senses using semantic reevaluation is an inherently subjective process.

4.1.2 Morphology in WordNet

4.1.2.1 Inflection and compounding

English inflection is incorporated into WordNet[52]. A preprocessor program called *Morphy* browses an exception list for each incoming WordNet query. This exception list contains entries for inflected forms of compounds, idioms, as well as irregular single words (e.g. *attorneys general*, *kept*, etc.). A look-up hit results in a pointer to the corresponding WordNet entry. When the form is not listed as an exception, "the rules of detachment" for each syntactic category are applied "in the following manner: whenever a matching suffix is found, a corresponding ending is added, if necessary, and WordNet is consulted to see if the resulting word is found in the database" (Tengi 1998:125). The electronic documentation of WordNet enumerates the following rules of detachment:

Part of Speech	*Suffix*	*Replaced by*
NOUN	"s"	""
NOUN	"ses"	"s"
NOUN	"xes"	"x"
NOUN	"zes"	"z"
NOUN	"ches"	"ch"
NOUN	"shes"	"sh"
NOUN	"men"	"man"
NOUN	"ies"	"y"
VERB	"s"	""
VERB	"ies"	"y"
VERB	"es"	"e"
VERB	"es"	""
VERB	"ed"	"e"
VERB	"ed"	""
VERB	"ing"	"e"
VERB	"ing"	""
ADJ	"er"	""
ADJ	"est"	""
ADJ	"er"	"e"
ADJ	"est"	"e"

[52] Unlike some other lexical information sources, WN features a query system, too, and this is where inflection is handled.

Since Morphy can check whether the resulting form is in the database, listing multiple rules for a single entry does not cause difficulties. The verb *abandoned* is translated into *abandone* first, which is incorrect and results in no database match. Then *abandon* is produced, which is in the database.

Phrasal verbs that are listed in the database (e.g. *ask for it*) are identified and handled properly by Morphy. The electronic documentation points out, however, that noun collocations that contain inflected forms, such as *line of products*, cannot always be identified by Morphy (e.g. the query string *lines of products* becomes *line of product*, but this is not listed in the database). The correct WordNet entry will be found, however, if WordNet gets the exact string (*line of products*) as input, without affixation, since Morphy will not be invoked by the system in that case.

I would like point out that the lookup system removes spaces, hyphens and underscore characters from the query string (when no match is found for the original string), which may come in handy when looking for compounds. When the database user does not know if the compound is written as a single word or two words or is hyphenated, he/she/it may use the hyphenated form or spell the compound as two different words, and WordNet will find it however it is spelled[53]. Note that the lookup string "black bird" matches "blackbird" in WordNet (which has its own semantic net stored in the database), which is an example how compounds looked up as two separate words may still cause confusion.

4.1.2.2 Derived Forms in WordNet

The aim of this section is to explore how WordNet 2.0 represents derived nouns. For this experiment, I compiled a list of nouns using a custom-made filtering program which took the WordNet noun index file as input. Then, using a concordancer[54], lists of words were produced that contained one of the following trailing character sequences: *-ee, -er, -ation, -ment, -al, -ness, -ian, -ity* or *-ism*. Note that these endings are not necessarily actual affixes: for instance, the *-ee* list contains the word *tree*, although it does not have the *-ee* affix. In cases of multiple derivation, only word-final suffixes were kept (e.g. *compartmentalization* was not listed under *-ment, -al* or *-ize*; *commercialism* was listed under *-ism* but not under *-al*). Derived forms with a single prefix and a single suffix caused

[53] You can even insert a space (or underscore or hyphen) after each character of the query string, which does not make much sense in human-computer interaction but may be exploited in an NLP environment.

no problems (e.g. *derailment*). Multi-word WordNet entries (e.g. *air sickness*, *altitude sickness*, *car sickness*, *decompression sickness*, *milk sickness*, *morning sickness*, etc., mostly compounds) were rejected[55].

The total number of WordNet entries that potentially end in one of those affixes exceeds 10,000 and thence it was infeasible to decide whether each of these entries contained the corresponding affix or merely ended in a suffix-like string. Therefore, a random sample of at least 5 per cent of each list was taken (see the "Sample size" column in table 4-1), then I examined the samples looking for word forms that actually contained the affix in question (e.g. *amazement* qualified positively while *condiment* was rejected; see the "Hits" column of table 4-1). This phase was facilitated by my concordancing program, too, since it allows the user to categorize wordlist and concordance items. A correction ratio was computed for each affix, which reflects the ratio of the number of words that actually had the affix in the sample ("Hits"/"Sample size"). Then, the number of words that exhibited the affix was projected to the whole set of candidate words (shown in the "Estimated WN figure" column in table 4-1). Each of the suffixes *-al*, *-ism* and *-ian* can attach to bases belonging to two grammatical categories (V and N for *-al*, Adj and N for *-ism* and *-ian*). I located occurrences of both base types in the samples of these affixes. In this way, it was possible to treat them as distinct suffixes, rather than suffixes that subcategorize with multiple syntactic categories.

Affix		**WN candidates**	**Sample size (words)**	**Hits**	**Estimated WN figure**
-er	(deverbal)	3702	185	112	2241
-ness	(deadjectival)	1920	100	99	1901
-ation	(deverbal)	1581	100	97	1534
-ity	(deadjectival)	908	100	81	735
-ment	(deverbal)	439	100	80	351
-ism	(denominal)	703	100	41	288
-ian	(denominal)	418	100	64	268
-ism	(deadjectival)	703	100	30	211
-al	(denominal)	511	100	36	184
-al	(deverbal)	511	100	16	82
-ee	(deverbal)	224	100	31	69
-ian	(deadjectival)	418	100	1	4

Table 4-1 Noun-forming derivational affixes in WordNet

[54] I used the Sisyphus Concordancer, which I had developed for educational purposes. It supports a full array of basic concordancing options (including alphabet redefinition for various languages) and regular expression searches.

The above results can be directly compared to Baayen and Lieber's (1991) data. Baayen and Lieber's study (partly repeated in Lieber 1992) utilized the English subset of the CELEX database compiled by the Dutch Center for Lexical Information in Nijmegen (cf. Burnage 1990), which is based on a corpus of 18 million words containing written as well as transcribed spoken material (an early snapshot of the Cobuild corpus, cf. Renouf 1987).

Baayen and Lieber studied noun-forming, adjective-forming and verb-forming affixes (22 suffixes and 9 prefixes) by looking up and counting all words that contained the affixes in question. Table 4-2 contains information about the noun-forming affixes they examined. You find the number of different words that exhibit the affix in CELEX in the first column and the Baayenian productivity index computed for each affix in the second column. This productivity index is directly proportional to the number of derived forms featuring this affix *occurring only once* in the CELEX database (hapax legomena), and is inversely proportional to the total number of tokens exhibiting the corresponding affix, i.e. Productivity = number-of-hapax-legomena / number-of-tokens-with-the-affix. According to Baayen, the affix *-ee* is fairly productive (it can easily coin new words); the most productive affixes are *-ness* and the deadjectival *-ian*; the rest is approximately as productive as the class of nouns in general (the productivity index is 0.0001 for simplex nouns, cf. Baayen and Lieber 1992:6). The estimated number of corresponding WordNet entries for these noun-forming affixes is repeated (from table 4-1) in column 3.

Affix		**CELEX word types**	**Productivity index**	**Estimated WN figure**
-er	(deverbal)	682	0.0007	2241
-ness	(deadjectival)	497	0.0044	1901
-ation	(deverbal)	678	0.0006	1534
-ity	(deadjectival)	405	0.0007	735
-ment	(deverbal)	184	0.0002	351
-ism	(denominal)	50	0.0006	288
-ian	(denominal)	27	0.0007	268
-ism	(deadjectival)	82	0.0005	211
-al	(denominal)	45	0.0001	184
-al	(deverbal)	38	0.0001	82
-ee	(deverbal)	23	0.0016	69
-ian	(deadjectival)	16	0.0040	4

Table 4-2 Productivity of noun-forming affixes

[55] The interaction between derivation and compounding is not examined in this experiment; therefore, the effects of compounding should be minimized.

Diagram 4-1 depicts the first and third data columns of Table 4-2, that is Lieber's number of word types with noun-forming affixes in the CELEX database and the estimated number of WordNet entries with these affixes. The WordNet part of Diagram 4-1 is scaled down by a factor of 2 to facilitate visual comparison.

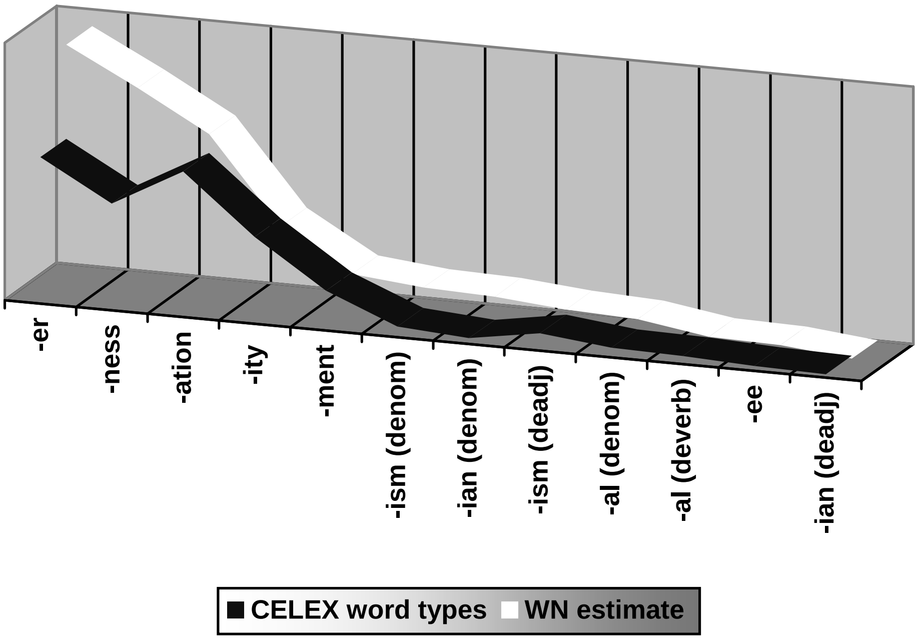


Diagram 4-1

My hypothesis (H_1) is that there is a positive correlational relationship between columns 1 and 3 of Table 4-2 (as depicted by Diagram 4-1). The null-hypothesis (H_0) includes the non-correlational and negative correlational cases. The statistical analysis[56] of these variables supports H_1: we can establish a strong positive correlational relationship between these two sets of data (Lieber's CELEX counts and my WordNet estimates).

The above results show that the CELEX database (that is its 18-million-word corpus source) shows similar proportions of derived nouns as the noun word-stock of WordNet. Also remember that I have had to scale down the WordNet part of the diagram since WordNet contains much more entries exhibiting these affixes than the CELEX database does.

Diagram 4-2 shows the tendency differences between the Baayen index for these derivational affixes and the estimated number of derived entries in WordNet. The Baayen index is scaled up by a factor of 400,000 to facilitate comparison.

[56] Pearson correlation: *0.943*, correlation is *significant* at the 0.01 level (significance: 0.000).

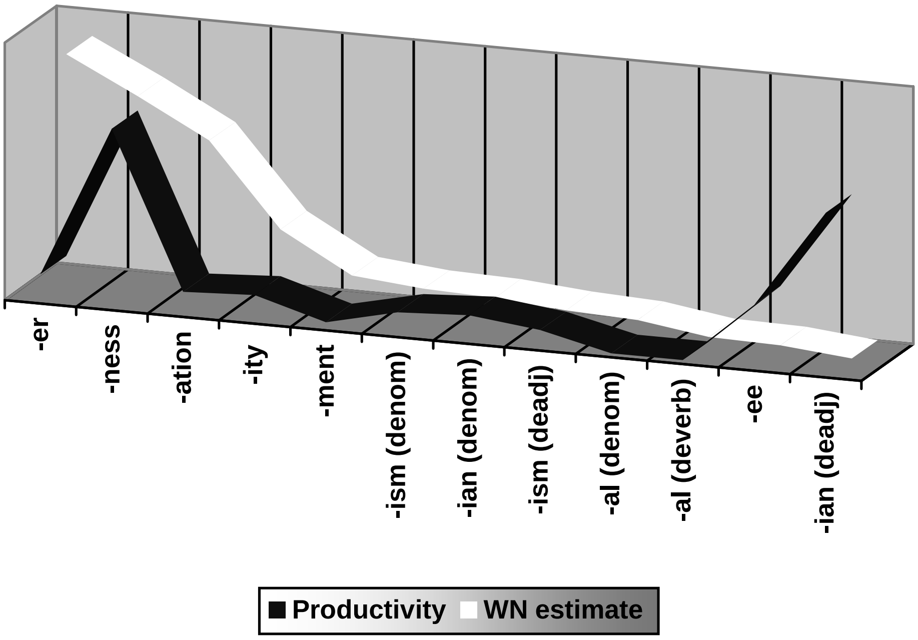


Diagram 4-2

An exploratory investigation has been carried out to verify if it makes sense to look for any kind of relationship between these two variables. According to a statistical analysis[57] of these variables (columns 2 and 3 of Table 4-2, as depicted by Diagram 4-2), it makes no sense to argue that there is a correlational relationship between these two variables, since the correlation test is not significant.

While the term productivity seems fairly well exploited in Morphology, it is often used intuitively in the literature. Chomsky (1965:186) refers to "quasi-productive processes, such as those that are involved in the formation of such words as *horror, horrid, horrify; terror, (*terrid), terrify; candor, candid, (*candify);* or *telegram, phonograph, gramophone*, etc...", which calls for a fuzzy or prototype-based definition (but consider the fundamental difference in treatment between productive and unproductive morphological phenomena offered by many authors, including Chomsky 1965). Aronoff (1976:35), in search for a scientific approach to productivity, emphasizes that it should be more than a simple comparison of the number of words exhibiting different affixes (or words undergoing the given Word Formation Rules). What he suggests instead is that we should list the potential bases the affix can attach to and count those cases in which the affixation does take place. Aronoff's productivity index has not been particularly popular, however.

Baayen's Productivity index (Productivity = number-of-hapax-legomena / number-of-tokens-with-the-affix, see above) is ideal for corpus-based studies. Using the right software tools, it is fairly easy to count word types and tokens, and one does not have to bother with an abundance of morphological and/or phonological restrictions on possible bases. It is useful, too, since it reflects the ease with which a new word can be coined on the spot by the speaker. One may describe the Baayen index as the "anticipated creative use" of a particular affix.

The unsignificant correlational relationship between the number of derived lexical entries in WordNet and the Baayen index is therefore inherent to Baayen's formulation of productivity. The compilers of WordNet cannot prepare for the unexpected; ad-hoc uses cannot be listed.

Derivational affixes with a high Baayen index do not cause major difficulties for the lexicographer compiling a dictionary for human use. Once you list *Asian*, *American* and *Australian* will be anticipated by and implied in the entry. WordNet, however, is built on the idea that all possible entries should be listed so that relations (and semantic information in general) could be added and expressed in an elaborate network. When you browse the list of nouns with the suffix *-ian* in WordNet, you find *Belorussian, Californian, Alabamian, Asian, Australian, Siberian, Brazilian,* etc. Although more than one hundred similar nouns are listed, WordNet just cannot cover all possibilities, not to mention the great variety of multi-word entries (mostly compounds) that may involve these nouns.

It is interesting to find, however, that Baayen's productivity index does not confirm our intuition that the denominal *-ian* (*Californian*, *Alabamian*, etc) is often used in an ad-hoc fashion. The reason for this is that its high creative productivity does not affect all noun bases in general, just geographical names. Coming up with such a restriction on the base is more like an Aronoffian method, however, and is out-of-reach of Baayen and Lieber (1991).

Two affixes, namely *-ness* and the deadjectival form of *-ian* are said to be extremely productive by Baayen and Lieber. The former is well represented in WordNet: the compilers seem to have undertaken the tedious task of listing many of the possibilities. In this sense, WordNet delivers more than traditional dictionaries do: dictionaries cannot list these forms for space limitations. In fact, they need not list them since the human reader's intuitions can easily make up for such a 'deficiency'.

[57] Pearson correlation: 0.218, but correlation is *not significant* (significance: 0.497).

While *-ness* is well-represented in WordNet, the deadjectival version of *-ian* is not, although it has the second highest Baayen index. It means that this affix is hardly used systematically, and only exploited by ad-hoc uses coined on the spot. The combination of the high productivity index and the extremely low representation of this affix in WordNet suggests that deadjectival nouns with the affix *-ian* do not easily infiltrate into the lexicon of English.

WordNet is not constructed to function as a *lexicon* proper: for instance, Predicate Argument Structure is not properly specified for the entries, phonetic/phonological information is missing, and it lacks rule-governed derivational morphology and compounding. On the one hand, an abundance of derived forms and compounds are incorporated, and we can live without Word Formation Rules and/or lexical redundancy rules as long as a matching entry is found in the database for each incoming form (word or phrase). Intuitively, a human listener does live without rule-based morphology (and perhaps syntax) as long as the incoming language chunk can be found in his or her long-term memory in time. On the other hand, however, there must exist a machinery that helps interpret new forms that are not directly "digestible" establishing missing information. If such a machinery was implemented in WordNet, it should facilitate form acquisition (supervised or automatic) rather than mere interpretation due to the full-listing approach inherent in the database.

A related question is the use of redundancy rules within the lexicon. Redundancy rules are useful in accounting for morphological and semantic relations between entries. As Jackendoff put it,

> Lexical redundancy rules are learned from generalizations observed in already known lexical items. Once learned, they make it easier to learn new lexical items: we have designed them specifically to represent what new independent information must be learned.
>
> (Jackendoff 1975:668)

Jackendoff also argues that redundancy rules, once acquired, may assume a more "unusual", creative role,

> … producing a class of partially specified possible lexical entries. For example, the compound rule says that any two nouns N_1 and N_2 can be combined to form a possible compound N_1N_2. If the context is such as to disambiguate N_1N_2, any speaker of English

> who knows N_1 and N_2 can understand N_1N_2 whether he has heard it before or not, and whether it is an entry in his lexicon or not.
>
> (Jackendoff 1975:668)

WordNet lacks redundancy rules. Some people may argue that WordNet glosses contain enough information to establish these rules for at least a few affixes (e.g. the gloss for *maker* is “a person who makes things”, a *ruler* is “a person who rules or commands”, and there is no gloss with a similar structure for *cooker,* which is exactly what we expect). I would still like to hypothesize that unsupervised (i.e. completely automatic) redundancy rule acquisition is not possible in WordNet.

Rule-based lexicon expansion and the related question of the introduction of redundancy rules into the database seem a logical next step in narrowing the gap between a “theoretically adequate” lexicon and WordNet. This gap is fundamental, however. One of the presuppositions behind WordNet is the *separability hypothesis* according to which “the lexical component of language can be isolated and studied in its own right” (Miller 1998:xv). The idea of an autonomous lexical component is challenged, however, by Lieber’s (1992) observations suggesting that “the rules of word formation are in fact the rules of syntax” (Lieber 1992:vii). If this holds true, a unified morphosyntactic device must be interwoven with the structures of the lexicon[58].

A common limitation of the Sense Enumerative Lexicon pointed out by Pustejovsky (1995) is that they do not account for the creative use of words. Traditional lexicography is not trapped by this phenomenon, since it is sufficient to list some examples, and the human dictionary user will recognize and even produce analogous forms. The human factor is missing from NLP, however. The feasibility of the implementation of an automatic *derived form acquisition device* is also dubious (at least, we may not be able to formulate all restrictions on the possible stems in terms of rules). Let me point out, however, that version 2.0 of WN has introduced the “derivationally related forms” relation, which may allow linguists to find regularities within the database and predict (acquire) new forms. Compounds are even more problematic to list, although this is what some theoretical works, including Jackendoff (1975) proposed. WN does include many compounds, but their compositionality is not indicated, so regularities are impossible to find.

Palmer (1998) observes that “world-knowledge” has been a factor in editing WordNet synsets. Another cognitivist feature which has appeared in the latest version of WN is the

[58] We do not disagree with the following statement, however: “useful contribution can be made at the level of words” (Miller 1998:xv).

support for *domains*. It seems to be experimental at this stage, and it is only provided for a small number of synsets. WN 2.0 also lacks multi-domain enrollment[59]. As a complementary, and possibly more promising approach, Frame Semantics has been introduced into NLP in the form of a large lexical database called FrameNet, compiled under Charles Fillmore's supervision.

4.2 FrameNet

FrameNet (Baker, Fillmore and Lowe 1998) is a project based on Frame Semantics (cf. Fillmore 1968, 1976; Fillmore and Atkins 1994), which aims to group words according to underlying conceptual structures (called *frames*) using corpus evidence. According to Johnson et al. (2004), the first release consisted of 6,000 lexical units[60] and 130,000 annotated sentences mainly taken from the 100-million-word British National Corpus marked up in XML by the FrameNet team. The underlying concepts of FrameNet are the following:

> A **lexical unit** is a pairing of a word with a meaning. Typically, each sense of a polysemous word belongs to a different **semantic frame**, a script-like structure of inferences that characterize a type of situation, object, or event. In the case of **predicates** or **governors**, each annotation accepts one word in the sentence as its **target** and provides labels for those words or phrases in the sentence which fill in information about a given instance of the frame. These phrases are identified with what we call **frame elements** (FEs) - participants and props in the frame whose linguistic expressions are syntactically connected to the target word.
> (Johnson et al. 2004)

Synonymy is treated in the following way: a lexical unit (LU) corresponds to a single sense of a word; variations of word meaning can be considered as belonging to a single sense as long as they fit the Frame and Lexical Unit specifications the editors have come up with. Meaning variations that do not fit the LU are treated in separate LUs. LUs are not related to one another, but the frames that "host" them may be related (see the section on frame relations below). In fact, the editors do not seek to describe non-fitting cases: the

[59] I would like to hypothesize, however, that in a critical mass, perhaps augmented by hypernym and troponym hierarchies, (Langackerian) conceptual domains could be constructed more or less automatically.

[60] As of writing this thesis, a slightly enhanced version (v1.1) is the current release.

extension of the set of available frames has priority over other means of database expansion.

The procedure of developing FrameNet can be summarized in the following way. Corpus sentences are selected that contain the lexical unit being investigated in a way that represents “the full range of combinatorial possibilities for that LU” (Johnson et al. 2004). Having selected the target word[61], linguists annotate the sentences for Frame Element (FE), Phrase Type (PT), and Grammatical Function (GF) relative to the target. Only those constituents are annotated that are related to the target. Automatic processes are also introduced to provide grammatical information on pre-selected phrases. The following example shows two possible valence patterns (in terms of FEs, PTs and GFs) for the verb *give* **(**taken from Fillmore, Johnson and Petruck 2003:238**):**

give	*FEs:*	Donor	Theme	Recepient
	PTs:	NP	NP	NP
	GFs:	Ext	Comp	Obj
give	*FEs:*	Donor	Theme	Recepient
	PTs:	NP	NP	PP-to
	GFs:	Ext	Obj	Comp

The list of *Phrase Types* include Noun Phrases (standard, possessive, non-referential), Prepositional Phrases, Verb Phrases (finite and nonfinite: bare stem, to-infinitive, gerundive), Complement Clause (finite: that, which, whether/if clauses; nonfinite: to-marked, bare stem, for-to-marked, gerundive), Subordinate Clause, Adjective Phrase types, Adverb Phrase, Quantifier Phrases and a Quote phrase type. Johnson et al. (2004) point out that this phrase-set has been developed to capture lexical information in the following way:

> In choosing the phrase types and grammatical functions to use, the major criterion was whether or not a particular label might figure into a description of the grammatical requirements of one of the target words of the project. The emphasis on what is relevant to lexical descriptions means that we limit ourselves, for the most part, to those phrase type labels which might appear in subcategorization frames. We do not include a complete list of all phrase types as would appear in more theoretically oriented syntactic descriptions
>
> (Johnson et al. 2004)

[61] In FrameNet II, phrasal verbs and idioms also qualify as targets (cf. Fillmore, Wooters and Baker 2001).

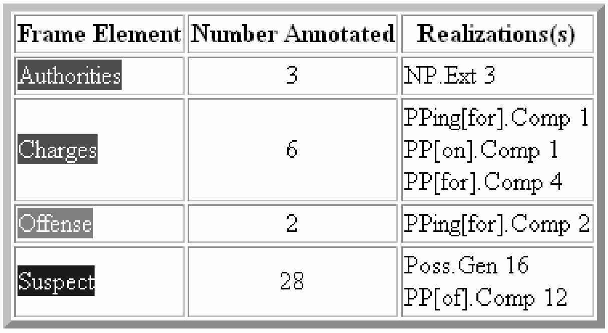
Grammatical Function annotation is also done relative to the target word. The following table contains the possible Grammatical Functions for every target type.

Verb targets	*Adjective targets*	*Preposition targets*	*Noun targets*
External Argument Object Complement Modifier	External Argument Head noun modified by attributive adjective Complement Modifier	External Argument Object	External Argument Complement Genitive determiner Modifier Appositive

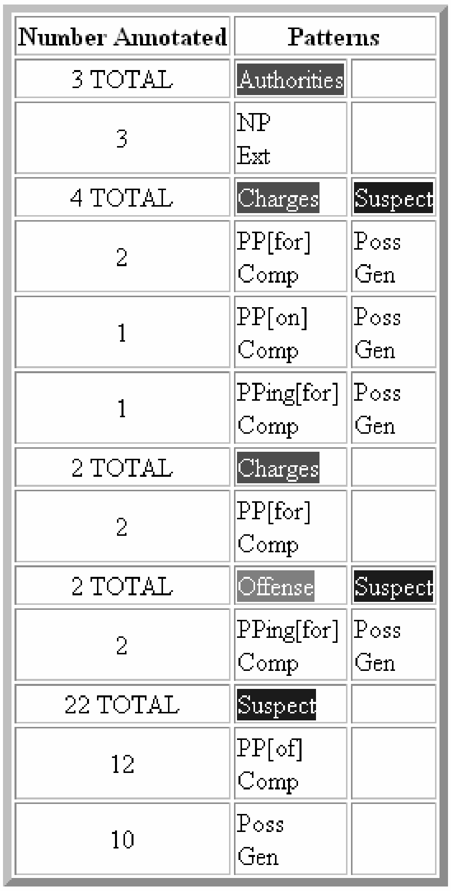
As a general rule, targets are not annotated for PT and GF, but they get Frame Element labels during the annotation process.

While annotated sentences (with FE/PT/GF labels) can be extracted from XML data for NLP applications, the editors have prepared lexical unit and frame specifications that make it easier for the human reader to study the database.

Lexical unit specifications contain 1) a references to the frame that “hosts” the LU, 2) a lexical entry, and 3) an annotation report: a list of corpus sentences annotated with Frame Element labels grouped together according to Phrase Type. A *lexical entry* (see Figure 4-1) contains a *definition* and an overview of possible *valence patterns*; the latter contains the syntactic patterns in which the frame elements occur.



Frame Element	Number Annotated	Realizations(s)
Authorities	3	NP.Ext 3
Charges	6	PPing[for].Comp 1 PP[on].Comp 1 PP[for].Comp 4
Offense	2	PPing[for].Comp 2
Suspect	28	Poss.Gen 16 PP[of].Comp 12



Number Annotated	Patterns	
3 TOTAL	Authorities	
3	NP Ext	
4 TOTAL	Charges	Suspect
2	PP[for] Comp	Poss Gen
1	PP[on] Comp	Poss Gen
1	PPing[for] Comp	Poss Gen
2 TOTAL	Charges	
2	PP[for] Comp	
2 TOTAL	Offense	Suspect
2	PPing[for] Comp	Poss Gen
22 TOTAL	Suspect	
12	PP[of] Comp	
10	Poss Gen	

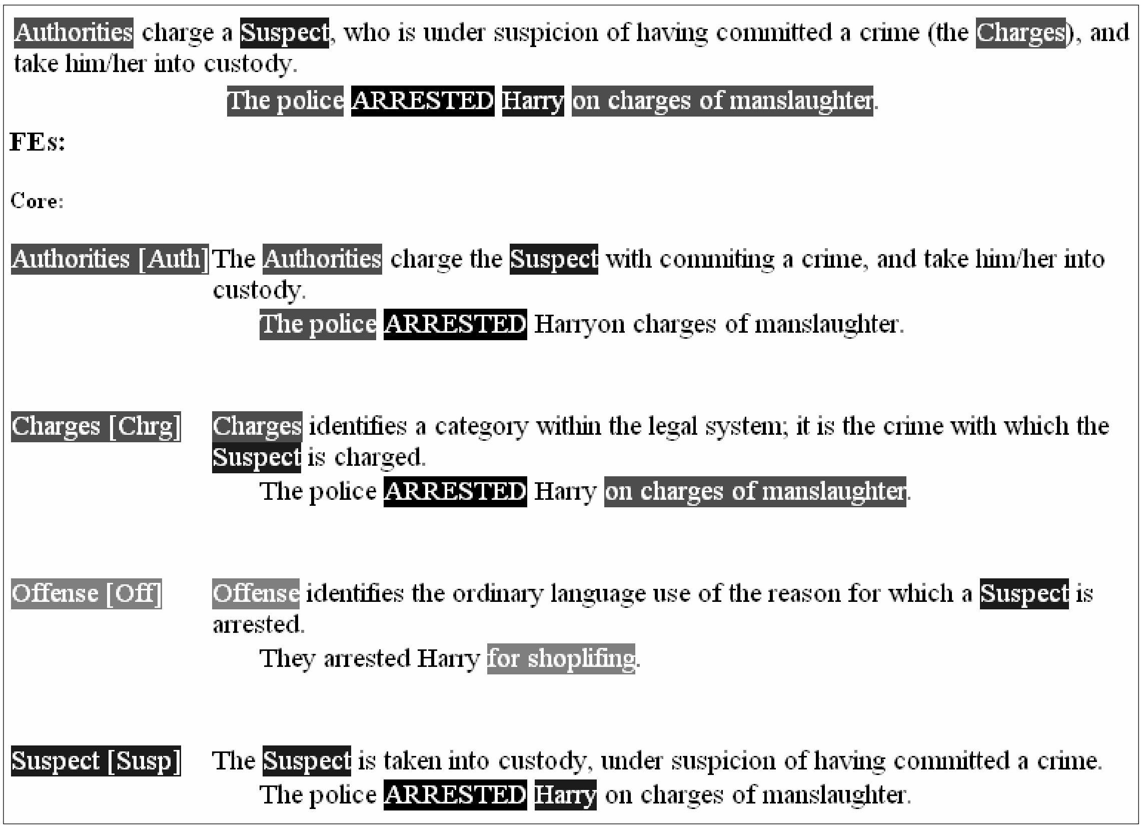
Figure 4-1 Lexical entry for the noun *arrest*: the *definition* part is on the left, the *valence patterns* section is on the right (screen capture of FN web-interface output, colors are replaced by gray levels)

Consider the following sentence (taken from the annotation report for the lexical unit *arrest.n:*

His ARREST was ordered by the Algiers judiciary after the Ministry of Defence accused him of inciting the army to mutiny.

Arrest is the target word in the sentence, and *his* takes the *Suspect* Frame Element of the *ARREST* frame. The sentence is listed in the “Poss” section of the annotation report, “Poss” being the Phrase Type label (an NP subtype, see above) of *his* in this sentence.

In addition to the lexical unit specifications, the editors have prepared a second access method to the database, which is based on frame specifications. Primarily, frame specifications summarize FE options. *Arrest.n* evokes the *ARREST* frame, which has the following data in the present version of FrameNet:



Authorities charge a Suspect, who is under suspicion of having committed a crime (the Charges), and take him/her into custody.

The police ARRESTED Harry on charges of manslaughter.

FEs:

Core:

Authorities [Auth] The Authorities charge the Suspect with commiting a crime, and take him/her into custody.

The police ARRESTED Harryon charges of manslaughter.

Charges [Chrg] Charges identifies a category within the legal system; it is the crime with which the Suspect is charged.

The police ARRESTED Harry on charges of manslaughter.

Offense [Off] Offense identifies the ordinary language use of the reason for which a Suspect is arrested.

They arrested Harry for shoplifing.

Suspect [Susp] The Suspect is taken into custody, under suspicion of having committed a crime.

The police ARRESTED Harry on charges of manslaughter.

Figure 4-2 Frame specification for the *ARREST* frame, part 1[62]

Also part of the frame specification is a list of related lexical units and the description of frame relations:

apprehend.v, apprehension.n, arrest.n, arrest.v, book.v, bust.n, bust.v, collar.v, cop.v, nab.v

Inherits From:	*INTENTIONALLY AFFECT*
Subframe of:	*CRIMINAL PROCESS*
Uses:	*CAUSE CONFINEMENT*
Is Used By:	*SURRENDERING*

Figure 4-3 Frame specification for the *ARREST* frame, part 2

The following is an explanation of the frame-relations that are used in the above example:

[62] Please note that the examples shown in the frame specification (for the frame *ARREST* in our example) involve a pre-selected target word (in this case, *arrest.v*) and not the lexical unit through which we reach the frame specification (in our example, *arrest.n*, but *apprehend.v*, *apprehension.n*, etc. also evoke the same frame).

- The *ARREST* frame inherits the Frame Elements of the *INTENTIONALLY AFFECT* frame (including the *Agent* and *Patient* FEs). Inheritance is used when a frame “is a more specific elaboration of the parent frame” (Johnson et al. 2004);
- The *ARREST* frame is part of a complex frame called *CRIMINAL PROCESS*. Complex frames usually designate “sequences of states of affairs and transitions between them” (ibid.). The sequence (which may include conditional branching, too) is made explicit in the definition of the complex frame. The *CRIMINAL PROCESS* complex frame has the following subframes: *ARRAIGNMENT, ARREST, SENTENCING* and *TRIAL*. The general FN policy of annotating individual *sentences* still applies, however. As the compilers put it, “[f]or our purely lexicographic purposes … we see no need to examine structures larger than the sentence” (FrameNet Frequently Asked Questions: “Your lexicon is based on sentences taken one at a time. Why don't you look at longer texts?”, n.d.).
- The *ARREST* frame makes reference to a more abstract (”schematic”) frame: *CAUSE_CONFINEMENT* and is used by the *SURRENDERING* frame.

The system of frame relations also helps to overcome the difficulties arising from an important feature of FrameNet’s design: frame elements are relative to frames. It is due to this design feature that FE labels introduced in different, unrelated frames under the same name may or may not be related. This issue is also discussed on the FrameNet website:

> Strictly speaking the frame element names proposed for one frame are relative to that frame, so decisions about choosing labels that are also used in other frames are always reparable. We want the cross-frame recycling of frame element names to be justified, ultimately, through establishing principles of frame inheritance. The picture is complicated, of course, because of the possibility of multiple inheritance: the same argument of a single predicate can be seen as an instance of one frame element by virtue of its membership in one frame, of another frame element through its participation in a different co-existing frame.
>
> (FrameNet Frequently Asked Questions: “Doesn't this frame-specific approach lead to multiple names for what is really the same frame?”, n.d.)

Gildea and Jurafsky (2002) have introduced a set of 18 abstract cases that replaces the frame-specific FrameNet roles in one of their FN-based experiments. The list of the roles is the following (taken from Gildea and Jurafsky 2002:280):

Role	*Example*
AGENT	**Henry** *pushed* the door open and went in.
CAUSE	Jeez, **that** *amazes* me as well as riles me.
DEGREE	I **rather** *deplore* the recent manifestation of Pop; it doesn't seem to me to have the intellectual force of the art of the Sixties.
EXPERIENCER	It may even have been that **John** *anticipating* his imminent doom ratified some such arrangement perhaps in the ceremony at the Jordan.
FORCE	If this is the case can it be *substantiated* **by evidence from the history of developed societies**?
GOAL	Distant across the river the towers of the castle rose against the sky straddling the only land *approach* **into Shrewsbury**.
INSTRUMENT	In the children with colonic contractions **fasting motility** did not *differentiate* children with and without constipation.
LOCATION	These fleshy appendages are used to detect and *taste* food **amongst the weed and debris on the bottom of a river**.
MANNER	His brow *arched* **delicately**.
NULL	Yet while she had no intention of surrendering her home, **it** would be *foolish* to let the atmosphere between them become too acrimonious.
PATH	The dung-collector *ambled* slowly **over**, one eye on Sir John.
PATIENT	As soon as a character lays a hand on this item, the skeletal Cleric *grips* **it** more tightly.
PERCEPT	What is *apparent* is **that this manual is aimed at the non-specialist technician, possibly an embalmer who has good knowledge of some medical procedures**.
PROPOSITION	It says that rotation of partners does not *demonstrate* **independence**.
RESULT	All the arrangements for stay-behind agents in north-west Europe collapsed, but Dansey was able to *charm* most of the governments in exile in London **into recruiting spies**.
SOURCE	He heard the sound of liquid slurping in a metal container as Farrell *approached* him **from behind**.
STATE	Rex *spied* out Sam Maggott **hollering at all and sundry and making good use of his over-sized red gingham handkerchief**.
TOPIC	He said, "We would urge people to be aware and be *alert* **with fireworks** because your fun might be someone else's tragedy."

It is important to underline that Gildea and Jurafsky have introduced these abstract roles to boost the performance of their shallow semantic interpreter while they also acknowledge the advantages of the frame-specific role assignment method implemented in FrameNet. They argue that FN describes semantic roles at an intermediate level (Gildea and Jurafsky

2002:249) between an abstract, general level of thematic role representation and a verb-specific thematic-role description system that features thousands of roles.

> Defining semantic roles at this intermediate frame level helps avoid some of the well-known difficulties of defining a unique small set of universal, abstract thematic roles while also allowing some generalization across the roles of different verbs, nouns, and adjectives, each of which adds semantics to the general frame or highlights a particular aspect of the frame. One way of thinking about traditional abstract thematic roles, such as AGENT and PATIENT, in the context of FrameNet is to conceive them as frame elements defined by abstract frames, such as *action* and *motion*, at the top of an inheritance hierarchy of semantic frames.
> (Gildea and Jurafsky 2002:249)

Finally, it should be noted that the editors of FrameNet had hoped to connect their database to the WN system. The FrameNet website discusses the original idea of connecting the two:

> Originally the hope was that WordNet synsets would serve as partial wordlists for particular frames …, and that FrameNet's unique contribution would be to expand the sentence templates by including all of the sentence types built around a word … The novelty would be that the sentence templates would not be limited to basic syntactic elements but would include linking between semantic relations and syntactic realization, while at the same time providing not just schematic templates but actual attested sentences illustrating each of these.
> (FrameNet Frequently Asked Questions: “What is the relation between FrameNet and WordNet?”, n.d.)

However, it “turned out to be easier” (ibid.) to come up with a new system than to use the categories of WordNet.

The goal of connecting the two databases was considered again, when the second (current) phase of the FrameNet project was outlined (cf. Fillmore, Wooters and Baker 2001). This feature has remained unimplemented, however. As the FrameNet website points out, the editors have recognized that the sense divisions found in WordNet “too often” mismatches the senses they have discovered (cf. FrameNet Frequently Asked Questions: “Why don't you use WordNet sense divisions?”, n.d.).

Finally, I would like to discuss the concept of lexical meaning underlying the FrameNet enterprise. Although Fillmore, Johnson and Petruck (2003:235) point out that the term "lexical unit" was borrowed from Cruse (1986), I cannot establish a more useful connection between Cruse's system and FrameNet. FN is built on *frame semantics,* which is "based on the idea that word meanings are organized around schematic conceptual scenarios, or *frames*, that underlie the use and interpretation of the lexical items and their general complementation and modification properties" (Fillmore, Johnson and Petruck 2003:241, emphasis original). The approach they follow also determines how they see the lexicon, which is summarized in the following excerpt:

> [T]he empirical work[63] allows us to develop a perspective on the lexicon, specifically one that is based on the uncontroversial assumption that to understand word meaning we must first have knowledge of the conceptual structures, or semantic frames, which provide the background and motivation for their existence in the language and their use in discourse.
>
> (Fillmore, Johnson and Petruck 2003:247, emphasis added)

4.3 MindNet and the question of representing Polysemy in a relational lexical database

The previous sections on WordNet and FrameNet have introduced rich databases that are freely available to the research community and have earned high reputation. MindNet (cf. Dolan, Vanderwende and Richardson 2000) is Microsoft's relational lexicon, and as such, is similar to WordNet to some extent, but the differences are considerable and noteworthy.

The MindNet database is derived automatically from formidable machine-readable sources: the *Longman Dictionary of Contemporary English* and the *American Heritage 3rd Edition* dictionaries, and is also augmented by the full text of Microsoft Encarta. The derivational process is carried out by a parser[64] that compiles syntactic trees and 'logical forms' (LFs). Logical forms are "directed, labeled graphs that abstract away from surface word order and hierarchical syntactic structure to describe semantic dependencies among content words" (Dolan, Vanderwende and Richardson 2000:7). The database can be treated and exploited as a relational lexicon that contains "about 25 semantic relation types …,

[63] It refers to the use of corpus sentences and the accompanying compilation method, which is built on discovering things intuitively (Fillmore, Johnson and Petruck 2003:246).

including *Hypernym*, *Logical_Subject*, *Logical_Object*, *Synonym*, *Goal*, *Source*, *Attribute*, *Part*, *Subclass* and *Purpose*" (p. 7). The relationships between the 'root word' of the LF (corresponding to the headword of the source MRD entry) and other words stored in the corresponding LF structure is expressed by a semantic relation and constitute a direct *path*[65], whereas the 'non-root' words of the LF are also connected to each other by (indirect) paths (p. 9). The following example illustrates a path between *car* and *person* (p. 9):

car←Logical_Object–*drive*–Logical_subject→*motorist*–Hypernym→*person*

Extended paths can be found between words of different LF graphs. For instance, we can join the following two paths (each from a different LF): *car*–Hypernym→*vehicle* and *vehicle*←Hypernym–*truck* into the extended path *car*–Hypernym→*vehicle*←Hypernym–*truck* (p. 9-10).

There is no explicit hierarchy of concepts in MindNet, but it offers a similarity measure that shows how similar two words are in some context (p. 13). Similarity is computed using the paths with the highest weights. The editors of MindNet collect and store information about the patterning of word pairs with known similarity (p. 14).

Chapter 4 of this thesis remains rather isolated from the preceding chapters. As far as WordNet is concerned, we have been able to analyze its content in morphological terms (inflection and derivation). We have also seen that it only handles homonymy (which is implemented by listing the same word form in multiple synsets), but it lacks any device to handle polysemy. To make up for the lack of a better way of accounting for polysemy and neighboring phenomena described in Cruse (1986, 2000; see section 3.3 and section 3.4 of this thesis), WN has to use extremely fine-grained sense distinctions. Fellbaum (1998) explains this property of WordNet in the following way:

> Other lexical semanticists have undertaken careful analyses of semantic and lexical relations and proposed subtle distinctions (Cruse 1986). These distinctions are valid in the context of a semantic analysis of conceptual relations, but they do not seem to be reflected in speakers' minds, where relatively few relations are salient.
>
> (Fellbaum 1998:10)

[64] The authors point out that the same parser is exploited in Microsoft Word 97. No further details are given about the parser or the parsing techniques.

[65] Paths are weighted on the basis of the frequency of the associated semantic relationships: "middle frequency" relations are favored over low and high frequency relations (p. 10).

In FrameNet, lexical units may be listed many times (and they may belong to different frames): this is the only way to account for different variations of sense. However, as far as MindNet is concerned, Dolan, Vanderwende and Richardson connect their research to Cruse's (1986) theory of lexical meaning. Their position is the following:

> A fundamental assumption underlying … MindNet's approach to lexical representation, is that *there is no such thing as a discrete word sense*. Instead, there are only usage patterns, and the system's understanding of a word's meaning is nothing more than the pattern of activation over the semantic network. While this runs counter to much current work in WSD, it directly parallels Cruse's notion of *sense modulation*
>
> (Dolan, Vanderwende and Richardson 2000:16, emphasis original)

They argue that they implement Cruse's sense-spectra (cf. section 3.3), which are amoeba-like objects of a continuous nature (Dolan, Vanderwende and Richardson 2000:6, 15). From a practical point of view, all they do is store the LFs *without disambiguation* (thereby eliminating disambiguating errors or the human intervention that would find and correct the errors). "[D]efinition and example sentence LFs within MindNet are allowed to overlap freely on shared words" (p. 21).

MindNet is pre-trained using two machine-readable dictionaries, but it is also augmented by additional knowledge sources (most importantly, MS Encarta) that deliver unknown words, too. MindNet stores the typical usage patterns of the new words and links them to usage information about known words.

> A word's meaning is nothing more than 'the company it keeps'[66], but this 'company' involves more than statistical co-occurrence information. Instead, context in our terms is a richly annotated linguistic analysis that normalizes long-distance dependencies, resolves intrasentential anaphora, and provides labeled relationships linking content words. Given this strong notion of lexical context, even a small number of encounters with a word can potentially provide a very detailed notion of what it must mean.
>
> (Dolan, Vanderwende and Richardson 2000:36)

Certain tasks of Natural Language Processing seem especially well suited for MindNet's approach. The authors often mention (and seem to analyze the needs of) *information retrieval*, which does seem a fitting task, since the query strings in an information retrieval system can be good sources of context to be matched to usage patters in the MindNet database. The authors also keep referring to *machine translation*, but they do not make it clear how they think it is feasible to use usage patterns stored in their database to gain target language patterns.

4.4 Word meaning as spreading activation

MindNet's approach to storing lexical meaning in a huge network is partly similar to the spreading activation network described by Véronis and Ide (1990). Véronis and Ide exploit the definitions of the *Collins English Dictionary* in the following way: each headword of the dictionary is represented by a "word" node (neuron), which is connected to nodes that stand for the senses that are listed in the dictionary for the headword. Each sense node is connected to all words that are present in the *definition* of that particular sense (words are lemmatized and function words are excluded). The words of the definition are themselves nodes with a sense-node structure; the resulting network is restricted to "a few thousand" nodes in the experiment, which is still a huge network. The system features inhibitory links between the sense nodes belonging to the same headword.

Querying ("running") the network involves the activation of at least two word nodes. The nodes activate the sense nodes, and through these sense nodes, huge subnets of word nodes (and that of the sense nodes accompanying them) can be activated. When paths are found connecting the initially activated nodes, those neurons that are along these paths will get more and more activated in multiple passes of spreading activation. The inhibitory links between the sense nodes (of a word node) will help the network to reach a stable configuration in which only one sense node per word is activated. At this final stage, the input words are disambiguated.

Note that the network built by Véronis and Ide is not only a storage space for information but also a query system, which means that no external tools are required to retrieve data from the network. Compiling the network means building an artificial neural network with the right topology, which is determined by the headword definitions in the source dictionary. Also note that the system works with unannotated input, i.e. part-of-speech labeling or syntactic parsing of the input is not required. MindNet and Véronis and Ide's neural network are for different purposes: MindNet outputs a similarity value which is useful for information retrieval and possibly for other "high-level" NLP tasks, while

[66] Taken from Firth (1957), cf. Dolan, Vanderwende and Richardson (2000:15)

Véronis and Ide's network model is created to carry out word-sense disambiguation, which is a major NLP subtask.

Natural Language Processing remains in perspective in the second part of the thesis. The next chapter sheds light on the nature of connectionism and its potential role in representing language.

5. ARTIFICIAL NEURAL NETWORKS IN LINGUISTICS

5.1 Introduction

Connectionist research utilizing Artificial Neural Network (ANN) models has been motivated by the enormous parallel processing power realized by the human brain. Although ANNs have not nearly reached the complexity of biological neural networks, many scientists find them useful in solving a growing set of problems.

Some early linguistic ANN models used strict *localist* representations: certain (linguistically) relevant symbols were embodied by specific units in the input, output and in some cases, even the hidden layers of the network. In such a system, the symbols and the corresponding ANN units are pre-designated, and this selection is external to the system. As an example, section 5.4 will briefly introduce McClelland and Rumelhart's (1981) model.

A different approach was propagated by the Parallel Distributed Processing (PDP) research group. They published two volumes of very influential papers in 1986 (Rumelhart, McClelland and the PDP Research Group 1986, McClelland and Rumelhart and the PDP Research Group 1986) based on the *distributed* connectionist approach. In a well-known paper, for instance, Rumelhart and McClelland described a sub-symbolic network that acquired English past tense verb forms "like children do", through the following stages: memorization, overgeneralization, and finally, normal production (Rumelhart and McClelland 1986). They soon received fierce criticism. Pinker and Prince (1988) claimed that Rumelhart and McClelland's network has the following weaknesses: it is unable to learn many existing rules, it learns rules that are not found in human languages, it cannot explain regularities in the morphology/phonology of past tense forms, it makes incorrect explanation for some developmental phenomena, etc. They go on to argue that these weaknesses are due to the connectionist system architecture itself.

Bullinaria reports that his distributed connectionist (PDP) system, which has no explicit lexicon, produces reliable lexical decisions allowing for "two distinct causes of priming: semantic priming due to semantic vector overlap and associative priming due to word co-occurrence during learning" (Bullinaria 1995:68). In this experiment, the input group takes a vector that accepts an onset consonant, a vowel and an offset consonant cluster index, so the input is localist at the phonetic level (the system is limited to monosyllabic words by design, and it is also limited to the 20 most frequent different consonant onsets, 20 consonant offsets, and 10 vowel clusters). These input patterns are generated for 200 monosyllabic words. A rather large hidden layer and standard backpropagation training (cf. Rumelhart, McClelland and the PDP Research Group 1986) with cross-entropy error is employed. A vector of 27 binary digits corresponding to semantic microfeatures is the output. He reports that output units have a tendency to take intermediate values between 0 and 1. This fuzzy output, characteristic of many ANN systems, is clearly a problem for Bullinaria since he assesses priming in terms of lexical decision performance. I would like to point out, however, that fuzzy classification exhibited by ANN systems is a feature rather than an error, although traditional research tasks (including the lexical decision task) may not be fully compatible with fuzzy output.

Levy highlights two aspects of PDP models (Levy 2002:4): (a) they use a special representation that is distributed, i.e. “the representation of a given entity or concept is not localized in some particular piece of memory, but instead spread over the entire network, or a large part of it”, and (b) early models work with fixed-width representations including the input groups. The latter feature is, in fact, a limiting factor in system design. Consider McClelland and Rumelhart (1981) and Bullinaria (1995) as examples: the former system works with words containing exactly four letters, while the latter is restricted to taking monosyllabic words of one onset consonant cluster, one vowel cluster and one offset consonant cluster. Representing input using *feature* vectors does not solve this problem; a technique for working with input *sequences* is discussed at the end of this chapter.

In what follows, we will briefly survey some interesting features of connectionist systems and examine selected neural network models to put my own frame-recognizing ANN system presented in chapter 6 into appropriate context. The models discussed here are also selected to demonstrate an ANN-specific dilemma that has become a concern in the literature: the appropriate representation of linguistic input.

Paláncz (2003) contains an excellent introduction to neural networks. A classification of major ANN models can be found in Borgulya (1998) and Vörös (1997). The description of the perceptron model (a simple feed-forward model) is discussed at the level of algorithms in the first chapters of Adámek (2002). Rohde (2002) gives a summary of linguistic ANN implementations including production and comprehension systems (parsing, sentence comprehension, word prediction, etc.). Langacker (1991:525-537) discusses the role of connectionism in cognitive linguistics, while Pléh (1998) elaborates the place of connectionism in the wider context of cognitive science.

5.2 Connectionism

Today, all linguists have a firm idea of how structures can be built out of (atomic) *symbols* using rules. Noam Chomsky has been successful in introducing a very compelling way of seeing natural languages (as well as artificial languages) as a result of generation and other symbol-manipulation *rules*. These rules are applied recursively until a whole sequence of terminal symbols is reduced into the start (e.g. the sentence) symbol, or vice versa, until the start symbol is rewritten as a sequence of terminal symbols only.

Connectionism is presented in this chapter as an alternative approach. The following survey of the general properties of connectionism is from Pléh (1998:176-179; translated from Hungarian):

- Connectionist systems use neural modeling: the “units of processing” are not symbols but the activation patterns of neuron-like units in a structure that mimics an abstract nervous system.
- All knowledge is represented by the activation of nodes and the (facilitating or inhibitory) connections between them.
- Networks connections are unlabeled; there are no “connection types” corresponding to semantic relations [such as the lexical and semantic relations in WordNet].
- No symbols – no rules processing: the “external world” of rules and symbols is not inherent to a connectionist system.
- Connectionist processing facilitates parallel activation of nodes and pathways and involves the competition of the activated network elements.
- Knowledge corresponds to activation patterns.
- Representation is distributed over multiple units. Partial representation of concepts is possible (when a more complete representation does not make sense for lack of appropriate input) with retaining the processing capabilities.
- Learning is done through the appropriate adjustment of [connection] weights during training.

The above list foreshadows the implementational peculiarities of connectionist systems that are discussed in the remaining of the chapter.

We will see in section 5.5 that connectionist models may exhibit a very peculiar memory effect, which allocates and shares network resources to create a working memory automatically, as required to solve the task. Rohde points out that the network “must first learn that it has a memory, meaning an ability to make use of information from a previous state, and then it must learn how to use that memory", Rohde 2002:12). He also compares this process of allocating network resources (for implementing memory) to the "memory as preexisting commodity" view:

> If memory is a commodity that can be easily expanded, as in most symbolic architectures, why has evolution settled for such a small bound on our working memory capacity for language and for other cognitive tasks?
> (Rohde 2002:9)

A properly trained connectionist system is able to reproduce the desired output for each input *and* to make precise guesses for novel inputs: they exhibit a remarkable ability to generalize to never-seen input patterns (this feature has been exploited in real-life applications such as stock-market prediction[67]). At the level of individual network units, processing involves collecting information from multiple sources: it is up to each unit to find the right source of information, since there are no pre-designated, "omnipotent", central processing units that could control this process.[68]

5.3 Artificial Neural Networks

In practice, connectionist models are implemented using Artificial Neural Networks. An ANN *unit* (also called *neuron, node*) is connected to other units by weighted links: this is to imitate dendrite and axon connections in biological neural networks[69]. A simple computation within each individual cell produces an output, which is in turn transmitted to connecting units in the network via the links. Units with the same function are often collected into groups, making it unnecessary to define units and unit connections one-by-one. These neuron groups form layers. In most cases, each unit of a layer has connections to all units in the next layer (Rohde 2002:10). *Input* and *output layers* have special functions: training examples contain *input patterns* to be fed into the input group(s) and *target patterns* that are supposed to appear in output group units for the corresponding input[70]. To achieve this goal, link weights are set in multiple passes of *training*. To assess the accuracy

[67] Commercial solutions (as of July 31, 2005) include the offerings of neuroshell.com, which are available at http://www.neuroshell.com/products.asp?task=comparison, a program from tradetrek.com, which can be found at http://www.tradetrek.com/education/ai/ai_stock_trading03.asp, and many more.

[68] For a survey of additional features see Rohde (2002:8-12), who compares symbol manipulation and connectionism, emphasizing certain differences between the two, and attributing certain hypothetical advantages to the connectionist approach.

[69] Using negative links, it is also possible to create an inhibitory effect.

of the network during training and testing, output groups usually use an error measure, which is a function of the target output and the actual output of the output layer. Some models (including Kohonen's Self-Organizing Map model) are trained in an *unsupervised* fashion, i.e. without specific target values, and they adjust weights without the usual error-computing procedure to accommodate known and novel input.

While feed-forward networks exhibit a direct flow of information from the input layer to the output layer, *recurrent networks* feature connections that form cycles. These cycles allow the network to integrate new information with old information (Rohde 2002:10).

Industry-standard computer hardware technology is designed for minimal or no parallel processing, and their programs are carried out by powerful central processing units. ANNs, on the other hand, assume a large number of fairly simple processing units. The flexibility and power of ANNs are due to the large number of interconnected processing units and the vast number of connections. This parallel computation can, however, be *simulated* in software by today's computers. I have used Douglas Rohde's "light, efficient network simulator" (LENS, Rohde 1999a) in my research, which is freely available for research purposes. Rohde (2002) claims that this program is one of the fastest and most efficient ANN simulators available today. I have also found it extremely stable, robust and flexible[71].

5.4 The Interactive Activation (IA) model of Letter Perception

McClelland and Rumelhart's (1981) ANN system is constructed to model human letter perception in a word recognition task. Each layer of the model has direct linguistic interpretation. The system takes orthographic features of four alphabetical characters comprising a single word as input, corresponding to the presence of lines in a special font type (special orthography) that should be used in their experiment. These features are translated into four letters (one unit represents each potential English letter in each

[70] Layers with no input or output functionality are referred to as *hidden layers.* Target patterns are only used for methods based on supervised learning.

[71] Lens supports an extendible set of input and output functions for defining ANN units (groups). It is up to the input function of a unit to compute an input value from values coming through the incoming connections. Main input functions include DOT_PRODUCT, DISTANCE, and PRODUCT. Output is computed by the output functions, including the following main types: LINEAR, LOGISTIC, TERNARY, TANH, EXPONENTIAL, SOFT_MAX, KOHONEN, and OUT_BOLTZ. In addition to the usual ANN linking method that employs links between each unit of the source group connected to all units of the target groups (FULL projection), Lens supports partial connections as well (e.g. RANDOM, FAN, ONE_TO_ONE), and links can be defined in a per-unit fashion, too. Available error functions include SUM_SQUARED, CROSS_ENTROPY, and DIVERGENCE.

character position). Each character position has its own group of units: the letter feature units excite or inhibit the units in the letter group that corresponds to their character position in the input word. The four letter groups are then mapped into words: one unit corresponds to each word of a pre-compiled vocabulary of four-letter words. The activation levels of these units comprise the output of the system.

The layers are interconnected in the following way:

– Feature layer → Letter layer excitatory and inhibitory links
– Letter layer → Word layer excitatory and inhibitory links
– Word layer → Letter layer excitatory links
– Word layer → Word layer inhibitory links (immediate word competition)

As you see, the system design is not restricted to a simple feed-forward architecture. Let me point out, however, that the recurrent structure is not introduced to handle temporal processes observed in the input, or to implement some sort of memory effect; it is merely a tool for achieving meaningful activation levels in multiple passes of the simulation.

Walter van Heuven has prepared a network simulator that implements the letter and word layers of the original experiment[72]. Diagram 5-1 shows the output of the simulator for the word *wish* (X-axis=the number of passes; Y-axis=activation level; default simulation parameters):

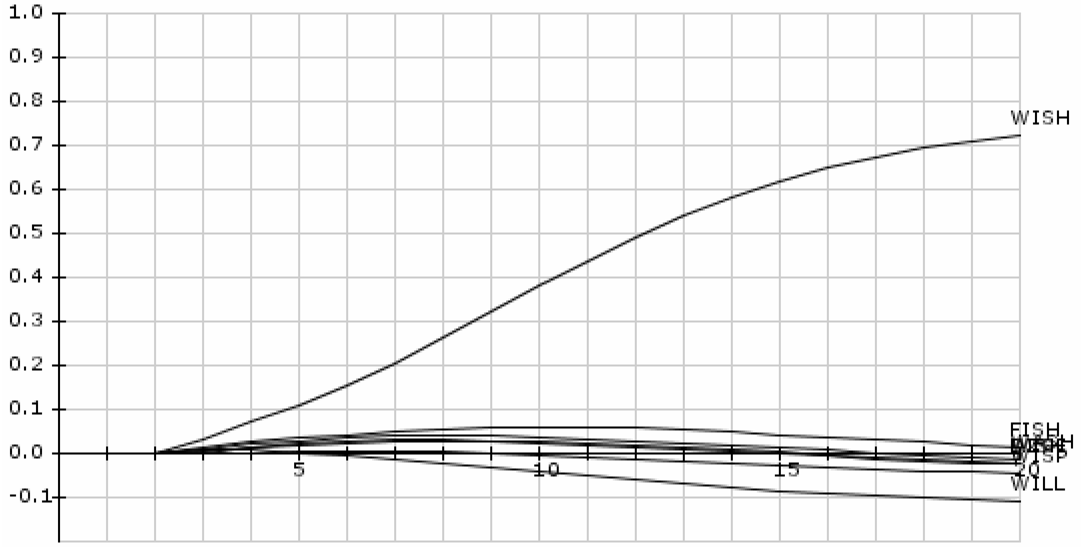


Diagram 5-1 Results for *wish* in van Heuven’s simulator

[72] The simulator is a Java applet, and is available from http://www.psychology.nottingham.ac.uk/staff/wvh/jiam/ as of May 31, 2005. Walter van Heuven uses a lexicon of 1323 English four-letter words.

The simulation results in a single dominant activated node (other activated nodes include *will, dish, fish, wash, wise* and *wisp*) after 20 passes of weight adjustment.

Diagram 5-2 depicts the result of a similar simulation for the nonsense word *wesh*:

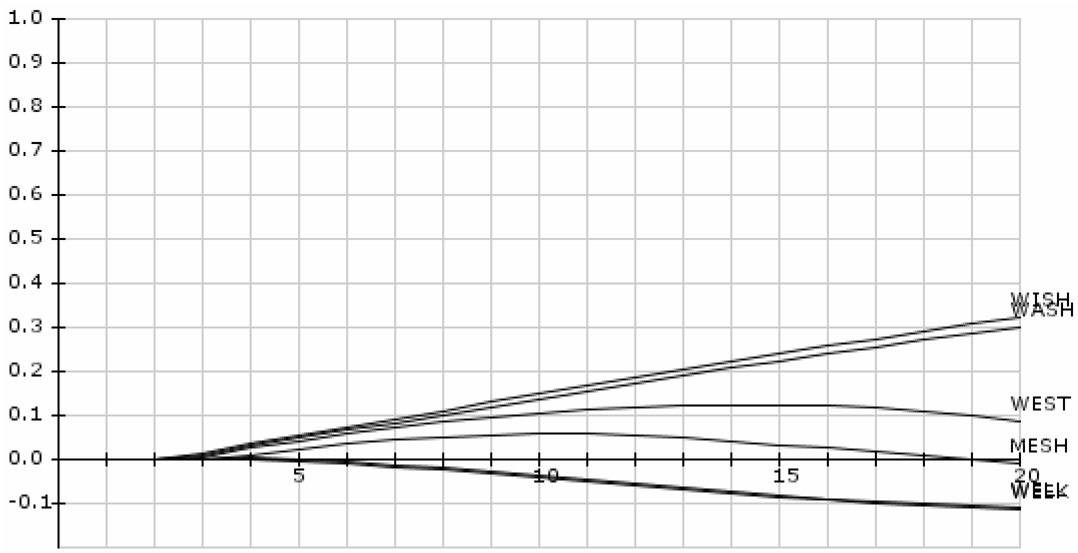


Diagram 5-2 Simulation for *wesh*

This simulation results in three activated units with positive values corresponding to *wish, wash* and *west*. This behavior can be directly exploited in linguistic applications, such as a spell-checker: the program could suggest these replacement words for the misspelled form *wesh*.

My final experiment is a simulation for the form *wa_h*. The meaning of the underscore character is “turn all units off”, and works as a joker sign in the input character string.

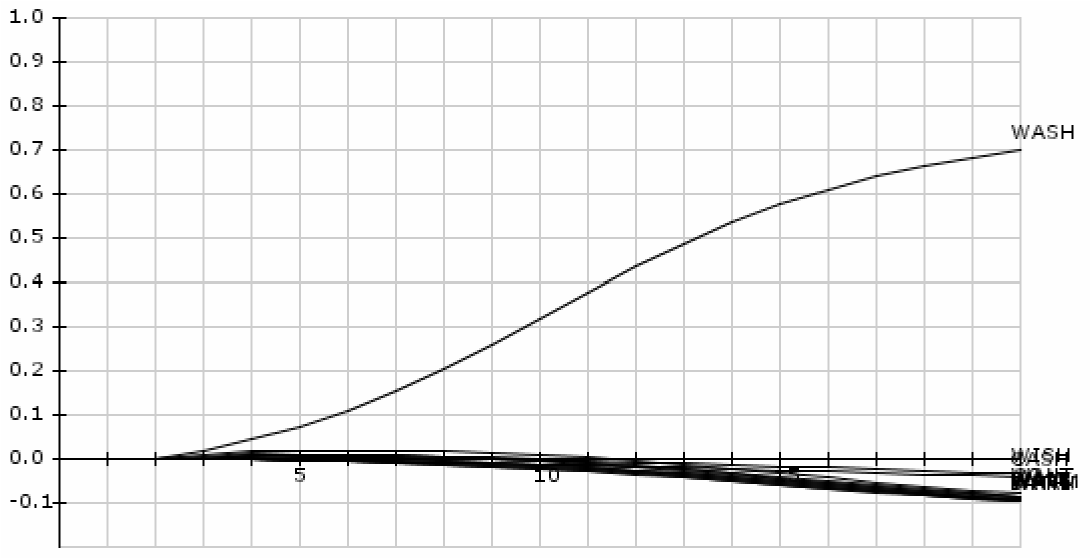


Diagram 5-3 Simulation for *wa_h*

In this simulation, the node for *wash* reached an activation level which is close to the activation level of the same node for the complete input (*wash*) in the first experiment (see diagram 5-1). It means that the system could guess at a missing letter successfully (please note that only a single word matches this input string in the lexicon). This behavior exemplifies the strong error correcting ability of neural networks: a missing input character was easily compensated for by the network. Other activated nodes included *wait, walk, wall, want, wash, wish, bath, cash, path, wake, warm, warn* and *wave*, but their activation levels were low even in the first passes and became negative during the process. The simulations presented in this section may be used to demonstrate (and perhaps to model, too) the word superiority effect: it is due to this psycholinguistic phenomenon that words are easier to recognize and to remember than random character strings (cf. Miller 1996:123).

The fixed word length of the model is a common limitation of similar systems, which affects more advanced models, too. The right representation of linguistic input has turned out to be one of the key lexicon-related problems of ANN experiments.

5.5 Elman's SRN – Finding Structure in Time

Using feed-forward connections exclusively, we can only represent time in an ANN explicitly, creating a spatial image. Meaningful implicit (temporal) representation can be created when we introduce recurrent links to the model. Although Elman (1990) gives credit to Jordan (1986) as the first to use recurrent connections *to establish some kind of memory* in an ANN, his own construction, which is now called Elman SRN (Simple Recurrent Network) after the author, has turned out to be much more influential.

An Elman SRN constitutes a hidden group and a context group (see Figure 5-1). Hidden units are connected to corresponding context group units in a one-for-one fashion via connections (one connection for each hidden-context pair of units) that have a non-trainable, fixed weight of 1 (the dotted line in Figure 5-1). All other connections are trainable. Input-to-hidden and hidden-to-output projections are usually full projections (i.e. each unit in the input group has links to all units in the hidden group and each unit in the hidden group has links to all units in the output group).

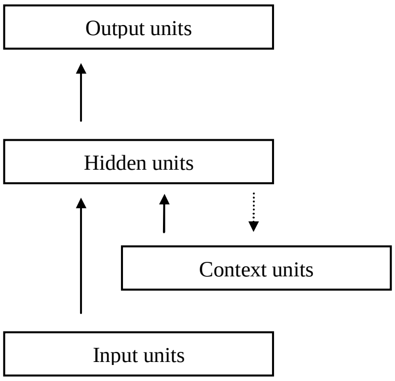


Figure 5-1 Elman's Simple Recurrent Network

Elman (1990) also presents four experiments to illustrate how his SRNs work.

The first experiment is a time-domain implementation of the XOR problem. Elman used a random sample of 3000 binary digits. The first and second digits were selected randomly and were followed by a third that was their XOR-ed value, then the whole process was repeated (resulting in sequences like this: 0 0 0 1 0 1 0 1 0 1 1 0 …). Elman used one input unit, two hidden units, two context units and one output unit in his network. In an analysis of how the network worked, Elman points out that the SRN model takes a non-traditional approach (unlike three-layer feed-forward ANNs working with input presented simultaneously):

> … one unit [in the hidden group] is highly activated when the input sequence is a series of identical elements (all 1s or 0s), whereas the other unit is highly activated when the input elements alternate. Another way of viewing this is that the network developed units which were sensitive to high and low-frequency inputs… This suggests that problems may change their nature when cast in a temporal form.
> (Elman 1990:7)

The second experiment investigates if and how SRNs can identify subsequences in a sequence of inputs. The network is constructed to accept three consonants and three vowels on its input; the representation is based on six phonological features (*consonant, vowel, interrupted, high, back, voiced*). In the experiment, an input sequence was generated in which consonants occurred randomly, but each consonant was followed by one or two vowels determined by the consonant. In practice, the following substrings appeared in the sequence: *ba, dii* and *guu*. 6 input units, 20 hidden units, 20 context units and 6 output units were used, and the task was to predict the next input. The SRN was able to learn to predict the vowels with great accuracy, since they depend on the previous character (consonant or vowel), and the error is high when the SRN had to predict the next consonant[73]. This is exactly what we expect, since consonants are selected randomly.

The third experiment features a network that takes sentences (that follow basic aspects of English syntax) as input, one letter at a time, without any kind of segmentation (i.e. breaks between words or sentences). Letters are represented by five binary digits (e.g. 00000=a, 00001=b, 00010=c, 00011=d, etc.), i.e. by a simple index that shows the place of the letter in the alphabet[74]. The input sequence itself was generated by a program that operated on a vocabulary of 15 words and generated 200 sentences altogether, which resulted in a stream of 1270 words, and ultimately, 4963 letters. The network uses 5 input units, 20 hidden units, 20 context units and 5 output units trained to predict the next letter in the sentence. The results are very similar to the results of the previous experiment: the error plot shows that word-initial letters are frequently misidentified, and the accuracy of letter prediction grows for the 2nd, 3rd etc. letters in the word. The following summary is by the author: "The simulation makes the simple point that there is information in the signal which could serve as a cue as to the boundaries of linguistic units which must be learned, and it

[73] To visualize the performance of the network, Elman used error graphs going down to the level of individual feature bits of the input pattern.

[74] An orthogonal input vector (e.g. 00001=a, 00010=b, 00100=c, 01000=d, etc.) is preferred by most authors, but non-orthogonal input does not seem to have caused any problems in this experiment.

demonstrates the ability of simple recurrent networks to extract this information" (Elman 1990:13).

The goal of Elman's last experiment is to show that an SRN can discover word classes from word order information. 10000 sentences containing 27354 word tokens were generated using 15 sentence templates. The templates consisted of non-terminal symbols only (e.g. S → *noun-hum verb-eat noun-food*). 13 non-terminals were used in these templates, each of them were rewritten to one or two words (e.g. *noun-hum* → man, *noun-hum* → woman, *verb-eat* → eat)[75]. A vocabulary of 29 words was used. Each word (terminal symbol) was represented by a 31-bit orthogonal vector (e.g. 0000000000000000000000000000010 = woman; the placement of the digit 1 in the vector was incidental). The 27354 word tokens corresponding to the 10000 sentences generated by a program were concatenated without breaks between words or sentences. The input and output groups used 31 units, the hidden and the context layers contained 150 units each. The network was trained to predict the 31-bit vector corresponding to the next word in the sentence, but consider the following comments, too:

> Successors cannot be predicted with absolute certainty. … Nonetheless, although the prediction cannot be error-free, it is also true that word order is not random. For any given sequence of words there are a limited number of possible successors. Under these circumstances, the network should learn the expected frequency of occurrence of each of the possible successor words; it should then activate the output nodes proportional to these expected frequencies.
> (Elman 1990:16)

To assess this behavior, Elman did not test the network for actual successor words, but for sets of potential successors. Units belonging to each word that could occur in that position had to have a non-zero value (this is the reason for using orthogonal input/output), and their activation levels had to be proportional to their probability of appearance in that position.

According to Elman's report, this network worked as expected: it extracted generalizations based on co-occurrence statistics. We must keep in mind that the source of this co-occurrence statistics lies in the sentence-templates: in fact, the network is expected to learn the generalizations that are present in the templates in the form of non-terminal symbols. In other words, generalizations corresponding to *noun-hum, verb-eat, noun-food* etc. are expected to appear somewhere in the network. Elman analyzed the hidden group patterns using hierarchical clustering analysis (Elman 1990:17) to see if these pre-coded "lexical classes" appeared in the hidden-group patternings. The hidden units in Elman's experiment developed a distributed representation (p. 21), and due to the nature of the task, this representation is linguistically interpretable. Figure 5-2 shows a hierarchical cluster diagram of hidden unit activations during the experiment. Nouns and verbs seem easy to tell apart, and two major categories in both the noun set and the verb set are well observable: animate vs. inanimate nouns and transitive vs. intransitive verbs.

Elman introduced a new bit pattern (corresponding to the nonsense word "zog", although the word itself did not appear in the network, it was replaced by its binary representation). No training was done for the new word, but a testing set was generated in which the bit pattern for "man" was replaced by the new pattern (p. 19). A new node appeared in the hierarchical cluster diagram that was inside the group "humans". In this sample run, the network showed an outstanding generalization capacity; please note that the ANN had to rely on contextual clues, no other source of information was available[76].

Elman points out that the literature is divided on the question whether lexical access is influenced by context (p. 19). He calls our attention, however, to Tabossi (1988) and Schwanenflugel and Shoben (1985) who argue that word *classes* are predictable on the basis of context. This is what we see in Elman's experiment, too.

[75] Elman did not use the Chomskian rewrite formalism in his paper; he presented the categories and templates in tabulated format.

[76] It is like "trying to learn a language by listening to the radio" (Elman 1990:19, the metaphor is by J. McClelland).

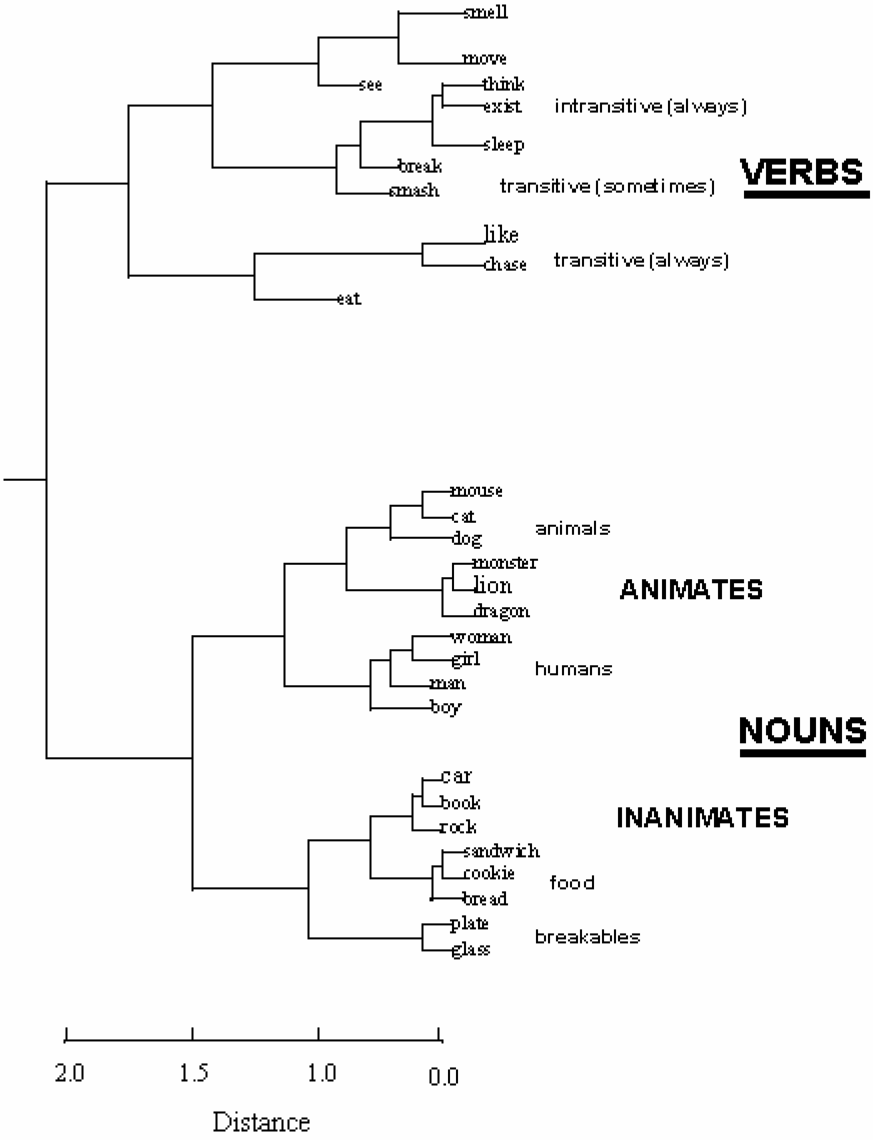


Figure 5-2 Hierarchical cluster diagram (Elman 1990:18)

Elman refers to the recurrent hidden layer - context layer structure as a type of *memory*. He argues that SRN memory "is neither passive nor a separate subsystem" (Elman 1990:24). When acquiring the right connection weights during training, this structure memorizes information to reduce overall error. There is no guaranteed duration, however, for which the information is kept (except for a one-step memory realized by the context layer). Elman points out that the analog hidden and context units in the SRN construction are theoretically capable of providing infinite memory (p. 21). The researcher should in fact be cautious not to use an "oversized" SRN, because it leads to memorizing the entire training set[77] while losing any kind of "willingness" to generalize and predict appropriate output for novel input.

The memory potential of SRNs was criticized shortly after Elman's publication of his model. The point made by Stockle (1990) and Mayberry and Miikkulainen (1999) are similar: while the SRN is sensitive to "sequential" dependencies present in the input, "long-term" dependencies will not be realized. We will see in section 5.8 how Mayberry and Miikkulainen (1999) try to remedy this problem.

5.6 Assumptions

In this section, I would like to point out that the connectionist literature (not unlike other linguistic fields) resorts to certain methodological *assumptions*. I will use Elman (1990) to illustrate my points.

Firstly, some architectural features of actual ANNs, including group dimensions, often seem arbitrary. For instance, Elman does not explain why he used a 150-unit hidden layer in the word-classes experiment (cf. section 5.5). It is assumed in the ANN literature that the author does everything he or she can to select the best construction based on experience in the field and through trials and errors. Some people even say, somewhat ironically, that this aspect of ANN modeling is more like an art than a science.

Secondly, it is customary to examine the results of a single simulation. Notice that Elman used randomized input in his experiments, but only one set of results (from a single sample run) was analyzed. Many authors follow this tradition, assuming that their systems are robust enough to give reproducible results. Although some scholars have opted for averaging results over multiple simulations (e.g. Mayberry and Miikkulainen 1999), it is in fact very exciting to see that while permuted input examples often lead to widely varying error figures during the initial phase of training, the error settles at a level that is consistent across multiple runs.

Lastly, it is generally accepted that linguistically relevant conclusions *can* be drawn from data coming from a corpus of *generated language*. It is customary to use language generators to prepare training input for ANN models for the following reasons:

– Data sparseness: ANN systems need a massive amount of data to set network weights appropriately. While huge corpora are available (for some well-researched languages at

[77] When it is coupled with a high number of training passes (provided that the task is suitable for SRN processing).

least), researchers still encounter the data sparseness problem: corpus data contain a huge variety of phenomena that act as free variables. Corpus sentences will only cover a tiny fraction of all the possible combinations. Under these circumstances, a network is not usually able to reach the primary objective of accurately approximating output for input data it has never seen. The morphological richness of some languages may further complicate the problem. Oravecz and Dienes (2002) point out that tagging highly inflective and agglutinative languages is a task which is more difficult than tagging morphologically less rich languages due to the problem of data sparseness:

> While the number of tags for non-agglutinative languages is generally between 50 and 200, systems for agglutinative and highly inflective languages use tagsets of cardinality with a magnitude higher… This entails that – in the case of an *n*gram statistical model – we have to estimate 10^n times more parameters, which would need bigger training corpora for these languages. Contrary to the needs, however, the amount of available annotated linguistic resources for these languages is much smaller than for well-researched languages, such as English.
> (Oravecz and Dienes 2002:280)

What we see is not a special weakness of ANN-based systems, however: statistical methods face the same problem[78].

– Elman (1991) argues that connectionist systems may benefit from staged input, which facilitates incremental training. Staged input is easy to prepare using language generators but hardly possible to extract from corpora. In Elman (1991), which is an updated version of the word-classes experiment of Elman (1990), the author used four input sets: 1) simple sentences only, 2) 25% complex sentences (they contained “who” plus an embedded clause) / 75% simple sentences, 3) 50%-50% complex and simple, 4) 75%-25% complex and simple. The training process started with the first corpus and ended with the fourth. Elman (1993:3) claims that the idea of incremental learning harmonizes with research according to which natural language learning and development co-occur (cf. Newport 1988, 1990). He also highlights the fact that ANNs are most sensitive during the early stages of learning (p. 16), and huge changes are difficult or impossible to make later by the commonly used learning mechanisms. Elman (1993) also describes simulations that support the idea that staged input is actually useful. He also argues, however, that there are cases in which staged input is

[78] Although the level of sensitivity to data sparseness may not be the same for statistical methods.

not required (referring to Harris 1991, for instance). In fact, the idea of staged input is not without criticism. Rohde and Plaut (1997) label Elman's suggestion "unnecessary" and even "counter-intuitive" (p. 1) as it imposes unnecessary and mostly unnatural limitations on the training data. They also carried out a set of SRN-based simulations on the basis of which they reject the idea of incremental training concluding that "starting with simplified inputs allows the network to develop inefficient representations which must be restructured to handle new syntactic complexity" (p. 661).

– Automatic annotation: it may well be very stimulating to know that the system we are developing or working with uses multi-million word corpora, but let us not forget that we need *annotated* corpora to train our systems. While corpus design criteria ensure that the corpus we use is a representative natural language sample of the language we investigate, annotation labels are usually generated by computational systems (rule-based or statistical), which leads to the question if we can treat *annotated* corpora as natural language samples. I would like to hypothesize that there is a gradience in quality between manually annotated corpus and unedited automatically annotated corpus, and the quality of annotated corpora is a function of the precision and recall of the automatic annotating method as well as the editing procedure carried out by linguists. Moreover, I doubt that less-than-exceptional-quality annotated corpora can be trusted as authentic samples of natural language.

While language-generation is preferred to using annotated corpora in ANN systems, it is important to consider possibilities to narrow the quality gap between pseudo-natural corpus data and pseudo-natural generated data. Rohde made efforts to make the training corpus used in his research (Rohde 2002) closely resemble natural language samples. He created a program called SLG (Simple Language Generator, Rohde 1999b) that generated sentences in "stochastic context-free languages", and also added the features (annotation labels) required for the training process. In his experiment, English-like production probabilities were extracted for the structures and words he decided to use from the Penn Treebank (p. 109). In this way, the independent sentences generated by SLG modeled English sentences as closely as possible (given the restricted vocabulary and the limited number of grammatical structures). Completely corpus-based ANN training is rare, but not unprecedented: for instance, Hammerton (2001) reports on an ANN system capable of clause identification - a shared task of the CoNNL 2001 (Conference on Computational Natural Language Learning) workshop. The goal of this CoNNL task was to develop machine-learning methods that recognize the clause segmentation of the test data. Training and testing data were taken from the Wall Street Journal (WSJ) part of the Penn Treebank. Hammerton was the only contributor to use an ANN-based system, and unfortunately, his system turned out to produce by far the worst precision and recall values in the competition. Neither later CoNNL tasks (all of them were corpus based), nor the Senseval competitions attracted connectionist solutions.

My own ANN system described in chapter 6 is trained to carry out FrameNet-style frame and frame element assignment. The first major system automating frame semantics is a recent study by Gildea and Jurafsky (2002). Later, the Senseval 3 task of automatic role labeling resulted in five submissions (Baldewein et al. 2004, Bejan et al. 2004, Moldovan et al. 2004, Ngai et al. 2004, Thompson, Patwardhan and Arnold 2004). These systems used hand-labeled FrameNet data as training input. Testing was also done using the FrameNet descriptions. Most of the Senseval 3 contestants implemented SVM (Support Vector Machines) algorithms. The average precision of the role-labeling systems was 0.803, the average recall was 0.757 (Litkowski 2004:3). We must keep in mind, however, that FrameNet gives frame-element specifications for a relatively small number of frames only. Systems suitable for frame and frame element identification in ordinary “unrestricted” corpora are not yet available, which also means that corpora annotated for FrameNet frames and frame elements have not been published.

5.7 Kohonen’s self-organizing maps

Let us now return to our overview of major ANN models. The approach presented in this section has great potentials in automatic input categorization. As will be shown in chapter 6, after applying some necessary extensions, we can use self-organizing maps to represent linguistic input, too.

Although the idea of *self-organizing feature maps* dates back to the 1970s (von der Malsburg 1973), it was not until the publication of Kohonen (1981a, 1981b, 1981c, 1981d and 1984) that self-organizing maps (SOMs, as we know them today) received wide acceptance. The following is a brief introduction; details supported by a mathematical apparatus can be found in Kohonen (1984).

In the SOM model, neurons are arranged in a two-dimensional sheet (the “map”), and an input layer is also present with each neuron connected to all neurons of the map. Initial connection weights are random. During an *unsupervised learning* process, map units become sensitive to various input group patterns. Input patterns are presented one after the other, and in each step, the weights of the “winning” unit (the most activated node for the given input pattern) and those of the direct neighbors of the winning unit are updated slightly. Remember that the learning is unsupervised, i.e. targets are not specified along with the input. Similar input patterns are mapped close to each other in the map: pattern classification is done automatically by the network.

The self-organizing map is a biologically motivated topology-preserving structure. Its biological motivation is exemplified by the following observation: “neighboring groups of cells in the retina project on to neighboring groups of cells in the thalamus, which in turn project on to neighboring regions of the visual cortex” (Barreto and Araújo 2001:2). In more general terms, “the spatial arrangement of the receptors in the peripheral sensory organs, such as the retina and the skin, is preserved in point-to-point or topographic connections in the sensory pathways throughout the central nervous system” (ibid.). The topology-preserving nature of the SOM is modeled after this functionality.

Barreto and Araújo also comment on the learning process exhibited by self-organizing maps:

> During learning … those neurons that are topologically close in the array … will activate each other (cooperate) to learn something from the same input x. This will result in a local relaxation or smoothing effect on the weight vectors of neurons in this neighborhood, which in continued learning over time leads to *global ordering*, meaning that the resulting map preserves the topology of the input samples in the sense that adjecent patterns are mapped onto adjecent regions of the map.
>
> (Barreto and Araújo 2001:4)

The SOM implementation in Lens is the following. The units in the Kohonen map will get a DISTANCE input function and a KOHONEN output function. The following is from the manual of Lens:

> The DISTANCE input function computes the input to the unit as the squared Euclidian distance between the incoming weight vector and the corresponding vector of input unit activations. If the pattern of activations on the input units is similar to the pattern of link weights from those units into a map unit, that map unit will have a small input.
>
> (Rohde 2000)

The KOHONEN output function is the following: *output = 1 - (input / max)*. As you see, the smaller the input, the larger the output. The output is in the [0;1] range.

It is up to the user, however, to select and change the size of the *neighborhood* that cooperates in the map in representing the input. Initially, a large neighborhood value[79] ensures that a large part of the map gets activated for any particular input pattern, which is necessary, since learning (i.e. weight-adjustment) will only affect the connections of activated units. As the training progresses, the user must decrease the size of the neighborhood, which usually stops "just a bit over" 1. As a result, only a small cross-shaped region gets activated (one unit in the "center" and four neighboring units) for any input pattern. Due to the analog nature of the units in the resulting cross, the center of the activation pattern is not the unit in the middle of the cross (when the four neighboring elements do not have the same activation level). It also means that an infinite number of input patterns can be represented by a relatively small map. This finding motivated my inclusion of a SOM-based input interface in my experiments (cf. chapter 6).

The original SOM model is developed for recognizing *spatial* input organization only, whereas a typical linguistic use involves the processing of sequences of input patterns, i.e. spatiotemporal organization. This leads us to the next section that shows how the SOM model can be extended to account for temporal features of the input.

5.8 Temporal pattern recognition in self-organizing maps

Representing time in self-organizing maps is a challenge that many researchers have tried to meet recently. Barreto and Araújo (2001) give the best overview of these attempts, listing and introducing 28 models[80] that have been implemented for various (mostly non-linguistic) purposes, ranging from image segmentation to the classification of EEG signals.

One of the few models created for linguistic purposes is the SARDNET (Sequential Activation Retention and Decay NETwork) model, which was introduced in James and Miikkulainen (1995) as an enhanced self-organizing map. The algorithm is the following:

```
INITIALIZATION: Clear all map nodes to zero.
MAIN LOOP: While not end of sequence
    1. Find unactivated weight vector that best matches the input.
    2. Assign 1.0 activation to that unit.
    3. Adjust weight vectors of the nodes in the neighborhood.
    4. Exclude the winning unit from subsequent competition.
```

[79] Barreto and Araújo (2001) recommend that the neighborhood value should start at more than half the diameter of the map (p. 4).

5. Decrement activation values for all other active nodes.

RESULT: Sequence representation = activated nodes ordered by activation values.

(James and Miikkulainen 1995)

As you see, winner units are computed for each input step, and their activation levels are decreased in each later step. Previous winners are excluded from further competition, since re-activating these units would result in loss of information. The authors recommend that the neighborhood radius be gradually set to zero, which means that a single unit must represent a single element in the temporal sequence. Training a SARDNET is otherwise very similar to training a SOM.

The authors tested their model in a task involving the classification of word forms according to phonetic features of their constituent phonemes. The distinguishing features that appeared on the input units were the following: place, manner, sound, chromacity and sonority[81]. Each feature was expressed by a real number, so the input stream consisted of feature vectors with five real numbers. Figure 5-3 shows a SARDNET activation map corresponding to the word “misplacement” generated in their experiment.

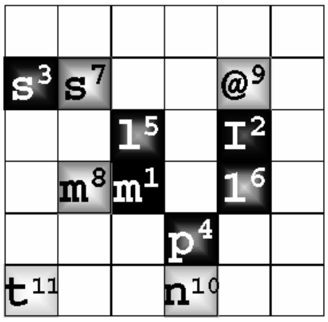


Figure 5-3 SARDNET map for the word “misplacement” (from James and Miikkulainen 1995:570).

SARDNET representations are remarkably dense, and the model offers high accuracy, too. The network presented in James and Miikkulainen (1995) was able to represent 1592 unique words (using a series of phonetic features as input) with 97.7% accuracy on a 16-

[80] Additional models have also been constructed, e.g. the Recursive SOM and the recurrent SOM models. For details on these systems, see Voegtlin and Dominey (2001).

[81] James and Miikkulainen used two feature vectors for representing the initial and final states of *diphthongs*.

unit map. Using 16 units for representing so many words is appealing, especially when compared to solutions using orthogonal input vectors[82].

Mayberry and Miikkulainen (1999) present a parser trained to carry out shift-reduce parsing. This parsing technique uses a stack to store all stages of the parsing process. The *shift* operation puts a new element onto the stack: the training target corresponds to the current input (auto-association). A *reduce* step transforms one or more of the top elements of the stack into one new element. In this experiment, the network was trained to output the top element of the stack in a compressed recursive auto-associative memory (RAAM) format; the training targets for the reduce operations were partial parse result[83]. While the shift-operation is straightforward, since the actual input element must appear on the top of the stack, the reduce operation requires the processing of long-term dependencies. The authors created different ANN setups, and assessed their performance:

a) FFN: a simple feed-forward network (with no memory effect in the form of recurrency or SARDNET)
b) SRN: Simple Recurrent Network (Elman 1990)
c) SARDNET: Sequential Activation Retention and Decay NETwork (James and Miikkulainen 1995)
d) NARX: Nonlinear Auto-Regressive model with eXogenous inputs (Chen at al. 1990): previous *outputs* are fed back into a context layer, they act as “delays”, and the whole construction is trained via Backpropagation through Time (Rumelhart, McClelland and the PDP Research Group 1986). Computational capabilities of NARX networks are equivalent to fully connected recurrent networks as well as to Turing machines according to Siegelmann et al. (1997).
e) SARDSRN: SARDNET + SRN
f) SARDNARX: SARDNET + NARX

For the experiment, the authors generated 436 sentences using a very limited number of rewrite rules and a vocabulary of 12 terminal symbols (including period signaling the end of the sentence).

Figure 5-4 is taken from Mayberry and Miikkulainen (1999), and it shows the performance of the models during the parsing task in terms of “average mismatches” (see the article for further details). The labels narx 0, narx 3 and narx 6 denote NARX-only networks with 0, 3 and 6 delays, respectively. The labels 20, 40, 60 and 80 on the X-axis

[82] We need a large, 1592-unit orthogonal input vector for the same task.

denote the size of the dataset used during training, e.g. 20 means that only 20% of the 436 sentences generated for the experiment was actually used (testing was performed using the remaining sentences).

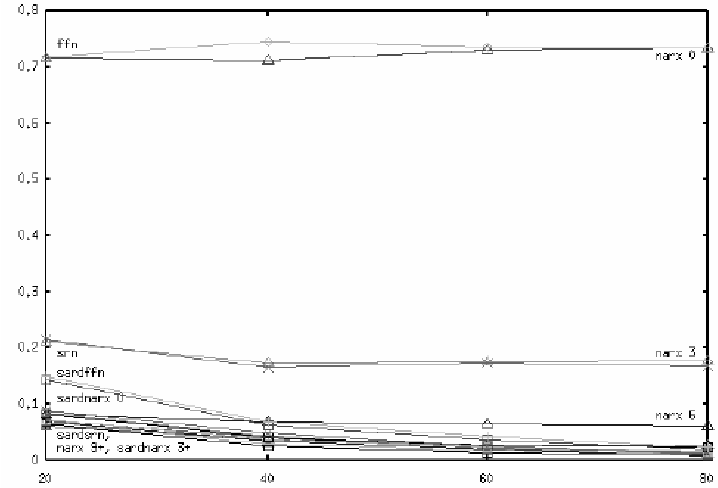


Figure 5-4 Parsing performance observed in Mayberry and Miikkulainen (1999)

We can see in Figure 5-4 that the feed-forward network and the equivalent NARX network (“narx 0”) exhibited a performance inferior to other models. The SRN and the “narx 3” models work fine, but the SARDNET model without recurrency (hence the label “sardffn”) performs even better. Combinations of SARDNET and recurrent structures (NARX or SRN) further improve the performance, but only with a negligible margin. The bottom line is that in this task, which requires memory for long-term dependencies, SARDNET performs better than SRN, while it is more flexible than NARX for which the system designer has to decide the number of delays that are hard-wired into the network.

As far as the criticism of SARDNET is concerned, Carpinteiro (1999) calls our attention to a property exhibited by SARDNET that prevents us from using it as a general solution for temporal pattern recognition. Since an activated unit has to be excluded from processing later input patterns, “identical or similar sub-sequences inserted into different points of the large sequence will never hold identical or similar representations in the map” (Carpinteiro 1999:209). This deficiency limits SARDNET’s suitability to the task presented in Carpinteiro’s article, which is the recognition of recurring patterns (the theme) in a musical piece. When using SARNET for classifying word forms, a prefix that matches a substring in the stem, or partial reduplication may create confusion, for instance.

[83] The RAAM output needs pre-training to be able to decode hidden layer representations into partial parse results.

I would like to add that the need for disabling previous winners might become problematic in a noisy environment, too. A single error may cause multiple errors, if a unit that is needed later gets activated due to the noise and becomes excluded from later analysis. While I am not hypothesizing that the use of SARDNET is impossible in the presence of imperfect input, we must consider that a single error or one noisy channel (feature vector position) may only have a tiny effect on the output map when using the static SOM model, but this is not the case here. In general, one of the advantages of neural networks with distributed representations is that they exhibit manageable performance degradation in the presence of noise or damaged input, and this feature is not necessarily available in the SARDNET model.

Self-organizing maps (the original SOM or its variants) may be used to categorize input in an unsupervised fashion, which is very useful at many levels of linguistic description, too. The properties of SARDNET make it a great candidate for receiving sequential input. James and Miikkulainen (1995) store a relatively high number of words in a fairly small SARDNET: the phonetic forms of the words that are being stored (through developing a map representation) constitute the input for the network. Mayberry and Miikkulainen (1999) find SARDNET sophisticated enough to carry out syntactic parsing, but note that they use a very limited lexicon of 12 terminal symbols (4 closed-class words, 7 open-class words and the end-of-the-sentence marker) represented by a unique ID and they even added part of speech information. In my experiment, which is described in the next chapter, semantic parsing is carried out using a multi-level recurrent structure augmented with an input interface that takes phonetic forms of words (with the option of combining them into larger chunks of speech) using a custom-designed SOM variant.

6. FORM AND MEANING IN AN ARTIFICIAL NEURAL NETWORK: A TWINMAP-DRIVEN FRAME-RELATIVE FRAME-ELEMENT RECOGNIZER NETWORK

This chapter reports on an ANN system that features a unique design that marries up a non-localist, self-organizing input interface taking syllabified phonetic transcriptions as input with a recurrent structure that carries out semantic parsing. The system is trained on a large and highly redundant set of generated sentences, which is reflected in the extremely high recall and precision figures that are achieved by the network. It will be illustrated that a meaningful linguistic task can be carried out without implementing a self-contained, separated lexical component in the model and inputting “raw” word forms and multi-word expressions with no lemmatization, disambiguation or annotation (except for the recognition target labels).

6.1 System design criteria

The network model was designed to meet the following criteria:

- The system should be able to handle an infinite number of input elements.
- The system should be able to take complex expressions (idioms, compounds or phrases) as input elements.
- No explicit morphological or syntactic information is inputted, but input elements should not be atomic, and the system should have access to the internal structure of input elements.
- The system should not be restricted to processing isolated sentences.

These design criteria have been met with constructions that are unique to this research.

I selected a semantic parsing task for this experiment: FrameNet-style Frame and Frame Element labels are used as recognition targets[84]. The compilation of the original FrameNet database has involved thorough lexicographic research of lexical units, and a labor-intensive procedure of assembling frames using corpus evidence (analyzing British National Corpus sentences containing the lexical units in question). The editors have included many frames that are considered important on the basis of their projected frequency, and they have annotated a large number of sentences using frame element labels. While annotated sentences are clearly enough to illustrate frames and frame elements, per-sentence frame element prediction seems infeasible to me, and I have opted for analyzing sentence sequences instead[85]. This approach also makes it possible to predict Frame Elements in a wider context of other Frames and Frame Elements. Note that existing systems predict Frame Elements for single sentences only (Gildea 2002, Baldewein et al. 2004, Bejan et al. 2004, Moldovan et al. 2004, Ngai et al. 2004, Thompson, Patwardhan and Arnold 2004)[86]. Please also note that these systems use non-connectionist techniques.

For the present experiment, I selected nine FrameNet frames and I also came up with frame elements for two new frames, because FrameNet lacked the highly specialized frames I needed to describe computer crime[87]. We must be cautious with the new frames, however, since the process of their definition has lacked the laborious research methods characterizing the FrameNet project (involving corpus-based balancing of frames and lexical units, and defining lexical units in terms of valence patterns)[88].

The corpus of input sentences is compiled in such a way that frame (FR) and a frame element (FE) labels are assigned to each input word form or multi-word expression. Elements that take no FR/FE labels get “all 0” target labels, which is different from the “not applicable” label attached exclusively to dummy input data (post-sentential gaps, as described below). Misidentifying the “all 0” label increases the error just like misidentifying other labels. The system takes fully segmented text with input element segmentation (mainly word forms and phrases) as well as sentence segmentation. The form *crack* has been introduced into the word stock[89] with the following senses: 1) *crack*: to get into a computer system illegally, 2) the adjective *crack* meaning “excellent”, “fantastic”, and 3) *crack*: illegal drug. To support the inclusion of these three senses, the following three situations[90] have been selected for writing the sentence templates:

– Computer crime
– Drug trade and consumption
– Communication

The following pool of frames and frame elements were used for FR/FE annotation:

[84] Phrase Type and Grammatical Function recognition is not a design goal, and many other features of the FrameNet project remain unimplemented, too.

[85] These sequences are limited in length.

[86] Moreover, a pre-selected frame and a pre-selected target word must be specified for the analysis.

[87] In general, the present version of FrameNet seems to lack highly specialized frames.

[88] I would like to point out, however, that as far as this experiment is concerned, giving full descriptions of the new frames, frame elements and related lexical units is not essential, since the model would work just fine with ill-formed frames and a bogus set of frame elements, too.

[89] Identity of written forms is not an issue in this experiment.

[90] The term “situation” is introduced and used in an intuitive fashion to refer to a collection of frames that are associated in the sentence sequence templates.

	Frame (FR)	*Origin*	*Frame elements (FE)* (FET = frame-evoking target)
01	Cause harm	FrameNet	00 FET, 01 Agent, 02 Victim, 03 Body part (puts victim if present to a lower layer), 04 Cause
02	Damaging	FrameNet	00 FET, 01 Agent, 02 Patient
03	Committing crime	FrameNet	00 FET, 01 Perpetrator, 02 Crime
04	Rewards and punishments	FrameNet	00 FET, 01 Agent, 02 Evaluee, 03 Response action, 04 Reason
05	Commerce	FrameNet	00 FET, 01 Seller, 02 Buyer, 03 Goods
06	Ingestion	FrameNet	00 FET, 01 Ingestor, 02 Ingestible
07	Communication manner	FrameNet	00 FET, 01 Message, 02 Speaker, 03 Topic
08	Expertise	FrameNet	00 FET, 01 Protagonist, 02 Role or Skill
09	Intoxicant	FrameNet	00 Intoxicant
10	Attack a computer	New	00 FET, 01 Agent, 02 Patient, 03 Attack
11	Protect a computer	New	00 FET, 01 Agent, 02 Patient system

Frame-evoking targets, which also receive frame element labels, evoke frames in a given sentence. In this experiment, they are mostly attached to verbs in the sentence templates. Frames were not exclusive to a single situation; the distribution of frames across the three situations is described in section 6.5.

6.2 System overview and architecture

The full system is a unique combination of a *self-organizing* lexical interface and a *recurrent* frame/frame element recognizer network. This latter part operates as a multi-layer simple recurrent network, which is modeled after the SRN idea (Elman 1990), with important modifications.

When selecting the input interface, I considered SARDNET (cf. section 5.8) as an option. The dense representations inherent to that model is usually considered an advantage, but notice that representing multi-word expressions in SARDNET is not straightforward since initial words interfere with upcoming words in the input string preventing the map from developing the same pattern as invoked by isolated words. Therefore, I decided to come up with a proprietary input interface (the "twinmap", see below) that can be trained to represent a number of words but the resulting patterns can be combined into multi-word chunks of speech during the simulation.

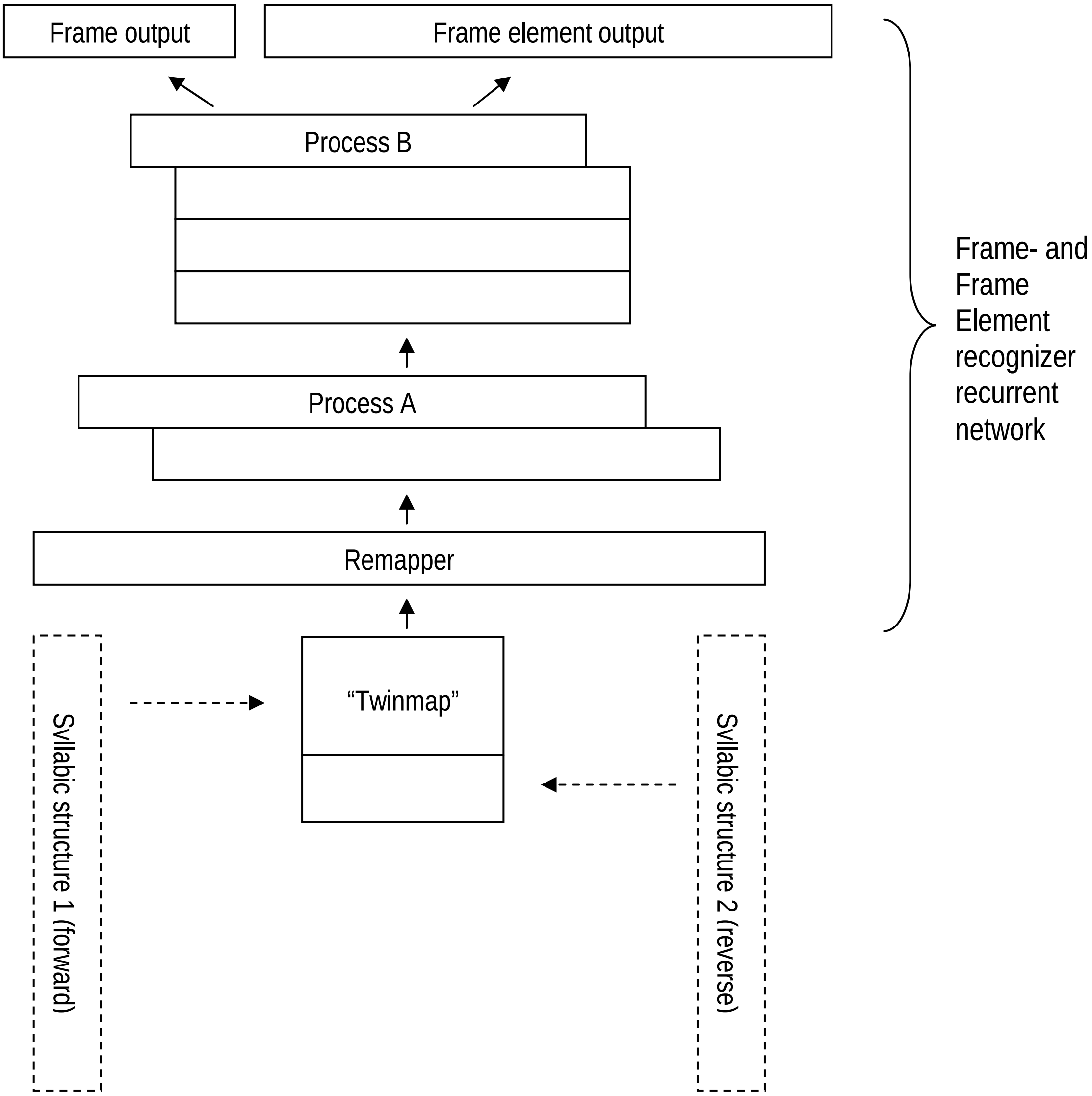

Frame output
Frame element output
Process B
Process A
Remapper
"Twinmap"
Syllabic structure 1 (forward)
Syllabic structure 2 (reverse)
Frame- and Frame Element recognizer recurrent network

Figure 6-1 General architecture

The input interface used in this experiment contains two self-organizing maps (cf. section 5.7, hence the name "twinmap"). It is driven by two "syllabic structure" groups, which are only used during pre-training, when the twinmap is trained to accommodate

20693 word forms selected from the CELEX database (Burnage 1990)[91]. The pre-training is based on two sets of feature vectors (one for each part of the map) pre-computed from CELEX phonetic transcriptions. The "syllabic structure" groups use 435 neurons each. When pre-training is done, the twinmap representations for the 296 different word forms that are used in the frame-recognition experiment are extracted. Details about planning and implementing the twinmap are in section 6.3. Figure A-1 in Appendix A depicts a single step of the training process.

It is only after all word forms that appear in the frame/frame element recognition task are extracted can the recurrent system above the twinmap be trained. The "syllabic structure" groups are not in use at that time: word forms and multi-word expressions making up the training corpus are put in the "twinmap" group as actual SOM activation patterns. When an input element contains multiple words, the twinmap patterns of all constituent words are combined into a single twinmap and are processed as a single event (see section 6.3.2 for details).

Directly above the twinmap layer, a remapper layer (60 logistic units, full projection) is inserted to facilitate pre-categorizing and remapping input into a representation which is then sent to the first recurrent construction ("Process A"). This first recurrent construction is forced to lose information after the last word form of each sentence of the incoming sentence sequences. The second recurrent construction ("Process B") is unaffected by this procedure. In Figure 6-1, the hidden layer has the label "Process B", and the three context groups are depicted below the hidden layer. The recurrent construction is described in more detail in section 6.4.

The system has two trained output groups. The smaller is for frame recognition: each recognizable frame is represented by a single unit (11 units altogether). The larger output group is for frame element recognition: 31 units are used to represent frame elements. Since frame elements are relative to frames, the "patient" frame element in frame X and the "patient" FE in frame Y are realized by different units. It also means that the inclusion of a frame-output group is redundant in the sense that we can identify the right frame element without a separate frame group, too, but in this way, frame and frame element recognition accuracy can be assessed independently[92].

[91] Details of the selection process are given in section 6.3. In the context of this experiment, the term *word form* refers to transcribed phonetic forms taken from the CELEX database, and the term *text* is used to refer to a sequence of sentences created by the language generator.

[92] The present system is trained using both output groups, and this architectural feature may (slightly) influence the performance of the model, too.

During frame and frame element recognition, the network receives twinmap representations of word forms and multi-word expressions, and processes each in a single *event*. An *example* is a set of related events; in this simulation, an example is a sequence of up to three sentences containing up to 25 events. The desired target values for the output groups are made available for each event. At the end of each example, a weight-adjusting error backpropagation phase is executed during training running from the end of the example to the beginning. The backpropagation algorithm used in this experiment is called *simple recurrent backprop through time* (SRBPTT, cf. Rohde 2002), and is offered by Lens for simulations using multi-event examples solving difficult temporal tasks. While Elman's original SRN backpropagates error after each *event* (training the network to recognize the frame and frame element based on whatever information is stored in the context groups of the network), the SRBPTT approach makes it possible to collect error information through the whole *example* and adjusts the weights accordingly. As Rohde (2002) points out, disadvantages of this method include the lack of biological motivation and increased resource needs, since it requires that the error values should be stored for each event. Otherwise, SRBPTT networks in Lens are very similar to Simple Recurrent Networks since information flows from the input to the output groups in a single step, and activation propagates from group to group determined by the group order (sequential updating).

Two training methodologies are combined in this experiment to reach fast but reliable results. Initially, error is computed after each example, as described above (examples are fed into the network in random order). Then, several passes of *batch learning* are carried out. In batch mode, the simulator collects error derivatives for a batch of examples (all examples in this experiment) before adjusting weights. Due to the large number of examples utilized here, batch learning is only used for fine-tuning the network.

Finally, let me briefly describe the software environment in which the simulations were prepared and carried out. As mentioned earlier, ANN components were trained and tested in *Lens* (Rohde 1999a). Additional programming, which I carried out, included the development of

– ToolBook scripts and C code for CELEX data extraction and the generation of twinmap training files,
– TCL scripts controlling Lens for training the two parts of the twinmap,
– TCL scripts for extracting the twinmap representations of the word forms that are used in the recurrent frame/frame element recognition task,

- ToolBook scripts and some C code responsible for creating actual input files containing twinmap input and frame/frame element targets for training/testing the recurrent network in Lens (sentence-sequence templates and the list of non-terminal symbols were stored as ToolBook objects),
- TCL scripts controlling Lens for training the recurrent network, and
- C code that examined Lens test-run dumps and assessed the performance of the recurrent network in terms of recall and precision.

All simulations were carried out on a Pentium 4 PC with 512MB RAM running Windows, Asymetrix ToolBook, gcc and the Cygwin version of Lens. Screenshots of Lens simulations are in Appendix A.

6.3 The "twinmap" input interface

The input interface of the network described in the present chapter was trained to represent a large number of word forms. The training features were *syllables* taken from the phonetic transcriptions of CELEX database entries[93]. As a preparation for the data extraction process, I carried out some preliminary measurements on the database, which are described in the following section.

6.3.1 Preliminary experiments

In the phonological part of the CELEX database, 64675 unique transcriptions were found corresponding to one or more single-word orthographic forms (including inflected and derived forms). A surprisingly high number of different syllables (11095 syllable types) were identified during the process. I also examined the lemmatized version of the database, which contained 34929 unique transcriptions and 6764 syllable types. Please notice that the number of different syllables is considerably higher in the unlemmatized part while the difference in the number of *morphemes* is supposed to be negligible (only a few suffix types are missing in the lemmatized part). It is due to the fact that morpheme and syllable boundaries do not match; nevertheless, a 60% (4331 new syllable types) increase in the number of syllable types resulting from inflectional affixation is quite surprising.

Before CELEX data extraction, I also examined the length of the word forms present in the database in terms of syllables. This step was necessary for me to determine the dimensions of certain data structures during programming the data extractor, but it is also interesting because it shows the number of active input units that are responsible for training the self-organizing input interface in the initial phase of the simulation. The results were the following (duplicate forms and multi-word entries have been excluded from this analysis, too):

Number of syllables	*Frequency*
1	19%
2	38%
3	26%
4	12%
5	4%
More	1%

Finally, I trained a 20x20 twinmap (12x20 units for the forward representation and 8x20 units for the reverse representation) to accommodate all (64675) phonetic transcriptions. The training process was identical to training the 16x16 twinmap used in the frame/frame element recognizer experiment (as detailed in the next section). When training was done, I extracted the SOM representations for all words, and compared them. My original plan was to carry out a hierarchical clustering of words and check if morphological paradigms appear in clusters, but the strong effect of inflection on *syllabified* transcriptions (as shown above) prevented me to come up with readily observable sets of morphologically relevant clusters. The single objective of the comparison of extracted twinmaps remained to see if completely identical map representations were developed during training. 221 different transcriptions were found to have non-unique representations, which is 0.3 per cent of the 64675 words. This introduces a negligible noise that should be easily compensated by the neural network that takes the twinmap input.

6.3.2 The twinmap for the frame/frame element recognizer

The input twinmap for the frame/frame element recognizer experiment is a combination of a 10x16 and a 6x16 self-organizing map. Since I had no access to computational resources

[93] These transcriptions came syllabified in CELEX.

that would have made it possible for me to train SOMs for 11095 input features, I had to reduce the number of allowed syllables. I created a tool that kept the most frequent syllables in addition to those that are required by the word forms present in the text corpus generated for frame/frame element label training and testing. The number of word forms that were covered by this reduced set of syllables was still high (20693).

During twinmap training[94], the following procedure was repeated for every word form of the CELEX database that was covered by the syllable set kept in the syllable-reduction phase. The larger half-map (forward representation) was trained first. The unit in "syllabic input group 1" that corresponded to the syllable type appearing as the first syllable was set to an activation level of 1. Units corresponding to the second, third etc. syllables were set to decreasing, non-zero values[95]. This weighting procedure was introduced as a way of representing the temporal sequence of syllables: in this way, the map is able to tell apart word forms containing permuted (but otherwise identical) syllable sequences or subsequences.

This input representation has to face two problems. Firstly, each syllable type can only be activated once, and the map loses its sensitivity to further tokens of the same syllable type. Note, however, that there is a second map (the lower part of the twinmap) that works with the reverse syllabic representation of the word, which means that each syllable type can occur twice without loss of information. In English, the chance of three or more occurrences of the same syllable type in the same word is low; should it occur, the model may introduce noise into the twinmap representations (depending on whether the lost syllables cause clash with other word forms).

Secondly, initial results showed that this method resulted in inadequate variation in the representation of short word forms: for monosyllabic words, for instance, the input vectors differed in one value only[96], so their SOM representations became very similar or identical. This phenomenon was pinpointed by a clustering analysis of the resulting maps. To increase the distance between SOM representations of any two word forms, I have added an "overtone" structure for each input unit: a syllable triggers its corresponding input unit and some close and distant neighbors (twelve units altogether) to make input vectors individual enough. As expected, monosyllabic word forms have profited the most from this modification.

[94] Which I also call "pre-training" since it precedes the training of the Frame/Frame Element recognizer recurrent network.

[95] Units corresponding to syllables that were *not* present in the word form were set to zero.

[96] Their overtone-less input pattern is series of 435 zeroes and a single non-zero value.

Due to the weighting process, trailing syllables in long word forms can hardly influence the resulting map representation, which may prevent the entire ANN model from using information that is present in word endings. For this and other reasons discussed above, a second map has been added with a different weighting function. When training the second (reverse) map, the unit of "syllabic input group 2" corresponding to the *last* syllable of the word form is set to 1, the last but one syllable gets 0.7, and other syllables that are present are set to significantly lower, decreasing values. An overtone structure is added here, too, for each non-zero value in the input vector. This smaller map is hypothesized to carry some morphologically relevant information: word forms with the same trailing syllables (which may or may not correspond to actual suffixes) develop similar representations in this part of the twinmap. In practice, hierarchical clustering of this smaller map showed that this size (6*16 units) resulted in a large number of word forms that have similar representations[97], and morphologically relevant correspondences were difficult to find. One of these correspondences was a group of word forms with the -ing suffix that ended up in a single cluster in hierarchical clustering.

The general system overview in section 6.2 points out that having trained the two parts of the twinmap, we extract representations for those word forms that appear in the frame/frame element recognition task. In the first step, the patterns appearing in both SOMs are combined into a single twinmap for each relevant word form. This representation should be unique enough to make it possible for the recurrent constructions to identify frame and frame-element outputs for each word form. The twinmap representation is based on the internal structure of incoming words and multi-word expressions, but in practice, this piece of information about the formal properties of the input only complements contextual information in the process. It is not possible to recover actual word forms from twinmap activation values. Figure 6-2 depicts a twinmap produced for a single word form with three syllables during pre-training. The syllable structure input vectors on both sides show three groups corresponding to the three syllables and some of the surrounding "overtones". Different gray levels show that the forward vector is weighted differently than the reverse input vector.

[97] We should keep in mind that these representations were developed for 20693 words and not just those word forms that were used in the corpus generated to train the recurrent structures.

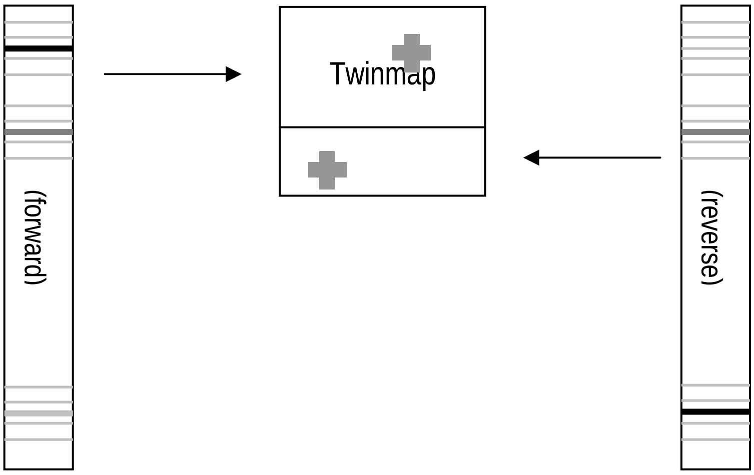


Figure 6-2 Twinmap for a single word form with three syllables

Since the recurrent structures are able to differentiate small activation-level differences in the input and delineate different input tokens accordingly, my only concern has been to eliminate the majority of completely identical twinmaps in the simulation. Out of the 296 orthographic word forms used for generating the training files for the frame recognizer network, the twinmap only developed identical representations for the homophones *buy* and *by*.

When the input is a multi-word expression (idiom, compound, phrase, etc.), all the relevant twinmaps are combined into a single map: every word form leaves its fingerprint in the twinmap as a cross in the upper part (forward representation) and a cross in the lower part (reverse representation). Please note that the same segmentation is required during recognition. A network working with these composite twinmaps may be able to exploit their compositionality automatically, when it is useful for the training process. Figure 6-3 depicts a twinmap representing the phrase “make mistakes”.

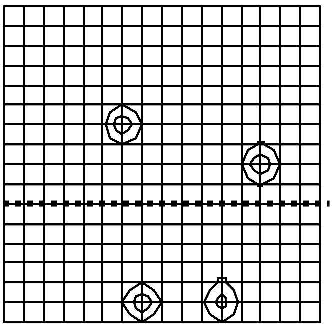


Figure 6-3 Twinmap representation for “make mistakes”

6.4 Two levels of recurrent processing

The present model is constructed in a way that twinmap processing is decoupled from actual frame/frame element recognition. When frame/frame element assignment is done in the recurrent structures, frozen twinmap activation levels corresponding to the syllable structure of one or more word forms are copied into the twinmap units that do not work as actual SOM units at that time, but as simple input to the remapper layer feeding the recurrent construction. The training and testing procedure used in this experiment is based on sequences of one, two or three sentences consisting of up to 23 twinmap inputs (word forms or multi-word expressions). An input pattern in the twinmap interface, which is accompanied by targets for the two output groups, constitutes an event, which is a discrete time step in the simulation. An *example* is a sequence of events and it corresponds to a sequence of sentences in this experiment.

The recurrent part of the system consists of two substructures ("Process A" and "Process B" in figure 6-1). The first structure ("Process A"; it contains a hidden and a context group, 2*30 units; each context unit receives exactly one connection with a frozen weight of 1.0, cf. Elman 1990) is reset after each sentence, which means that no memorization beyond the sentence level is possible here. The second structure has three Elman context groups containing 15 units each connected in a way that hidden unit activation flows to the first context group (which is depicted directly below the hidden layer), and the activation of the first context group is copied to the next group one event later. It is only during the third event that the third context group receives the original hidden group representation. The construction is guaranteed to keep information that appears in its hidden group for three events. Each context layer is connected to the hidden layer using full projections, which means that any given hidden group activation level is influenced by three sets of previous activation levels, which are stored in the context groups, as well as the current input coming from the hidden layer of the first recurrent construction. Please note that in theory, Process A (intra-sentential processing only) and Process B are both able to memorize event sequences of an infinite length as activation patterns. As a result, this architecture exhibits a three-event look-ahead and an infinite look-back capability[98] with sensitivity to sentence boundaries. The fixed look-ahead capability of three events (accepting three words, phrases or larger chunks of speech) has been adjusted to the training corpus used in this experiment, but its extension to an arbitrary length (which ought to be adjusted to human memory limitations) should not cause difficulties.

The recurrent structures are trained using a training corpus described in chapter 6.5. To illustrate how this network handles sentence sequences, let us take the following example:

> Crackers crack the system to steal data by deploying a virus. Stealing data is illegal, so crackers commit crime.

Let us suppose that this sequence is annotated for *frame* labels in the following way:

> $_{10}$[Crackers] $_{10}$[crack] $_{10}$[the computer] $_{03}$[to steal data] by deploying a virus. $_{03}$[Stealing data] is illegal, so $_{03}$[crackers] commit $_{03}$[crime].

The *frame element* annotation for this sentence is as follows (in each $[]_{xx/yy}$ *xx*=frame ID, *xx/yy*=frame element ID; frame elements are relative to frames, so the interpretation of *yy* depends on *xx*):

> [Crackers]$_{10/01}$ [crack]$_{10/00}$ [the computer]$_{10/02}$ [to steal data]$_{03/02}$ by deploying a virus. [Stealing data]$_{03/02}$ is illegal, so [crackers]$_{03/01}$ commit [crime]$_{03/00}$.

The following is a tabulated version of the above diagrams, modified to illustrate that FR/FE labels appear on the output with a delay for reasons described later. Each row of the table is a single event.

Event	Twinmap corresponding to…	Target FR (frame)	Target FE (frame element)	Reset “Process A” hidden layer
0	crackers	–	–	
1	crack	–	–	
2	the computer	–	–	
3	to steal data	10	10/01	
4	by deploying a virus	10	10/00	
5	[empty twinmap]	10	10/02	
6	[empty twinmap]	03	03/02	
7	[empty twinmap]	all 0	all 0	
8	stealing data	-	-	
9	is	-	-	Yes

[98] We will see shortly that other network features are also introduced to support this effect: a three-event gap is used between sentences to further weaken inter-sentential dependencies, and the network output is produced with a three-event delay.

10	illegal	-	-	
11	so	03	03/02	
12	crackers	all 0	all 0	
13	commit	all 0	all 0	
14	crime	all 0	all 0	
15	[empty twinmap]	03	03/01	
16	[empty twinmap]	all 0	all 0	
17	[empty twinmap]	03	03/00	

The first column contains the event index; it is for our reference only. The second column shows the word form or expression for which the relevant twinmap is retrieved (more twinmaps are combined into a single one for multi-word expressions in events 2, 3, 4 and 8, cf. section 6.3).

The third and fourth columns show the output expected to appear in the output groups (frame and frame element output). The interpretation of these target values is described in section 6.1. The value of 02 for frame, for instance, means “Damaging”, and the FE value “02/01” means “Agent” of “Damaging”. The FR target of 2 is represented by the vector [0,1,0,0,0,0,0,0,0,0,0] in the FR group, so this value is simply an index. Note that the output units are analog in both groups, they are not restricted to binary output (and not even “encouraged” to produce values close to 0.0 or 1.0), and these units can work independently of each other. The FE target vector is slightly more complicated to compute. Each FR/FE pair gets an index (e.g. 02/01=0, 02/02=1, 04/02=2, 01/01=3, etc., in the order of their appearance in the sentence sequence generating templates, all FR/FE pairs are listed in section 6.1). This index is then used to identify FR/FE combinations in the FE output vector. In the above example, the FE target 02/01 will be represented by the target vector {1,0,0,0,0,0,0,0,…,0}. The actual output of the network will be compared to the FR and FE targets during training, and any value other than the pre-designated target increases the overall error of the example. The goal of the training is to minimize overall error by adjusting connection weights in the whole system. In the optimal case, all outputs match the pre-specified targets precisely.

When “–“ is given as target in the above table, the output is irrelevant, and it will not influence the error. This is only used for [empty twinmap] inputs. These inputs appear in three-event post-sentential gaps that follow each sentence of the example. In all other cases, appropriate output is expected, including the “all 0” cases. The “all 0” case is an event for which FR and FE targets are not specified, but the network must identify this case with zero outputs in all output units: anything greater than 0 will be treated as error (please remember that all units are analog).

Notice that the output is *delayed* by three events, i.e. the target for the first word form (event 0) is only expected in event 3, and the target of the last word form (“crime”, event 14) is expected in event 17. The idea behind this procedure is to facilitate proper recognition of the first word form (an *agent* frame element in most cases in this simulation). This delay is the primary reason for introducing the 3-event forced memory in the second recurrent construction (“Process B”). The three-event post-sentential gap described above is used to weaken inter-sentential dependencies.

The last column shows that the first recurrent construction (“Process A”) is reset in event 9: in practice, a pre-event TCL procedure resets the activation of units in the hidden layer to the initial value of 0.5. This procedure is employed after each sentence of the example to prevent the first recurrent structure (“Process A”) from storing information across sentence boundaries. All hidden and context groups are automatically reset to 0.5 by the simulator before each example (i.e. before each sentence sequence).

The tabulated format of our example sequence is used in this section for illustrative purposes. A fragment of an actual Lens example file is shown in Appendix D. Example files for this experiment have been compiled by a custom-made program, which is introduced in the next section.

Please see *Figure A-2* in Appendix A, which explains a Lens screenshot showing a simulation event.

6.5 Generating training data

This section describes the process of compiling input files (Lens example files) for the ANN experiment. I have created a ToolBook application that handled sentence sequence templates with terminal and non-terminal symbols, with optional frame/frame element specifications, and generated Lens training example files with input and target specifications. It was also up to this application to handle the non-terminal sets, and store twinmap representations for all terminal symbols (word forms and multi-word expressions).

Please note that the generator program uses orthographic words to stand for twinmap representations of corresponding spoken words, but the written forms do not appear in the input files and therefore they are not accessible for the neural network. Also note that homographs resulting in ambiguities in the written form → spoken form mapping should and can be accounted for in the language generator program.

I used 38 sentence sequence templates in the simulation. The following table shows the number of sentence sequences and constituent sentences for each of the three situations I used:

Situation	*Number of sentence sequence templates*	*Number of sentence templates*
Cracking a computer	11	18
Drug trade and consumption	22	37
Communication	5	10

The full list of frames and frame elements used as annotation labels is specified in section 6.1. The following table shows the distribution of frames across the three situations I used.

Situation *Frame*	*Computer crime*	*Drug trade and consumption*	*Communication*
Cause harm		yes	
Damaging	yes		
Committing crime	yes	yes	yes
Rewards and punishments	yes	yes	yes
Commerce		yes	
Ingestion		yes	
Communication manner			yes
Expertise	yes	yes	yes
Intoxicant		yes	

Attack a computer	yes		
Protect a computer	yes		

Appendix C shows samples of the sentence sequence templates that were used for generating the training data. The following is one of the sentence sequence templates used in this experiment:

> *#PE1_1001 cracked the #comp_um_1002 * #PE1_0201 #crkgoal_past * #PE2_1101 realized that the $comp_um_1002 was cracked_1000 * #PE2_1101 #patch_past_1100 the $comp_um_1102 **

This template uses all features supported by the sentence sequence generator code:

– Terminal symbols (unmarked): character strings that are present in the CELEX database. During a pre-training procedure, these word forms are located in the database, rewritten to a series of syllables, and placed on the 16*16 twinmap based on their syllable structure. When sentence sequence generation takes place, we look up each terminal symbol in the twinmap database and replace it by the twinmap activation pattern observed for this word in the final stage of twinmap pre-training.

– Non-terminal symbols (marked by the # and $ prefixes): these symbols are rewritten to terminal symbols placed in a named *set* (“um”, “past”, etc.) which belongs to a non-terminal *superset* (e.g. “comp”, “crkgoal”). For instance, the non-terminal superset “crkgoal” consists of three non-terminal sets: “um”, “ing”, “past”. The following table shows the members of these three sets:

um (unmarked)	*ing*	*past*
steal data	stealing data	stole data
steal personal data	stealing personal data	stole personal data
steal money	stealing money	stole money
do harm	doing harm	did harm
do wrong	wrongdoing	did wrong
cause damage	causing damage	caused damage
cause loss	causing loss	caused loss
cause data loss	causing data loss	caused data loss
cause disadvantage	causing disadvantage	caused disadvantage
get secrets	getting secrets	got secrets

Non-terminals that take the *#superset_set* form are rewritten to each member of the selected set of the selected superset one by one. This process is repeated recursively for each #-prefixed non-terminal, their number is only limited by the stack used by the scripting environment (ToolBook). Non-terminals using the *$superset_set* formalism behave differently. They are rewritten to a single terminal (word form or multi-word expression) that corresponds to a #-prefixed non-terminal belonging to the same superset that have been used previously in the same sentence sequence. The template *"peter #crkgoal_past * $crkgoal_ing is illegal *"* is rewritten to:

peter stole data * stealing data is illegal *
peter stole personal data * stealing personal data is illegal *
peter stole money * stealing money is illegal *
etc.

But combinations like *"peter stole data * stealing money is illegal *"* and *"peter stole personal data * stealing money is illegal *"* are not generated.

The above table also illustrates that non-terminal symbols may contain multi-word expressions. They are not rewritten to a series of individual word forms, but replaced by a single unified twinmap[99] that represents every word in the expression. This is simply done by summing corresponding map elements and scaling the result back to the interval [0,1].

– Named entity placeholders references (#PE1 and #PE2). #PE1 and #PE2 are non-terminal symbols. A template containing at least #PE1 (or both #PE1 and #PE2) is processed in two passes (thus the number of generated sentence sequences will double). In the first pass, #PE1 is rewritten to *peter*, and #PE2, if present, is rewritten to *thomas*[100]. In the second pass, #PE1 becomes *thomas* and #PE2, if present, becomes *peter*. In this way, *thomas* and *peter* act as named-entity placeholders, preventing the network from associating fixed FR/FE labels with either *peter* or *thomas*. While it is true that these terminals will mostly get an "agent" FE label, these labels are relative to frames, so "agent" of "attack a computer" and "agent" of "protect a computer" are different and unrelated frame elements. You can think of #PE1 and #PE2 as indexed pronominals, too (he$_i$ and he$_j$). Due to the way these two non-terminals are resolved,

[99] In this way, a whole phrase (e.g. an NP) can take a single FR/FE annotation label.
[100] These are arbitrarily chosen names from the proper names listed in CELEX.

$PE1 and $PE2 are not available; you can use #PE1 and #PE2 several times in the same sequence with the same translation (*peter* / *thomas*).

- FE targets: these annotation labels have the format *_xxyy*. They can be attached to terminal or non-terminal symbols. The generator code prepares a FR group target vector and a FE group target vector, with one "1" value in each, for each event that has the *_xxyy* annotation label. Events that have no target labels will also be associated with a target vector for both FR and FE groups, with all elements set to 0 (with the exception of post-sentential dummy events). The procedure of translating FE annotation labels into FR and FE group target vectors is detailed in chapter 6.4. Notice that separate FR labels are not used since *xx* in the *_xxyy* FE label identifies the frame.
- Sentence segmentation marks ("*"). The present model requires sentence segmentation to process the input as a sequence of individual sentences rather than a long sequence of individual word forms. Sentence boundaries are translated into three-event gaps realized by dummy events with no targets. The first recurrent structure, "Process A" is also reset during this gap.

The 38 sentence sequence templates used in this experiment were rewritten to 21610 actual sequences of 1-4 sentences. The number of events with FR and FE target vectors (including forms or multi-word expressions with the "all 0" FR and FE output vectors) was 268870. The size of resulting Lens example text file (containing input twinmap specifications and target vectors for the two output groups) was approximately 64 MB. Example files were processed in Lens controlled by TCL scripts that I wrote to create, train and test the neural network.

6.6 Training and Basic Testing

Training the recurrent structure was fast, not exceeding 2 hours in duration, due to the modest size of the network. The hidden layer of the second structure ("Process B"), which is the only direct source of information for the two output groups (FR and FE), features a tiny group of 15 units. In fact, the model performs well with fewer units, too, with somewhat more fluctuating error results when comparing multiple test runs.

The network is trained to minimize output errors as measured at the two output groups. The units are analogue, and we do not expect them to show an activation level exactly[101]

[101] There are methods to encourage binary representations in the network, including the implementation of an output cost function. Lens provides us with a tool, the *polarity* command that helps to examine the

corresponding to 1.0 and 0.0. The error is measured using the following formula ("cross entropy", as specified in Rohde 2000): *t log(t/o) + (1-t) log((1-t)/(1-o)),* summed over all units of the two output groups, where *t* is the desired target and *o* is the actual output. No error is computed for events corresponding to post-sentential gaps. All other events have valid targets, either a single[102] 1.0 in each output group and 0.0 for the remaining units, or zeroes only (the "all 0" case).

The training set included 268870 events with valid targets. The network was exposed to the whole training set 21 times. First, the network was trained on-line (without batching). Then, 20 passes of fully batched training were used to fine-tune the system. This unusual procedure speeds up training and, more importantly, it seems to have decreased the level at which the error value stagnated.

Having completed the training procedure, I tested the network to get detailed error statistics. The total error was 50864 (for your reference, the same network produces an overall error around 10 million when untrained, i.e. reset to a random initial state), the error for the FR and FE groups were 24726 and 26184, respectively. The per-example (=per-sequence) error in the FR group was 1.1442, the corresponding figure for the FE group was 1.2116.

While this model has analog outputs and it would make good sense to use the fuzzy output of the network, I wrote a defuzzifying classifier to assess the performance of the system in terms of precision and recall.

Recall is the proportion of those events for which an output reaching or exceeding a pre-specified threshold is produced. Precision is the proportion of correct recognition of those events that are accounted for (in terms of recall) by the network. In this experiment, the threshold was set to 0.5. For instance, an FR *output activation* pattern (0.03, 0.11, 0.07, 0.53, 0.08, 0.99, 0.01, 0.02, 0.03, 0.04, 0.05) is understood as (0, 0, 0, 0.53, 0, 0.99, 0, 0, 0, 0, 0). Since this vector contains non-zero elements, it is treated as a valid recognition, i.e. it increases the recall value, and it influences the precision figure, too: a *target* vector of (0,0,0,0,0,1,0,0,0,0,0) matches the above output vector since the index of the output unit with the highest activation (value=0.99, index=6) corresponds to the target unit (value=1, index=6) in the target vector. Precision and recall were computed for both output groups

polarization of the network (i.e. the degree to which the output gets close to the minimum or maximum value, which is 0 and 1 for logistic units). Polarization techniques were not used in this simulation, and the network performed very well without them, too.

[102] Multi-target labeling (involving target vectors with multiple units with non-zero values) was not used in this simulation, but its implementation is technically unproblematic.

(FR and FE). F-score values (F-score= (2 * recall * precision)/(recall + precision)) were also computed to make precision/recall comparisons easier.

When tested on all examples that the ANN had been exposed to in the training phase, the network produced the following precision and recall figures:

FR recall (threshold=0.5):	98.87 %
FR precision:	97.99 %
FR F-score	***0.9843***
FE recall (threshold=0.5):	98.00 %
FE precision:	98.53 %
FE F-score	***0.9826***

The above figures show that this Artificial Neural Network is able to learn the task very well. Note that the dimensions of the network are fairly modest: a 16*16 twinmap (6*16 + 10*16 units) is the input interface which is pre-trained to accommodate 20693 word forms, and a 15-unit hidden group (belonging to "Process B") is responsible for driving a 11-unit FR and a 31-unit FE output group, while also representing more than 22 thousand different sentence sequences accounted for by 38 different templates. Also note that the network has no explicit information about the presence of the templates and the non-terminal symbols in the templates: generalization is only possible upon the detection of common input patterns scattered through the permuted input examples.

While I did not carry out a thorough investigation of the possible sources of error due to the high recognition performance, I studied network output for a number of sentences using the Unit Viewer facility in Lens, which made it possible to observe activation levels for each unit of the network for each example (i.e. sentence sequence) and each event (i.e. form or multi-word expression). This quick examination showed that the single major cause of error was the presence of conflicting templates. Consider the following two templates, for instance:

> *#drugcons_ing_0600 #tox_um_0602 is hazardous to health * people_0601 who $drugcons_um_0600 $tox_um_0900 may die **

and

> *#drugcons_ing #tox_um_0104 is hazardous to health * people_0102 who $drugcons_um_0600 $tox_um_0900 may die **

Sentence sequences generated by these templates contain conflicting frame element targets. The first non-terminal (*#drugcons*) produces events with the 06 target FR label and the 0600 FE label in the first template and "all 0" target vectors in the second template. Labels for another non-terminal (*#tox*) and those for a terminal symbol (*people*) also collide. These templates were introduced to check if multi-way recognition could have been achieved using this simple approach (with a multi-way analog classifier instead of our one-way binary defuzzifying classifier). The observation of actual activation levels in the unit viewer facility showed that the network became unstable during testing when it reached an event for which conflicting targets had been available, which was probably due to the lack of "dependable" target signal, so neither of the targets were observable in the activation levels of the output groups in many cases.

In addition to the Lens example files, I also generated corresponding text files that contained the input examples in human-readable form (i.e. text files with the orthographic word forms that had been translated into twinmaps by the generator code). These text files also made it possible to count the word types and tokens in the training corpus. Using a concordancer program, I found that the training corpus contained 296 word types and more than 340,000 word tokens, which gives a type/token ratio of 0.09%. This figure is unnaturally low, so steps were taken to narrow the gap between the training corpus and a comparably sized natural language sample in terms of type/token ratio. I kept only 5% of the training examples (1068 examples that were randomly selected; they contained 235 word types and 17173 word tokens, type/token=1%), and the remaining examples (20542 sentence sequences) were set aside for testing. Training consisted of one on-line pass and 250 full batch passes[103] then I tested the network using a) the training corpus (1068 examples), and b) all the unknown examples, with the following results:

a) Test set = training set (1068 sentence sequences)	
FR recall (threshold=0.5)	96.85 %
FR precision	99.72 %
FR F-score	***0.9826***
FE recall (threshold=0.5)	95.48 %
FE precision	99.67 %
FE F-score	***0.9753***

[103] This is the same training procedure as used above, but more batch passes were required before the error curve started to flat out and error change remained within an interval of 1%-4% when neighboring training batches were compared.

b) Test set = unknown examples (20542 sentence sequences)	
FR recall (threshold=0.5)	96.21
FR precision	99.24
FR F-score	***0.9771***
FE recall (threshold=0.5)	95.01 %
FE precision	99.31 %
FE F-score	***0.9711***

As you see, network performance remained excellent even after this drastic change in the density of the training corpus. Surprisingly, never-seen examples cause hardly any difficulties in the recognition task, which means that the network fully generalized its strengths and weaknesses to cover novel input. The 1068 training examples seem to have been sufficient for the network to make the necessary generalizations about the 38 sentence sequence templates and about the non-terminal sets (41 sets were used containing 4-40 words or multi-word expressions each).

6.7 Test experiments

The experiments described in this section were carried out with the network trained on all examples to eliminate network error resulting from the sparse training dataset.

In the first experiment, I examined whether the homophones crack$_1$ ("to get into a computer system illegally"), crack$_2$ ("excellent", "fantastic") and crack$_3$ ("illegal drug") cause problems in the frame/frame element recognition process. The twinmap corresponding to *crack* had been referred to in 21 templates. 11 of them contained the word as a terminal symbol (with the following FE targets: _0900, _0602, _1000) and the remaining 10 templates contained either the non-terminal *tox_um* (_0900, _0602, _0104, _0503) or *ace_um* (_0800): the form *crack* occurred once in each set. As you see, the network had been trained to associate six different FE targets with the same form.

For this experiment, I selected 10 templates that contained the non-terminal symbol *tox_um* or *ace_um*. By restricting the generator program to use 4 words or multi-word expressions from each non-terminal set, I generated two sets of 336 test examples. In the first set, *tox_um* was rewritten to *marijuana, cannabis, cocaine* and *heroin* while *ace_um* was replaced by *ace, proficient, superb* and *outstanding*. In the second set, *tox_um* became *marijuana, cannabis, cocaine* and *crack*, while *ace_um* was rewritten to *ace, proficient, superb* and *crack*, that is, one word in both sets was omitted and replaced by *crack.* Please note that *crack* was the member of both sets originally (*tox_um* contained 10 words, *ace_um* consisted of 12 adjectives including *crack* during training), which means that if homonymy is not an issue, the network should produce comparable results whether the test corpus contains the word *crack* or a replacement word (*heroin* in the *tox_um* set and *outstanding* in the *ace_um* set).

a) without "crack"	
FR recall (threshold=0.5)	94.42 %
FR precision	97.54 %
FR F-score	***0.9596***
FE recall (threshold=0.5)	92.52 %
FE precision	99.04 %
FE F-score	***0.9567***
b) with "crack"	
FR recall (threshold=0.5)	94.63 %
FR precision	97.63 %
FR F-score	***0.9611***
FE recall (threshold=0.5)	92.63 %
FE precision	99.29 %
FE F-score	***0.9585***

As shown by the above table, the use of multiple senses of *crack* with a high number of target FR and FE labels in the training corpus caused no recognition ambiguity in this control environment. In fact, sentences sequence groups with *crack* fared slightly better in the recognition task than those with *heroin* and *outstanding*, although these words had not been associated with other non-terminal sets (other than *tox_um* and *ace_um*, respectively), which means that they are theoretically much easier to categorize than the various uses of *crack*. In practice, the neural network seems to have had no difficulties in assigning the right network resources for disambiguating the homophones. Diagrams for visual examination of FR and FE outputs for two of the ten sentence sequence templates used in this experiment are in Appendix B. The diagrams show that the examples with the word *crack* behave very similarly to the examples without *crack*.

In the <u>second experiment</u>, I carried out measurements with two sentence sequence templates, each containing two sentences. The templates were the following:

1 crackers_1001 crack_1000 the #comp_um_1002 to #crkgoal_um by #crkmethod_ing_0302 * $crkmethod_ing_0302 is illegal so crackers_0301 commit #crime_um_0300 *

and

2 #PE1_0702 often #comm_s_0700 #tox_about_0703 * I think #PE1_0301 is a criminal and will #gpunished_um_0403 sooner or later #drugbuy_for_0404 *

I generated 500 sentence sequences for the first template (by restricting the rewrite process to only use the first five elements in the non-terminal sets). The precision was 100% at a recall rate of 100%. The test set for the second template consisted of 512 sentence sequences. Again, the precision was 100% at a recall rate of 100%.

Then I manipulated the templates.

Firstly, the second sentence was deleted from both of them. I expected no performance degradation here, since all we did was stopping the recognition of the input earlier than expected, which should not affect the performance. The results confirm the expectations: precision and recall remained 100%.

Secondly, the first sentence was deleted from the templates. Here, the situation is more problematic since the network had to find a way to skip the first sentence. In practice, precision and recall remained at the original level.

In the final step, the two sentences of each template were swapped. The change in precision and recall is shown in the following table:

	change in recognition performance	
Template 1	FR recall:	-13.4%
	FR precision:	(unchanged)
	FE recall:	-15.1%
	FE precision:	(unchanged)
Template 2	FR recall:	-2.2%
	FR precision:	-0.1%
	FE recall:	-3.7%
	FE precision:	(unchanged)

While the omission of the first or second part of the templates caused no performance degradation at all, that is the ANN was able to process the two parts of the examples as two independent sentences, swapping the two sentences did have a slight impact on the network: for some events, it produced no valid recognition. This result shows that inter-sentential contextual effects do appear in the model, which has in fact been a design goal. Please note that the precision of the network remained practically unchanged: when a valid recognition was produced, it remained precise.

In the third experiment, I deleted a single non-terminal symbol in each test phase from the same templates as used above. Practically, the generated test sequences lacked a single constituent when compared to the original training data. To elicit more information, two non-terminals were deleted in two separate experiments (case A and B) from each template. In template 1, the constituent *to #crkgoal_um* was omitted first (actual terminal elements include "to steal data", "to cause damage" and "to get secrets"). Case B involved the omission of the direct object *the #comp_um_1002* (e.g. "the computer", "the server"). In template 2, I removed the verbal predicate *#comm_s_0700* in the first part of the experiment ("whispers", "mumbles", etc.). In case B, I deleted a prepositional phrase (*#tox_about_0703*, examples include "about marijuana" and "about opium"). The results were the following:

	performance change, case A	*performance change, case B*
Template 1	FR recall: -12.3%	FR recall: -9.3%
	FR precision: -1.4%	FR precision: -5.8%
	FE recall: -16.5%	FE recall: -21%
	FE precision: -1.8%	FE precision: -2%
Template 2	FR recall: -11.8%	FR recall: -8.6%
	FR precision: -6.2%	FR precision: -6%
	FE recall: -8.6%	FE recall: -12.5%
	FE precision: -6.8%	FE precision: (unchanged)

In this experiment, the removal of a single constituent had a moderate negative effect on the recall and a slight effect on the precision of the network. Precision figures remained above 93% in all cases. The explanation is that the network had enough contextual clues to make up for the loss of one constituent (at least when it was stored as one event, as in the above examples). These clues may have included function and content words as well as multi-word expressions.

In the fourth experiment, I was interested in the performance change caused by the *replacement* of words or expressions by other forms that were originally associated with different FR/FE targets (or "all 0" targets). I worked with the following sentence sequence template:

3 *administrators_1101 prevent crackers_1001 from #crkgoal_ing_1000 by #patch_ing_1100 the #comp_um_1102 **

I replaced the non-terminal set *comp_um* (the unmarked set of the *comp* superset) by non-terminal symbols that were rewritten to:

a) unknown mix: words that had not been used at all during training the FR/FE recognition network[104]: *ablaut, ablauts, wash, washing, xerox, xeroxing, ogled,*

b) known and “situation-friendly” words and expressions: *protect, fixing, secured, spoof address, patching, protected,*

c) known but “situation-external” words and expressions: *cocaine, heroin, mescaline, trading, produce, obtain, buy opium,*

d) the plurals of the original *comp_um* set. During training, the network had been exposed to sequences generated by the following two templates:

i) *administrators #patch_um the #comp_**um** to protect $comp_**um** from attacks* *

ii) *administrators #patch_um the #comp_**plur** to protect $comp_**plur** from attacks* *

These templates generated parallel texts differing only in a noun (singular vs. plural) repeated twice in the sentence, therefore the network was expected to establish some kind of connection between the two sets.

The sequences generated for the original template in 3 are accounted for by the network with a recall of 99.8% and 99.6% for the FR and FE groups, respectively. The precision figure was 100% for both groups.

The following table shows the change in performance for the manipulated texts (test cases a-d):

	performance change	
Case A	FR recall:	-10.3%
	FR precision:	-0.2%
	FE recall:	-7.3%
	FE precision:	-0.2%
Case B	FR recall:	-8.7%
	FR precision:	(unchanged)
	FE recall:	-4.9%
	FE precision:	-1%
Case C	FR recall:	-7.5%
	FR precision:	-0.1%
	FE recall:	-5.4%
	FE precision:	-0.3%

[104] Of course, the twinmap component had been pre-trained to come up with a representation for these words, too.

Case D	FR recall:	(unchanged)
	FR precision:	(unchanged)
	FE recall:	-1%
	FE precision:	(unchanged)

The results show that replacing *computer* by *computers*, or *server* by *servers*, etc. in case *D* had a negligible effect: the ANN had indeed established some sort of connection between the two sets of forms. Cases *B* and *C* showed more change and were similar to one another: although case *B* contained situation-friendly words (related to computer crime), while case *C* contained words from another situation type (drug trade), they caused a similar performance degradation. The low *FR recall* figure in case *B* is a surprise, since some of the replacement words had been trained to recognize the same frame (frame 11, albeit a different FE[105]) as the original *#comp_um*. It is also true, however, that precision remained a perfect 100% in this case. Finally, the greatest decrease in performance was triggered by the use of unknown words that had not been used in the training process at all.

I would like to conclude that this neural network is able to compensate for missing or distorted input[106]. In the first experiment, it perfectly compensated for missing sentences, while in the second experiment, although missing constituents seem to have caused more trouble, precision remained high. The constituent replacement experiment was also handled successfully, and various levels of unexpectedness were reflected well in the recall and precision figures.

6.8 Limitations

The ANN simulation presented in this chapter has many limitations. Most importantly, the training and assessment processes are not corpus-based: we use a limited number of sentence sequence templates instead. It foreshadows a practical problem, too: an authentic training corpus[107] would be a source of considerably less dense training information, and

[105] It is not clear whether identical FR targets coupled with different FE targets cause more confusion than different FR *and* FR labels: FE and FR outputs are realized by different groups, but the hidden layer in "Process B", which is directly connected to the output groups, is not likely to contain (completely) separated pathways for driving these groups.

[106] It is a general observation that artificial neural networks are very good at compensating for noise, as mentioned in the previous chapter, too.

[107] Please note that corpora with FrameNet annotations have not yet been published. Also note that the present implementation of the twinmap interface works with phonetic transcriptions. When switching to a corpus of authentic *written* language, special attention must be paid to handle homographs to differentiate between, for instance, 'will *read*' and 'have *read*', or 'he was born to *lead*' and '*lead* is a chemical element'.

the scalability of this model in such a situation is unknown. Remember that a training corpus restricted to only 5% of the generated examples resulted in a type/token ratio of 1% (cf. section 6.6) with excellent recognition results, but it is still roughly an order of magnitude lower than the type/token ratio exhibited by comparably sized natural language samples.

As another practical consequence of switching to an authentic corpus, we would see a dramatic increase in the size of the input vocabulary, too, which might result in an increasing number of input units in the "syllabic structure" groups, which are used to pre-train the twinmap. In this situation, an input representation based on phonetic *features* of syllables (rather than representing syllable types locally) may turn out to be a better solution.

A group of potential problems is related to the partial implementation of FrameNet features in this model. FrameNet frame annotation can (and in some cases, should) be done in multiple layers. Consider, for instance, the definition for the frame Cause_harm (cited from the FN database):

> The words in this frame describe situations in which an Agent or a Cause injures a Victim. The Body_part of the Victim which is most directly affected may also be mentioned in the place of the Victim. In such cases, the Victim is often indicated as a genitive modifier of the Body_part, in which case the Victim FE is indicated on a second FE layer.
>
> (Frame Report: Cause_harm, n.d.)

A related issue is the "officially" suggested procedure of fully annotating sentences or text. For full annotation of running text, we should "one by one declare each word in a sentence a target, select a frame relative to which the new target ought to be annotated, get a new set of annotation layers (frame element, grammatical function, phrase type) and appropriate frame elements tags, and begin to annotate" (Johnson et al. 2004). This sort of multiple layering has not been used in practice by any project, and the present project is not an exception. As a result, only one particular FE structure for each text is acquired and reproduced by the network. Since PT and GF markup is not an issue in this experiment, focusing on a single reading in terms of FEs seems a reasonable compromise.

Layering is not the only FrameNet feature that is unsupported in the present version of this model: frame-to-frame relations (cf. section 4.2) are also missing. For instance, there is no connection between the *Agent* frame elements of the frames Cause_harm and

Cause_motion: in FrameNet, both frames inherit the *Agent* FE from the Intentionally_affect frame. In the present version of the frame/frame element recognizer, frames remain isolated entities. Notice that actual applications (e.g. a text retrieval system) may work fine without frame hierarchies, too, as long as a well-planned and unambiguous set of annotation labels can be selected for both annotating and querying the system.

Plans for an update version of the system also include frame feedback in the form of NARX recurrency or SARDNET self-organizing sequence-memory. This procedure and other measures may allow a similar network setup to work with texts rather than sentence sequences of limited length and content.

6.9 Overview of achievements

The recognition system presented in this chapter has illustrated that a recurrent network (even with consciously limited resources) exhibits a remarkable potential of handling frame-semantic annotation: near perfect recall and precision were measured using a highly redundant training set and a fully trained test corpus. One of the goals of the experiment was to demonstrate the potentials of the self-organizing twinmap input interface, which is introduced here as a workaround for the self-organizing map approach to represent temporal patterns while preserving all advantages of using an existing, widely implemented ANN architecture (SOM) based on unsupervised learning. The twinmap solution has turned out to be robust enough to represent a large number of different word forms; furthermore, combining several twinmaps for representing multi-word expressions has resulted in no difficulties. The present system works with unannotated input: no morphological or syntactic information is added to help the learning process. Ambiguous entries have also been used, and no noticeable performance degradation has been found. It is also important to note that the system processes sentence sequences rather than individual sentences, which is implemented by a unique combination of recurrent structures. Finally, the task itself, FrameNet-style semantic parsing, is also a novelty in the connectionist literature.

7. SUMMARY

While specifying a lexicon is sometimes treated as a follow-up to developing a new model of grammar or an application, lexicon design, that is the description of how we store and handle idiosyncratic building blocks (usually morphemes and/or words) of language, is one of the most complex problems of linguistics and by no means secondary in importance. We should consider a wide variety of theoretical questions while keeping an eye on the implementational consequences.

My thesis tackles problems that are connected in one way or another to considerations about the structure and function of the lexicon. Chapter 2 on Generative Grammar uses morphological processes to illustrate how meaning and form are manipulated in various Chomskyan frameworks, and to show the fluctuating nature of the lexicons these frameworks utilize. Chapter 3 discusses the problem of homonymy, polysemy and other aspects of Lexical Semantics to show that meaning is not easy to grasp. Since a lexicon should store (and perhaps work with) meanings, these are relevant considerations.

Chapter 4 is a report on two major relational lexical databases, WordNet and FrameNet. They are likely to influence NLP system design for a utilitarian reason: building a database that represents the idiosyncrasies of a language is a labor-intensive task, so existing major data sources that are freely available are being integrated into a large number of NLP systems. WordNet's sense-relations and the frame-relations in FrameNet are exciting new approaches to representing meaning in a database that is designed to contain lexical information. MindNet, which is also discussed in this chapter, illustrates that it is possible to store lexical information that seems compatible with the important advancements of lexical semantics (more specifically, Cruse's conception of word meaning). The discussion of a spreading activation approach to storing word meaning is also included.

In linguistics, Generative Grammar is the flagship of *symbol manipulation*, which is an appealing approach to Cognitive Science (cf. Pléh 1998:79-98). A different approach, *connectionism*, is exploited in neural network simulations. While the literature of Generative Grammar is abundant in references to lexicon-related considerations (chapter 2 can only survey a fraction of the relevant literature), chapters 5 and 6 are devoted to a much less researched topic: representing linguistic input and the emergence of lexis in connectionist models. Having seen the level of variation in the function of the lexicon depending on which model of Generative Grammar we choose, we should not be surprised that the lexis of neural network models differ significantly from the lexicons accompanying generative models.

Quasi-productive derivational processes in Morphology (see section 2.2.2), as well as *partially predictable* systematic polysemy in Lexical Semantics (see section 3.4) do not seem to have been very useful for symbol manipulation. *Continuous* natural language phenomena, such as sense-spectra and the polysemy-monosemy continuum (see section 3.4) are also difficult to grasp. These kinds of information may appear in ANNs, controlled by a rich network of analog units, without any conscious external effort. It is not to say, however, that only neural network models can approach (at least some of) these phenomena, as shown by the example of MindNet (section 4.3).

The necessity of even the most fundamental NLP tasks has already been relativized in the literature. The following statement is due to Rohde: “symbolic models of sentence processing have mainly been applied to the task of parsing, which only by supposition is necessary for natural language processing” (Rohde 2002:8). Word sense disambiguation is also treated as a well-encapsulated NLP task, encouraging us to base models on disambiguated input (i.e. words belonging to a single morphological, syntactic, semantic class), while the existence of disambiguated natural language input may not be justified. Sections 4.3 illustrates that non-connectionist authors have also been experimenting with undisambiguated solutions: Dolan, Vanderwende and Richardson (2000), who decided not to use disambiguated input in the MindNet project, conclude that “the traditional view of WSD as involving the assignment of one or more discrete senses to each word in the input string” cannot be used to implement broad-coverage NLP systems (p. 5), and “like humans, machines cannot be expected to perform reliably on a task that is incorrectly formulated” (ibid.; this quotation also refers to their argument according to which even humans do not seem to excel in carrying out “traditional” word-sense disambiguation tasks). Also note, however, that connectionism is not incompatible with these tasks by nature, and ANN models can be constructed to solve regular NLP tasks, too, including syntactic parsing (cf. section 5.8) and word-sense disambiguation (see, for instance Véronis and Ide 1990).

The experiment presented in chapter 6 has been designed to show that a well-composed linguistic task can be performed without having to carry out some traditional tasks of computational linguistics. Notice that the simulation described in chapter 6 uses *unannotated input* consisting of actual word forms fetched from the training corpus. No explicit morphological or syntactic information was included to help the learning process. Even homonyms were used (various senses of *crack*), but performance degradation was not detected. Lexical ambiguity caused no problems for Rohde's sentence comprehension and production model, either (Rohde 2002:172-178). Drawing on the findings of these experiments, I would like to argue that, even in a more realistic task of natural language processing, a connectionist model may perform "disambiguation" on its own using unannotated input while selecting the right information sources present in the context automatically.

Connectionist models are capable of storing meaning and form in a unified structure (cf. section 5.5). We see that information is being distributed or integrated into categories automatically, as required by the training process, and due to the paucity of initial information about any kind of systematicity in the training examples, regularities and idiosyncrasies are stored together before the network has any chance to tell them apart. The result is a **dissolved lexicon – distributed lexis** approach to handling and storing syntagmatic and paradigmatic (and other) relations between words or other elements of linguistic input.

Linguists should seriously consider connectionism when they find themselves compiling complex systems that contain a rich network of lexical, syntactic, morphological and other types of information and a host of relations – whose automatic discovery becomes a necessity. The virtually unrestricted disintegration and reintegration of information in a neural network makes processing non-transparent in the symbol-manipulation sense, in which symbols ought to have relevant and identifiable functions. While steps have been taken to examine the activation patterns of hidden layers of artificial neural networks (as illustrated in section 5.5, too), notice that we only observe, but do not modify the functioning of the network in this way. In other words, we assume full control of working with existing results of linguistic description when constructing a symbol-manipulation system, while researchers using connectionist methods use more and more results of linguistic description to prepare training and testing data and to construct, train and test networks that can *learn* to analyze or produce more and more aspects of language. I would like to emphasize, however, that the present thesis does not try to prove that connectionism offers superior solutions to those aspects of lexicon design that are presented here as problem issues; neural network modeling has its own pitfalls and problems to solve. I only aim to add a noteworthy, highly neglected perspective to lexicon design: a non-traditional, connectionist approach to representing certain aspects of linguistic knowledge.

BIBLIOGRAPHY

Adámek, J. (2002). *Neural networks controlling prosody of Czech language*. Unpublished MA thesis, Charles University, Prague.

Allen, J., Hunnicutt, S., & Klatt, D. (1987). *From text to speech – the MITalk system*. Cambridge, MA: MIT Press.

Antworth, E. (1990). *PC-KIMMO: a two-level processor for morphological analysis. Occasional Publications in Academic Computing No. 16*. Dallas, TX: Summer Institute of Linguistics.

Agirre E., Atserias J., Padró L., & Rigau G. (2000). Combining supervised and unsupervised lexical knowledge methods for word sense disambiguation. In M. Palmer and A. Kilgarrif (Eds.), *Computers and the humanities: Special double issue on Senseval*.

Anderson, S. R. (1982). Where is Morphology? *Linguistic Inquiry, 13(4).*

Andor, J., Hollósy, B., Laczkó, T., & Pelyvás, P. (Eds). (1998). *The diversity of linguistic description: Studies in Linguistics in honour of Béla Korponay*. Debrecen: University of Debrecen, Institute of English and American Studies.

Apresjan, J. D. (1973). Regular polysemy. *Linguistics, 142,* 5-32.

Aronoff, M. (1976). *Word formation in Generative Grammar*. Cambridge, MA: MIT Press.

Baayen, H., & Lieber, R. (1991). Productivity and English derivation: A corpus based study. *Linguistics, 29,* 801-843.

Baker, M. (1988). *Incorporation: A theory of grammatical function changing*. Chicago: University of Chicago Press.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. *Proceedings of the COLING-ACL,* Montreal, Canada.

Baker, C. F., & Ruppenhofer, J. (2002). FrameNet's frames vs. Levin's verb classes. In J. Larson, & M. Paster (Eds.), *Proceedings of the 28th annual meeting of the Berkeley Linguistics Society,* 27-38.

Baldewein, U., Erk, K., Padó, S., & Prescher, D. (2004). Semantic role labelling with similarity-based generalization using EM-based clustering. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text,* 64-68, Barcelona.

Banerjee, S., & Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *Proceeding of the Fourth International Conference on Computational Linguistics and Intelligent Text Processing (CICLING-02).* Mexico City.

Banerjee, S., & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence IJCAI-2003,* Acapulco, Mexico.

Barreto, G. A., & Araújo, A. F. R. (2001). Time in self-organizing maps: An overview of models. *International Journal of Computer Research, Special Issue on Neural Networks: Past, Present and Future, 10(2),* 139-179.

Basili, R., DellaRocca, M., & Pazienza, M. T. (1997). Contextual word sense tuning and disambiguation. *Applied Artificial Intelligence, 11(3),* 235-262.

Basu, S., Mooney, R. J., Pasupuleti, K., & Ghosh, J. (2001). Using lexical knowledge to evaluate the novelty of rules mined from text. *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources,* Pittsburgh.

Bejan, C. A., Moschitti, A., Morărescu, P., Nicolae, G., & Harabagiu, S. (2004). Semantic parsing based on FrameNet. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text,* 73-76, Barcelona.

Berlin B., & Kay, P. (1969). *Basic color terms: Their universality and evolution*. Berkley: University of California Press.

Black, A., Ritchie, G., Pulman, S., & Russell, G. (1987). Formalisms for morphographemic description. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, 11-18.

Bloomfield, L. (1933). *Language*. London: George Allen and Unwin.

Borgulya, I. (1998). *Neurális hálók és fuzzy-rendszerek*. Budapest, Pécs: Dialóg Campus Kiadó.

Botha, R. P. (1968). *The Function of the Lexicon in Transformational Generative Grammar*. The Hague: Mouton.

Botha, R. P. (1984). *Morphological mechanisms*. Oxford: Pergamon Press.

Bresnan, J. (1982). The Passive in Lexical Theory. In J. Bresnan (Ed.) *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.

Bullinaria, J. A. (1995). Modelling lexical decision: Who needs a lexicon? In J. G. Keating (Ed.), *Neural Computing Research and Applications III (Proceedings of the Fifth Irish Neural Networks Confrence)*, 62-69. Maynooth, Ireland.

Burnage, G. (1990). *CELEX - A guide for users*. Nijmegen: Centre for Lexical Information, University of Nijmegen.

Carpinteiro, O. A. S. (1999). A hierarchical self-organizing map model for sequence recognition. *Neural Processing Letters archive, 9(3),* 209-220.

Carter, D. (1995). Rapid development of morphological descriptions for full language processing systems. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, 202-209.

CELEX English database (Release E25) [On-line]. (1993). Available: Nijmegen: Centre for Lexical Information [Producer and Distributor].

Chen, J. N., & Chang, J. S. (1998). Topical clustering of MRD senses based on information retrieval techniques. *Computational Linguistics, 24(1).*

Chen, S., Billings, S., & Grant, P. (1990). Non-linear system identification using neural networks. *International Journal of Control*, 1191-1214.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Chomsky, N. (1965). *Aspects of the theory of Syntax*. Cambridge, MA: MIT Press.

Chomsky, N. (1970). Remarks on nominalization. In R. Jacobs, & P. Rosenbaum (Eds.), *Readings in English Transformational Grammar*. Waltham, MA: Gin and Company.

Chomsky, N. (1986). *Barriers*. Cambridge, MA: MIT Press.

Chomsky, N. (1986b). *Knowledge of language: Its nature, origin, and use.* New York: Praeger.

Cole, R., Mariani, J., Uszkoreit, H., Varile, G. B., Zaenen, A., Zampolli, A., & Zue, V. (1998). *Survey of the state of the art in human language technology*. Cambridge: Cambridge University Press.

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge: Cambridge University Press.

Cruse, D. A. (2000). *Meaning in language.* Oxford: Oxford University Press.

Deane, P. D. (1987). *Semantic theory and the problem of polysemy*. PhD dissertation. Chicago: University of Chicago.

Dolan, W., Vanderwende, L., & Richardson, S. (2000). Polysemy in a Broad-Coverage Natural Language Processing System. In Y. Ravin, & C. Leacock (Eds.), *Polysemy: Theoretical and computational approaches*, New York: Oxford University Press, 178-204. Retrieved 30 July, 2005 from http://research.microsoft.com/research/pubs/view.aspx?pubid=1039

Dorr, B. J., & Jones, D. A. (1996). Acquisition of semantic lexicons: Using word sense disambiguation to improve precision. *Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14,* 179-211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7,* 195–225.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition, 48,* 71–99.

Fellbaum, C. (Ed.) (1998). *WordNet: An electronic lexical database.* Cambridge and London: MIT Press.

Fellbaum, C., Grabowski, J., & Landes, S. (1995). *Matching words to senses in WordNet: Naive vs. expert differentiation of senses. Technical report.* Mannheim: University of Mannheim.

Fellbaum, C., Palmer, M., Dang, H. T., Delfs, L., & Wolf, S. (2001). Manual and automatic semantic annotation with WordNet. *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh.

Fillmore, C. J. (1968). The case for case. In E. Bach, & R. T. Harms (Eds.), *Universals in linguistic theory*, 1-88. New York: Holt, Rinehart, and Winston.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, 280,* 20-32.

Fillmore, C. J., & Atkins, B. T. S. (1994). Starting where the dictionaries stop: The challenge for computational lexicography, In B. T. S. Atkins, & A. Zampolli (Eds.), *Computational Approaches to the Lexicon*, 349-393. Oxford: Oxford University Press.

Fillmore, C. J., Johnson, C. R., & Petruck, M. R. L. (2003). Background to Framenet. *International Journal of Lexicography, 16(3),* 235-250.

Fillmore, C. J., Wooters, C., & Baker, C. F. (2001). Building a large lexical databank which provides deep semantics. *Proceedings of the Pacific Asian Conference on Language, Information and Computation*, Hong Kong.

Frame report: Cause_harm (n.d.). Retrieved 30 May, 2005, from http://framenet.icsi.berkeley.edu/index.php?option=com_wrapper&Itemid=118&frame=Cause_harm&

FrameNet Frequently Asked Questions (n.d.). Retrieved May 30, 2005, from http://framenet.icsi.berkeley.edu/ , FAQs.

Garside, R., Leech, G., & Sampson, G. (1987). *Computational analysis of English: A corpus-based approach*. London: Longman.

Geeraerts, D. Prototype Semantics. (1994). In R. E. Asher (Ed.), *The encyclopedia of language and linguistics, Volume 6*. Oxford: Pergamon Press.

Gildea, D., & Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics, 28(3)*, 245-288.

Halle, M. (1973). Prelogomena to a theory of word formation. *Lingusitic Inquiry, 4(1)*.

Hammerton, J. (2001). Clause identification with Long Short-Term Memory. In *Proceedings of CoNLL-2001*, Toulouse, France.

Harris, C. (1991). *Parallel distributed processing models and metaphors for language and development*. Ph.D. dissertation, University of California, San Diego.

Hendrick, R. (1995). Morphosyntax. In G. Webelhuth (Ed.), *Government and Binding Theory and the Minimalist Program*. Oxford and Cambridge: Basil Blackwell Limited.

Hirst, G., & St-Onge, D. (1998). Lexical Chains as representations of context for the detection and correction of malapropisms. In Fellbaum (1998), 305-332.

Hollósy, B. (1997). *Introduction to computer-aided text handling and analysis.* Debrecen: Lajos Kossuth University.

Hollósy, B., & Kiss-Gulyás, J. (Eds.). (2002). *Studies in Linguistics, 6.* Debrecen: Institute of English and American Studies.

Hunyadi, L., Gósy, M., & Olaszy, G. (Eds.) (1995). *Studies in Applied Linguistics, Volume 2*. Debrecen: Department of General and Applied Linguistics, Lajos Kossuth University.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics, 24,* 1-40.

Jackendoff, R. (1975). Morphological and semantic regularities in the lexicon. *Language, 51(3),* 639-671.

James, D. L., & Miikkulainen, R. (1995). SARDNET: A self-organizing feature map for sequences. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Processing Systems, Volume 7,* 557-584, Cambidge, MA: MIT Press.

Jiang, J., & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan.

Johnson, C. R., Petruck, M. R. L., Baker, C. F., Ellsworth, M., Ruppenhofer, J., & Fillmore, C. J. (2004). FrameNet: Theory and practice. Retrieved December 2, 2004, from http://www.icsi.berkeley.edu/framenet/book/book.html

Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach. Institute for Cognitive Science Report 8604.* University of California, San Diego.

Karlsson, F., & Karttunen, L. (1998). Sub-sentential processing. In Cole et al. (1998).

Karov, Y., & Edelman, S. (1996). Learning similarity-based word sense disambiguation from sparse data. *Proceedings of the 4th Workshop on Very Large Corpora*, Copenhagen.

Karttunen, L. (2001). *A short history of two-level morphology.* Retrieved August 26, 2005, from http://www.ling.helsinki.fi/~koskenni/esslli-2001-karttunen/

Kashket, M. B. (1986). Parsing a free-word order language: Warlpiri. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, New York.

Kilgarriff, A. (1992). *Polysemy*. PhD thesis, University of Sussex.

Kilgarriff, A. (1997). *I don’t believe in word senses. ITRI-97-12.* University of Brighton.

Kilgarriff, A., & Gazdar, G. (1995). Polysemous relations. In F. Palmer (Ed.), *Grammar and Meaning: Essays in Honour of Sir John Lyons*, 1-25. Cambridge: Cambridge University Press.

Kohonen, T. (1981a). Automatic formation of topological maps of patterns in a self-organizing system. In Oja, E., & Simula, O. (Eds.), *Proceedings of 2SCIA, Scand. Conference on Image Analysis*, Helsinki, 214-220.

Kohonen, T. (1981b). *Construction of similarity diagrams for phonemes by a self-organizing algorithm. Technical Report TKK-F-A463.* Helsinki University of Technology, Espoo.

Kohonen, T. (1981c). *Hierarchical ordering of vectorial data in a self-organizing process. Report TKK-F-A461*. Helsinki University of Technology, Espoo.

Kohonen, T. (1981d). *Self-organized formation of generalized topological maps of observations in a physical system. Report TKK-F-A450*. Helsinki University of Technology, Espoo.

Kohonen, T. (1984). *Self-Organization and Associative Memory*. Berlin and New York: Springer-Verlag.

Kohonen, T., Mäkisara, K., & Saramäki, T. (1984). Phonotopic maps - insightful representation of phonological features for speech recognition. *Proceedings of 7ICPR, International Conference on Pattern Recognition*, 182-185. Los Alamitos, CA: IEEE Computer Soc. Press.

Korponay, B., & Hollósy, B. (Eds.) (2000). *Studies in Linguistics, 5*. Debrecen: Institute of English and American Studies.

Koskenniemi, K. (1983). *Two-level Morphology: A general computational model for word form recognition and production.* Helsinki: University of Helsinki, Department of General Linguistics.

Koskenniemi, K. (1990). Finite-state parsing and disambiguation. In *Proceedings of the the 13th International Conference on Computational Linguistics (COLING 90)*, 229–232.

Kövecses, Z. (2002). *Metaphor: A practical introduction*. Oxford: Oxford University Press.

Kövecses, Z., Tóth, M., & Babarci, B. (1996). *A picture dictionary of English idioms. Volume 1: Emotions.* Budapest: ELTE Eötvös Kiadó.

Kwong, Oi Yee. (2001). Word sense disambiguation with an integrated lexical resource. *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh.

Laczkó, T. (1998). Some remarks on argument structure inheritance by derived nominals. In J. Andor, B. Hollósy, T. Laczkó, & P. Pelyvás (Eds.), *The diversity of linguistic description: Studies in Linguistics in honour of Béla Korponay*. Debrecen: University of Debrecen, Institute of English and American Studies.

Laczkó, T. (2000). Variations on the Theme of Thematic Roles. In B. Korponay, & B. Hollósy (Eds.), *Studies in Linguistics, 4.* Debrecen: Institute of English and American Studies.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. Chicago: University of Chicago Press.

Langacker, R. W. (1987). *Foundations of Cognitive Grammar. Volume 1: Theoretical Prerequisites*. Stanford: Stanford University Press.

Langacker, R. W. (1991). *Foundations of Cognitive Grammar. Volume 2: Practical Applications*. Stanford: Stanford University Press.

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In Fellbaum (1998), 265-283.

Levin B., & Rappaport M. (1986). The Formation of Adjectival Passives. *Linguistic Inquiry, 17.*

Levy, S. D. (2002). *Infinite RAAM: Initial explorations into a fractal basis for cognition.* PhD dissertation, Brandeis University.

Li, Xiaobin, Szpakowicz, S., & Matwin, S. (1995). A WordNet-based algorithm for word sense disambiguation." *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1368-1374, Montreal.

Lieber, R. (1981). *On the organization of the lexicon*. Bloomington, IN: Indiana University Linguistics Club.

Lieber, R. (1992). *Deconstructing Morphology*. Chicago and London: The University of Chicago Press.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics,* Madrid.

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI.

Litkowski, K. (2004). Senseval-3 task: Automatic labeling of semantic roles. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text,* 9-12, Barcelona.

Lyons, J. (1977). *Semantics*. Cambridge: Cambridge University Press.

Lyons, J. (1995). *Linguistic Semantics: An introduction*. Cambridge, Cambridge University Press.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in striate cortex. *Kybernetik, 14,* 85-100.

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T.J., & Xu, F. (1992). *Overregularization in Language Acquisition*. *Monographs of the Society for Research in Child Development, 57.*

Mayberry, M. R., & Miikkulainen, R. (1999). Using a Sequential SOM to Parse Long-term Dependencies. *Proceedings of the 21st Annual Meeting of the Cognitive Science Society (COGSCI-99)*, 367-372, Vancouver, Canada.

McArthur, T. (1992). *Longman Lexicon of Contemporary English*. Longman Group (Far East) Ltd., Hong Kong.

McClelland, J. L., & Rumelhart, D. E. (1981). An Interactive Activation Model of Context Effects in Letter Perception. Part 1: An Account of Basic Findings. *Psychological Review, 88,* 375-407.

McClelland, J.L., Rumelhart, D.E., & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models*. Cambridge, MA: MIT Press.

Meyers, A. (1994). *A Unification-Based Approach to Government and Binding Theory.* PhD dissertation, New York University.

Mihalcea, R. (2004). WordNet bibliography. Retrieved April 30, 2005, from http://engr.smu.edu/~rada/wnb/

Mihalcea, R., & Moldovan, D. I. (1998). Word sense disambiguation based on semantic density. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal.

Mihalcea, R., & Moldovan, D. I. (1999). *Word sense disambiguation and its application to Internet search.* Masters Thesis defense, Southern Methodist University, April 13, 1999.

Mihalcea, R., & Moldovan, D. I. (2000). An iterative approach to word sense disambiguation. *Proceedings of FLAIRS 2000*, 219-223, Orlando, FL.

Mihalcea, R., & Moldovan, D. (2001). EZ.WordNet: Principles for automatic generation of a coarse grained WordNet. *Proceedings of FLAIRS 2001*, 454-459, Key West, FL.

Mihalcea, R., & Moldovan, D. I. (2001b). Highly accurate bootstrapping algorithm for word sense disambiguation. *International Journal on Artificial Intelligence Tools, 10(1-2),* 5-21.

Miller, G. A. (1996). *The Science of Words*. New York: Scientific American Library.

Miller, G. A. (1998). Foreword. In C. Fellbaum (Ed.), *WordNet. An Electronic Lexical Database*. Cambridge and London: MIT Press.

Moldovan, D., Gîrju, R., Olteanu, M., & Fortu, O. (2004). SVM classification of FrameNet semantic roles. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 167-170, Barcelona.

Nagy, T., Furkó, P., & Tóth, Á. (1997). Foundations of Better Pronunciation *NovELTy (A Journal of English Language Teaching and Cultural Studies in Hungary), 4(1).*

Narayanan, U. E., & Bhattacharyya, P. (2002). Word Sense Disambiguation Using Semantic Graphs. *Proceedings of the first International WordNet Conference*, India.

Nastase, V., & Szpakowicz, S. (2001). Word sense disambiguation in Roget's thesaurus using WordNet. *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources,* Pittsburgh.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science, 14*, 11-28.

Newport, E. L. (1998). Constraints on learning and their role in language acquisition: Studies of the acquisition of American Sign Language. *Language Sciences, 10,* 147-172.

Ngai, G., Wu, D., Carpuat, M., Wang, C., & Wang, C. (2004). Semantic role labeling with Boosting, SVMs, Maximum Entropy, SNOW, and Decision Lists. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text,* 183-186, Barcelona.

Oflazer, K. (1993). Two-level description of Turkish morphology. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics,* Utrecht, The Netherlands.

Oravecz, Cs., & Dienes, P. (2002). Large scale morphosyntactic annotation of the Hungarian National Corpus. In B. Hollósy, & J. Kiss-Gulyás, *Studies in Linguistics, 6.* Debrecen: Institute of English and American Studies.

Paláncz, B. (2003). *Neurális hálózatok.* Manuscript. Retrieved April 30, 2005, from http://www.epi.bme.hu/infotsz/palancz/wwwroot/neural/cover/index.html

Palmer, M. (1998). Are WordNet sense distinctions appropriate for computational lexicons? *SIGLEX-98, SENSEVAL.* Herstmonceux, Sussex, UK.

Patwardhan S., Banerjee S., & Pedersen, T. (2002). Using semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico.

Pedersen, T., & Bruce, R. (1997). Distinguishing word senses in untagged text. In *Proceedings of the 2nd Conference on Empirical Methods in NLP (EMNLP-2)*, Providence.

Pedersen T., Patwardhan S., & Michelizzi J. (2004). WordNet::Similarity - Measuring the Relatedness of Concepts. *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, San Jose, CA.

Pethő, G. (2001). What is Polysemy? – A Survey of Current Research and Results. In E. Németh, & K. Bibok (Eds.), *Pragmatis and Flexibility of Word Meaning*. Amsterdam-London-Oxford-New York-Paris-Shannon-Tokyo: Elsevier.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28,* 73-193.

Pléh, Cs. (1998). *Bevezetés a megismeréstudományba*. Budapest: TYPOTeX.

Pollack, J. B. (1989). Connectionism: Past, present, and future. *Artificial Intelligence Review, 3,* 3-20.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14(3),* 130-137.

Proctor, P. (Ed.) (1978). *Longman Dictionary of Contemporary English*. London: Longman Group.

Prószéky, G. (1989). *Számítógépes nyelvészet: Természetes nyelvek használata számítógépes rendszerekben.* Budapest: Számítástechnika-alkalmazási Vállalat.

Prószéky, G. (1994). Industrial Applications of Unification Morphology. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, 157-159, Stuttgart, Germany.

Prószéky, G. (1996). Morphological Analyzer as Syntactic Parser. In *Proceedings of COLING 1996, 16th International Conference on Computational Linguistics*, 1123-1126, Center for Sprogteknologi, Copenhagen, Denmark.

Prószéky, G. (1997). Újra papír? Lexikonok, enciklopédiák, szótárak - másképp. In I. Polyák (Ed.). *A VII. Alkalmazott Nyelvészeti Konferencia előadásai*, 23-27, Budapest.

Prószéky, G., & Kis, B. (1999). A Unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 261–268, University of Maryland.

Pulman, S. (1991). Two level morphology. In H. Alshawi, D. Arnold, R. Backofen, D. Carter, J. Lindop, K. Netter, S. Pulman, J. Tsujii, & H. Uskoreit (Eds.) *ET6/1 Rule Formalism and Virtual Machine Design Study*, chapter 5. CEC, Luxembourg.

Pustejovsky, J. (1995). *The Generative Lexicon*. Cambridge, MA: MIT Press.

Rappaport, M., & Levin, B. (1988). What to do with θ-Roles. In W. Wilkins (Ed.), *Syntax and Semantics. Volume 21: Thematic Relations*. New York: Academic Press

Ravin, Y., & Leacock, C. (2000). Polysemy: An overview. In Y. Ravin, & C. Leacock (Eds.), *Polysemy: Theoretical and computational approaches*, New York: Oxford University Press. Retrieved 30 July, 2005 from http://www.oup.co.uk/pdf/0-19-823842-8.pdf

Renouf, A. (1987). Corpus development. In J. Sinclair (Ed.), *Looking up: An account of the COBUILD Project in lexical computing and the development of the Collins COBUILD English Language Dictionary*, 1-40. London: Collins COBUILD.

Resnik P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448-453, Montreal.

Rice, K. (1985). On the placement of inflection. *Linguistics Inquiry, 16.*

Richie, G. (1992). Languages generated by two-level morphological rules. *Computational Linguistics, 18(1),* 41-59.

Roeper, T., & Siegel, M. (1978). A lexical transformation for verbal compounds. *Linguistic Inquiry, 9(2).*

Rohde, D. L. T., & Plaut, D. C. (1997). Simple recurrent networks and natural language: How important is starting small? *Proceedings of the 19th Annual Conference of the Cognitive Science Society,* 656-661, Hillsdale, NJ.

Rohde, D. L. T., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition, 72,* 67-109.

Rohde, D. L. T. (1999a). *LENS: The light, efficient network simulator. Technical Report CMU-CS-99-164.* Carnegie Mellon University, Department of Computer Science. Pittsburgh, PA.

Rohde, D. L. T. (1999b). *The Simple Language Generator: Encoding complex languages with simple grammars. Technical Report CMU-CS-99-123.* Carnegie Mellon University, Department of Computer Science. Pittsburgh, PA.

Rohde, D. L. T. (2000). *Lens: The light, efficient network simulator. Version 2.63.* Retrieved December 31, 2004, from http://tedlab.mit.edu/~dr/Lens/

Rohde, D. L. T. (2002). *A connectionist model of sentence comprehension and production*. Unpublished PhD thesis, School of Computer Science, Carnegie Mellon University. Pittsburgh, PA.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart and McClelland (1986), 318-362.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In McClelland and Rumelhart (1986), 216-271.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.

Ruszkiewicz, P. (1997). *Morphology in Generative Grammar*. Gdansk: Wydawnictwo Uniwersytetu Gdanskiego.

Samuelsson, C., & Voutilainen, A. (1997). *Comparing a Linguistic and a Stochastic Tagger*. Retrieved 26 August, 2005, from http://www.ling.helsinki.fi/~avoutila/cg/doc/e-acl97/e-acl97.html

Schwanenflugel, P., & Shoben, E. J. (1985). The influence of sentence constraint on the scope of facilitation for upcoming words. *Journal of Memory and Language, 24,* 232-252.

Seagull, A. (2000). *A Compaction of WordNet Senses for Evaluation of Word Sense Disambiguators. TR726.* Computer Science Department, University of Rochester.

Selkirk, E. (1978). On prosodic structure and its relation to syntactic structure. In T. Fretheim (Ed.), *Nordic prosody, 2,* 111–140. Trondheim: TAPIR.

Shaban, M. (1993). *A Minimal GB parser.* BU-CS Tech Report #93-013. Boston, MA: Boston University.

Siegelmann, H. T., Horne, B.G., & Giles, C.L. (1997). Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 27(2),* 208-215.

Smith, G. W. (1991). *Computers and human language*. New York: Oxford University Press.

Spencer, A. (1991). *Morphological Theory: An introduction to word structure in Generative Grammar*. Cambridge, MA: Basil Blackwell.

Stolck, A. (1990). *Learning feature-based semantics with simple recurrent networks. Technical Report TR-90-015*. ICSI, Berkeley, CA.

Stowell, T. (1981). *Origins of Phrase Structure*. PhD thesis, MIT. Cambridge, MA.

Tabossi, P. (1988). Effects of context on the immediate interpretation of unambiguous nouns. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14,* 153-162.

Taylor J. R. (1995). *Linguistic Categorization. Second edition.* New York: Oxford University Press.

Tengi, R. I. (1998). Design and implementation of the WordNet lexical database and searching software. In C. Fellbaum (1998).

Thompson, C., Patwardhan, S., & Arnold, C. (2004). Generative models for semantic role labeling. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 235-238, Barcelona.

Tóth, Á. (2000). Szóhálózat. *Magyar Tudomány*, *10*, 1235-1237.

Tóth, Á. (2002). Derived nouns in WordNet and the question of productivity. *Studies in Linguistics, 6,* 433-449.

Tóth, Á. (2002). [Review of the book Lawler, J., & Dry, H. (Eds.) (1998). *Using computers in Linguistics: A practical guide*.] *Studies in Linguistics, 6,* 490-493.

Tóth, Á. (2003). Derived nouns and gerunds in WordNet. In M. Fabian (Ed.), *Sučasni doslidženńa z inozemnoï filologiï. Volume 1.* 149-154. Uzhhorod: Department of Foreign languages, Uzhhorod National University.

Tóth, Á. (2004). Polysemy and Homonymy in WordNet. In M. Fabian (Ed.), *Sučasni doslidženńa z inozemnoï filologiï. Volume 2.* 28-36. Uzhhorod: Department of Foreign languages, Uzhhorod National University.

Tóth, Á. (2005). Form and meaning in a recurrent neural network. In M. Fabian (Ed.), *Sučasni doslidženńa z inozemnoï filologiï. Volume 3.* 22-31. Uzhhorod: Department of Foreign languages, Uzhhorod National University.

Vergnaud, J. R. (1973). Formal properties of lexical derivations. *Quarterly Progress Report of the MIT Research Laboratory of Electronics, Number 108*.

Véronis, J., & Ide, N. (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of COLING90*, 289-295.

Verspoor, C. M. (1997). *Contextually-Dependent Lexical Semantics*. PhD thesis. Edinburgh: The University of Edinburgh.

Voegtlin, T., & Dominey, P.F. (2001). Recursive Self-Organizing Maps. In N. Allinson, H. Yin, L. Allinson, & J. Slack (Eds.) *Advances in Self-Organizing Maps*, 210-215, London: Springer.

Voorhees, E. M. (1993). Using WordNet to disambiguate word senses for text retrieval. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.) *Proceedings of the 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, 171-180, Pittsburgh.

Voutilainen, A. (1997). *The EngCG-2 tagger in outline*. Retrieved 26 August, 2005, from http://www.ling.helsinki.fi/~avoutila/cg/doc/engcg2-outline/engcg2-outline.html

Vörös, G. (1997). *Bevezetés a neurális és minősítő számítástechnikába*. Budapest: LSI Oktatóközpont.

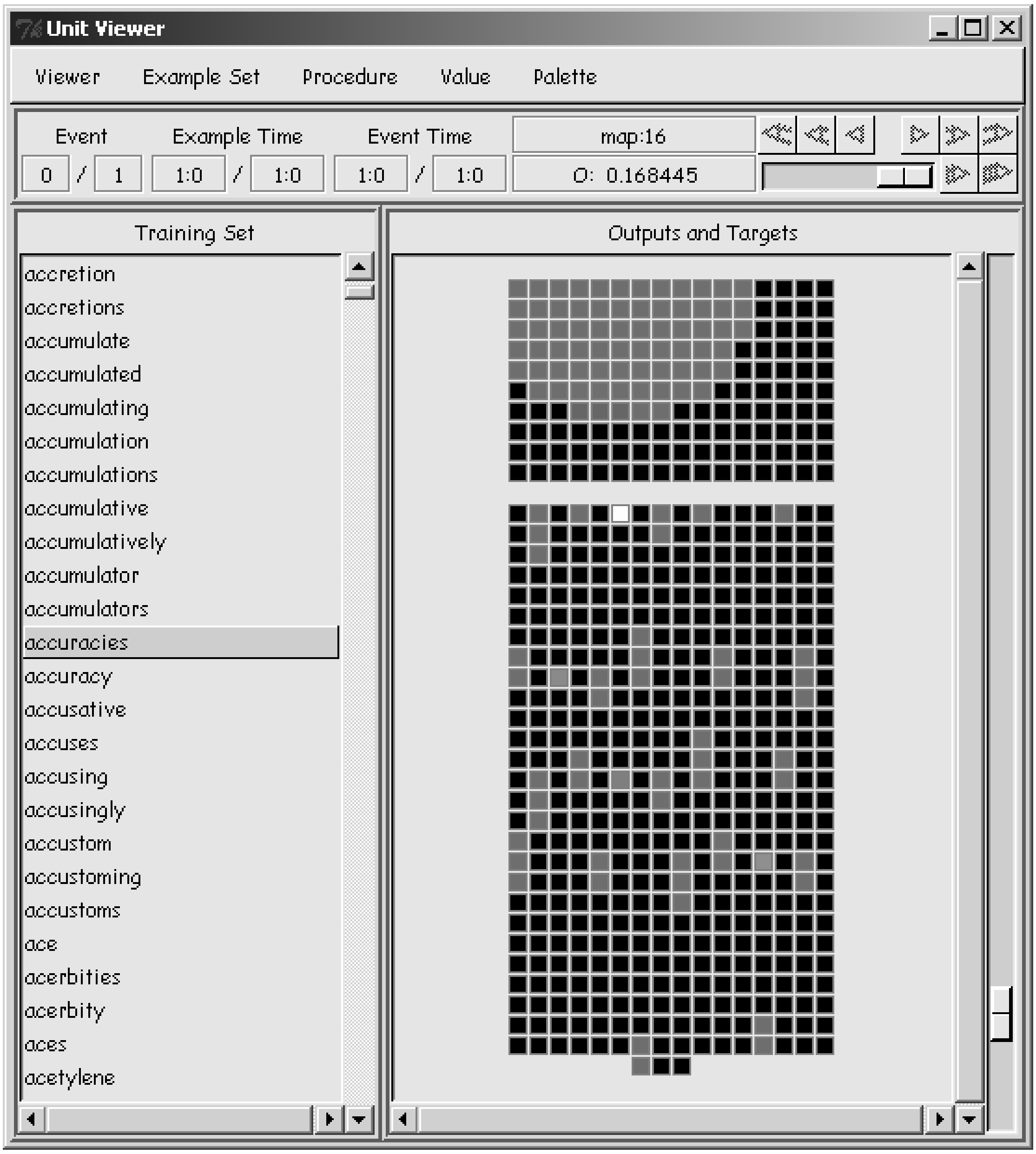
Webelhuth, G. (1995). X-bar Theory and Case Theory. In G. Webelhuth (Ed.), *Government and Binding Theory and the Minimalist Program*. Oxford and Cambridge: Basil Blackwell Limited.

Wiebe, J., O'Hara, T., & Bruce, R. (1998). Constructing Bayesian networks from WordNet for word-sense disambiguation: Representational and processing issues. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal.

Williams, E. (1981). On the notions ‘Lexically Related’ and ‘Head of a Word’. *Linguistic Inquiry, 12(2).*

Wu Z., & Palmer, M. (1994). Verb semantics and lexical selection. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico.

APPENDIX A: Lens screenshots



Event 0 / 1

Example Time 1:0 / 1:0

Event Time 1:0 / 1:0

map:16

O: 0.168445

accretion
accretions
accumulate
accumulated
accumulating
accumulation
accumulations
accumulative
accumulatively
accumulator
accumulators
accuracies
accuracy
accusative
accuses
accusing
accusingly
accustom
accustoming
accustoms
ace
acerbities
acerbity
aces
acetylene

Figure A-1 LENS Unit Viewer displaying one step of training the upper part of the twinmap (cf. section 6.3) Labels for the training examples are on the left (the current example is labeled *accuracies*); corresponding neural activation levels are shown on the right. The lower part of the right pane depicts the input group ("Syllabic structure 1", cf. section 6.4): the most activated node corresponds to the first syllable of the word *accuracies* (notice that we are training the part of the twinmap which is responsible for forward syllable structure representations), other syllables and the "overtones" of the syllables are also activated. The upper part of the right pane shows the corresponding Kohonen-map activation levels. As you see, the neighborhood value is still high (cf. section 5.7), indicating that this screenshot depicts an early phase of the training process.

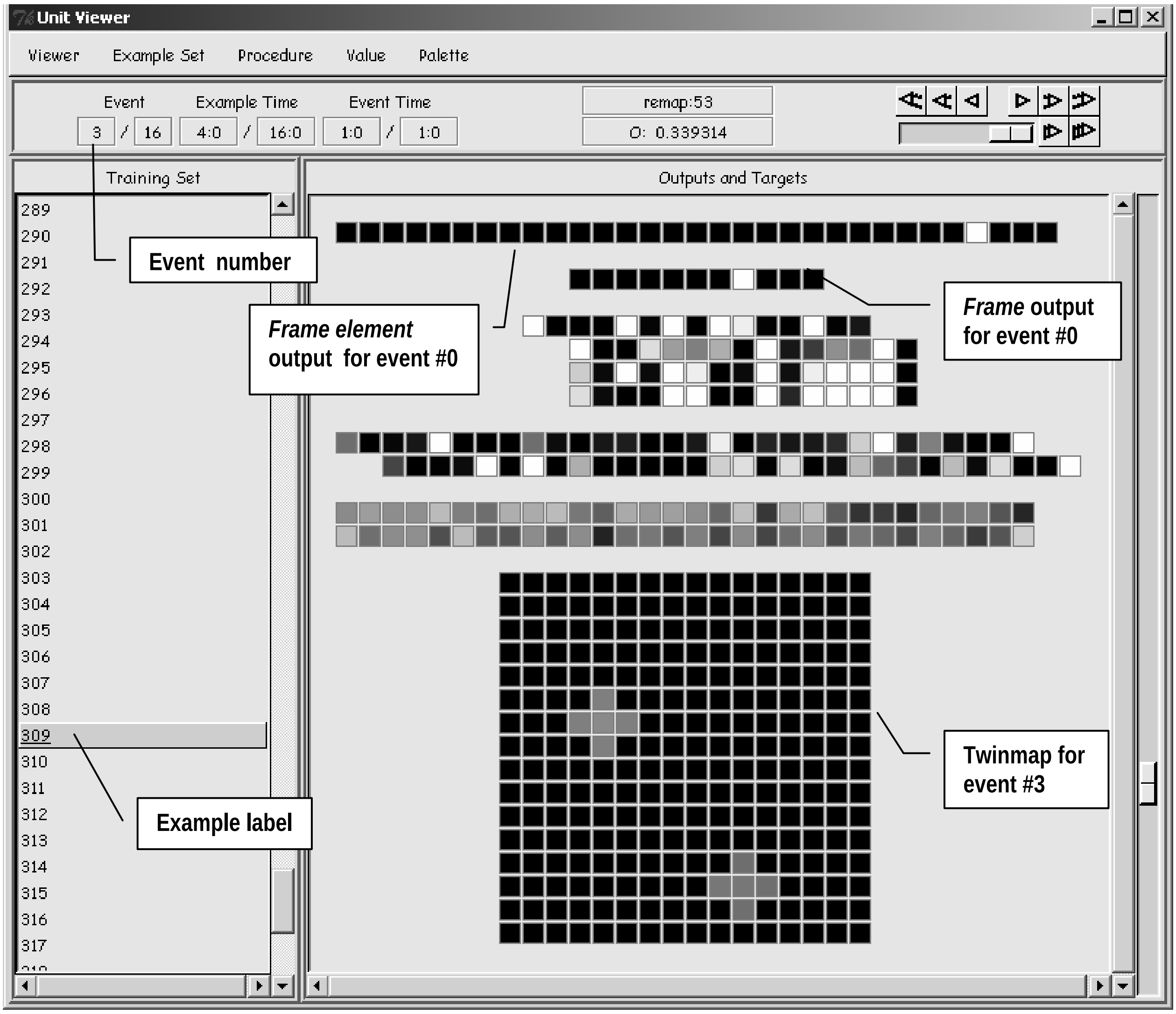


Figure A-2 LENS Unit Viewer depicting a single event (#3 of test example #309).

The textual format of the current example is the following:
(0)thomas (1)is (2)a (3)competent (4)linguist (5)* (8)thomas (9)often (10)blusters (11)about blustering (12)*

The simulator displays the frame and frame element output for "thomas" (event #0, note that the network output is delayed by three events, cf. section 6.4) and the twinmap representation of the word *competent.*

APPENDIX B: Visual assessment of the *crack*-homonymy experiment

Diagrams in this appendix give you visual feedback on the *crack*-homonymy experiment described in section 6.7 (“first experiment”). In the experiment, I selected 10 sentence sequence templates, and generated two sets of 336 sentence sequences with them. In the first set, the non-terminal *tox_um* was rewritten to *marijuana, cannabis, cocaine* and *heroin* while *ace_um* was replaced by *ace, proficient, superb* and *outstanding*. In the second set, *tox_um* became *marijuana, cannabis, cocaine* and *crack*, while *ace_um* was rewritten to *ace, proficient, superb* and *crack*, i.e. the word *crack* appeared in the second set only. Error statistics for both sets can be found in section 6.7: those figures show that labeling the word *crack* did not present difficulties for the network.

I randomly selected 2 templates out of the 10 for further examination. The templates generated 16 examples each (corresponding to 2x16 actual sentence sequences appearing in the second sequence set of the original experiment described above), one fourth of them contained the word *crack*. Then I collected the network output for these examples, and located the output for the event that corresponded to *tox_um* in the first template (diagram B-1 and diagram B-2), and *ace_um* in the second template (diagram B-3 and diagram B-4). Diagrams B-1 and B-3 show *frame output* activation values, while diagrams B-2 and B-4 depict *frame-element output* values for the events in question. The examples are along the x-axis, the y-axis shows the activation level from 0 to 1, and the FR or FE output group units are on the z-axis (11 units in the FR group and 31 units in the FE group, cf. section 6.2).

To interpret the analog output values and collect recall and precision information, I use a defuzzifying classifier throughout my experiments (cf. section 6.6). It locates the most activated output node for each event (its output must also reach a threshold, which is at 0.5). The diagrams show the actual analog output values before they are processed by the classifier. In the diagrams, the most activated node (observed along the z-axis) for each event is considered the output for the given event. As you see, these units are well above the threshold (and very near to the target value of 1.0). The outputs *correspond* to the target FR or FE values for each example event, i.e. the relevant event (in which *crack* shows up) of each example was annotated perfectly with both FR and FE labels. Please note that the *crack* events show very similar output patterns to non-*crack* events: additional output units do not become activated, and the actual activation levels of the most salient nodes seem fairly constant (especially in diagrams B-3 and B-4), despite the fact that the form *crack* are associated with more FE and FR targets during training than other words in the same non-terminal set (e.g. *cocaine*, *competent*).

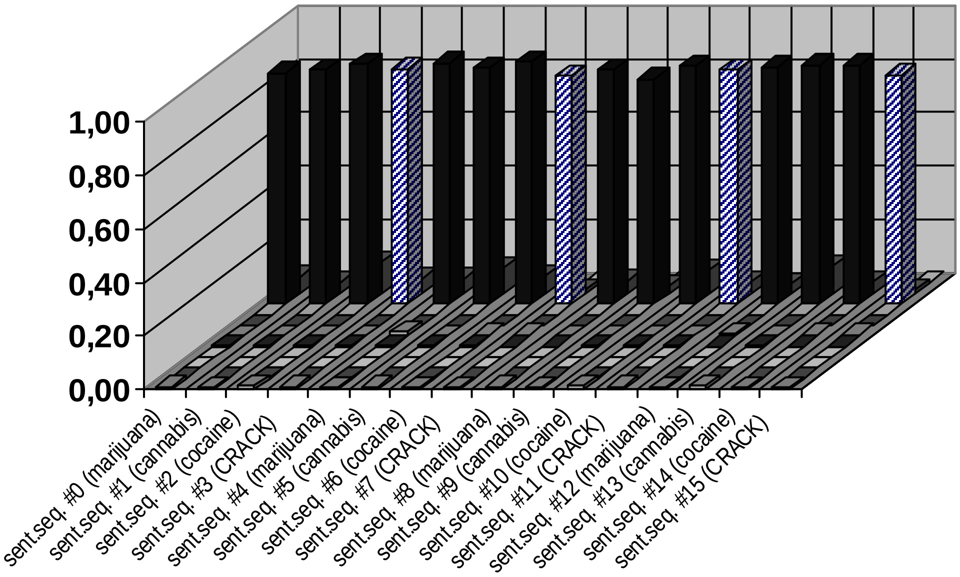


Diagram B-1 FR output for a *tox_um* event (event #13 in sentence sequences #0-#15 of the original experiment). Solid bars are "non-crack" events; events containing *crack* are represented by striped bars.

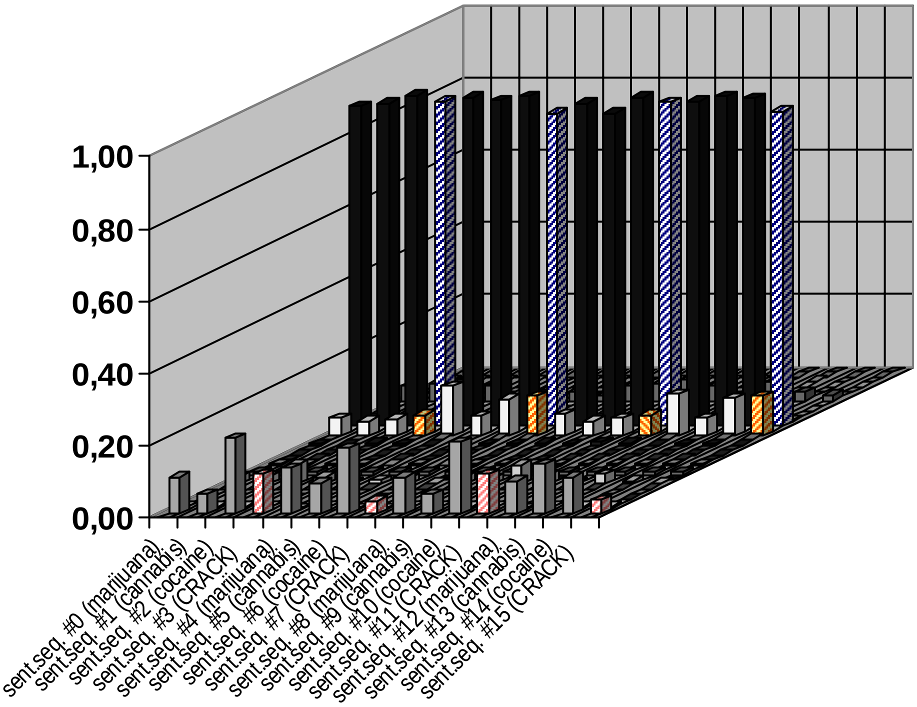


Diagram B-2 FE output for a *tox_um* event (event #13 in sentence sequences #0-#15)

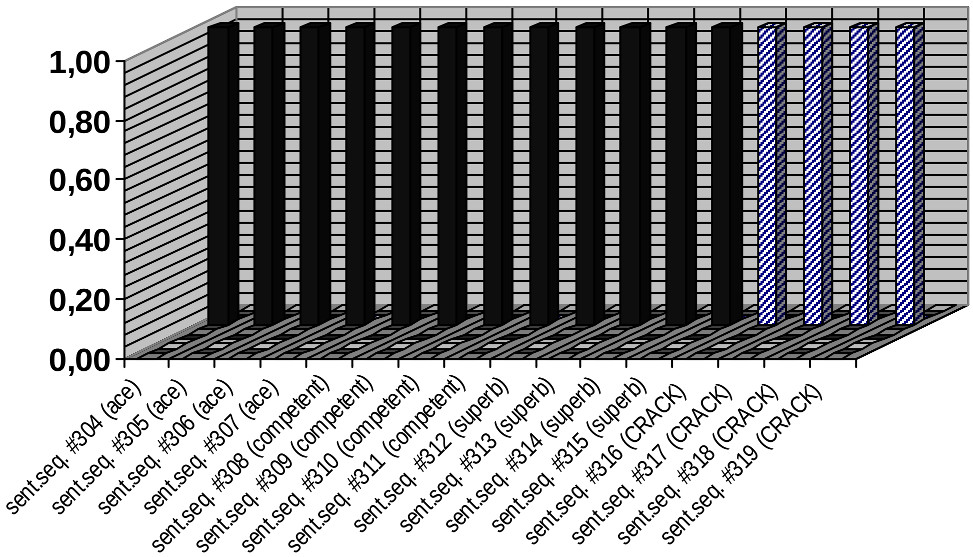


Diagram B-3 FR output for an *ace_um* event (event #6 in sentence sequences #304-#319). Please note that *figure A-2* in *Appendix A* shows an event from one of the examples used here (#309).

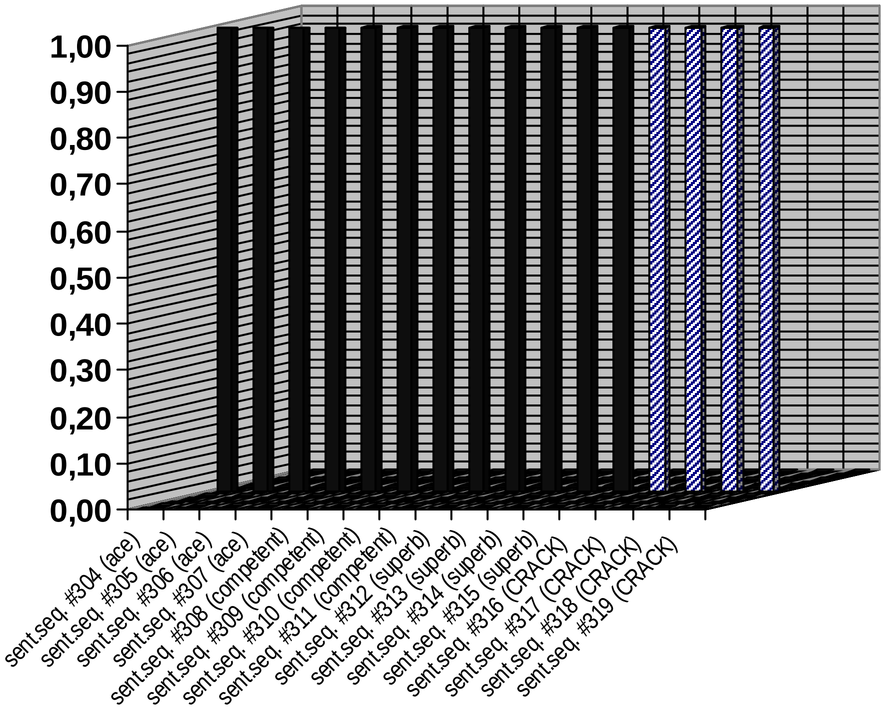


Diagram B-4 FE output for an *ace_um* event (event #6 in sentence sequences #304-#319)

APPENDIX C: Templates and non-terminal sets

Contained in this appendix are some sentence-sequence templates that were used in generating the training data in the ANN simulation (cf. section 6.5).

*#PE1_1001 cracked the #comp_um_1002 * #PE1_0201 #crkgoal_past * #PE2_1101 realized that the $comp_um_1002 was cracked_1000 * #PE2_1101 #patch_past_1100 the $comp_um_1102 **

*crackers_1001 crack_1000 the #comp_um_1002 to #crkgoal_um by #crkmethod_ing_0302 * $crkmethod_ing_0302 is illegal so crackers_0301 commit #crime_um_0300 **

*administrators_1101 prevent crackers_1001 from #crkgoal_ing_1000 by #patch_ing_1100 the #comp_um_1102 **

*#drugcons_ing_0600 #tox_um_0602 is hazardous to health * people_0601 who $drugcons_um_0600 $tox_um_0900 may die **

*#drugcons_ing #tox_um_0104 is hazardous to health * people_0102 who $drugcons_um_0600 $tox_um_0900 may die **

*#PE1_0702 often #comm_s_0700 #tox_about_0703 * I think #PE1_0301 is a criminal and will #gpunished_um_0403 sooner or later #drugbuy_for_0404 **

*#PE1_0801] is a #ace_um_0800] administrator_0802] * #PE1_0702] keeps #comm_ing_0700] #patch_about_0703] **

The following table shows the non-terminal sets that are used in the above templates:

ace_um	*ace* *competent* *superb* *outstanding* *crack* *proficient* *splendid* *fantastic* *excellent* *great* *good* *fantastic*
comm_ing	*babbling* *blustering* *chanting* *chattering* *drawling* *gabbling* *jabbering* *lisping* *mumbling* *muttering* *prattling* *ranting* *shouting* *slurring* *stammering* *stuttering* *whispering*

comm_s	*babbles* *blusters* *chants* *chatters* *drawls* *gabbles* *jabbers* *lisps* *mumbles* *mutters* *prattles* *rants* *shouts* *slurs* *stammers* *stutters* *whispers*
comp_um	*system* *computer* *server* *programme*
crkmethod_ing	*using vulnerability* *using key logger* *using a security flaw* *cracking the password file* *using denial of service attack* *deploying a trojan* *deploying a virus* *infecting with virus* *circumventing virus scanner* *disabling virus scanner* *spoofing address* *stealing password* *finding insecure service*

crkgoal_ing	*stealing data* *stealing personal data* *stealing money* *doing harm* *wrongdoing* *causing damage* *causing loss* *causing data loss* *causing disadvantage* *getting secrets*
crkgoal_past	*stole data* *stole personal data* *stole money* *did harm* *did wrong* *caused damage* *caused loss* *caused data loss* *caused disadvantage* *got secrets*
crkgoal_um	*steal data* *steal personal data* *steal money* *do harm* *do wrong* *cause damage* *cause loss* *cause data loss* *cause disadvantage* *get secrets*

drugbuy_for	*for buying drugs*
	for obtaining drugs
	for buying intoxicants
	for obtaining intoxicants

drugcons_ing	*using* *consuming* *taking* *applying*

drugcons_um	*use* *consume* *take* *apply*
gpunished_um	*go to jail* *go to prison* *get imprisoned* *get sentenced* *get punished* *get severely punished* *get punishment* *get severe punishment* *get prison sentence*
patch_about	*about patching computers* *about patching systems* *about patching servers* *about protecting computers* *about securing computers* *about fixing computers*
patch_ing	*patching* *applying a patch* *protecting* *securing* *fixing*
patch_past	*patched* *applied a patch* *protected* *secured* *fixed*
tox_about	*about marijuana* *about heroin* *about opium* *about crack*

tox_um	*marijuana* *cannabis* *cocaine* *heroin* *crack* *LSD* *magic mushroom* *mescaline* *joint* *opium*

APPENDIX D: An input-file fragment for Lens

The following is a Lens example file fragment containing twinmap input (“{tm}”) and FR/FE targets (“{outFR}”, “{outFE}”) for a *single* training/testing example (cf. section 6.5). See Rohde (2000) for detailed instructions on how the example files should be interpreted. The human-readable form of this example:

[3]crackers [4]crack [5]the [6]system [7]to [8]steal personal data [9]by [10]cracking the password file [11]* [14]cracking the password file [15]is [16]illegal [17]so [18]crackers [19]commit [20]crime [21]*

Event numbers used above ([3]-[21]) correspond to the event when output is expected for the given words/multi-word expressions. Events 0, 1, 2, 11, 12 and 13 are inter-sentential gaps.

```
22
T: {-} *
T: {-} *
T: {-} *
I: {tm 0} *  {tm 0.47} 104 {tm 0.35} 119 {tm 0.47} 120 {tm 0.45} 121 {tm 0.45} 136 {tm 0.52} 169 {tm 0.52} 184 {tm 0.54} 185 {tm 0.51} 186 {tm 0.49} 201
T: {outFR 0} * {outFE 0} *  {outFR 1} 9 {outFE 1} 0
I: {tm 0} *  {tm 0.55} 105 {tm 0.53} 120 {tm 0.57} 121 {tm 0.56} 122 {tm 0.54} 137 {tm 0.57} 169 {tm 0.58} 184 {tm 0.59} 185 {tm 0.56} 186 {tm 0.57} 201
T: {outFR 0} * {outFE 0} *  {outFR 1} 9 {outFE 1} 1
I: {tm 0} *  {tm 0.94} 16 {tm 0.97} 32 {tm 0.96} 33 {tm 0.93} 48 {tm 0.77} 220 {tm 0.72} 235 {tm 0.92} 236 {tm 0.8} 237 {tm 0.82} 252
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.73} 14 {tm 0.72} 29 {tm 0.74} 30 {tm 0.66} 31 {tm 0.63} 46 {tm 0.64} 162 {tm 0.66} 163 {tm 0.54} 164 {tm 0.61} 179
T: {outFR 0} * {outFE 0} *  {outFR 1} 9 {outFE 1} 2
I: {tm 0} *  {tm 0.56} 120 {tm 0.5} 135 {tm 0.59} 136 {tm 0.57} 137 {tm 0.53} 152 {tm 0.6} 184 {tm 0.59} 199 {tm 0.61} 200 {tm 0.58} 201 {tm 0.57} 216
T: {outFR 0} * {outFE 0} *
I: {tm 0} * {tm 0.54} 105 {tm 0.52} 120 {tm 0.55} 121 {tm 0.55} 122 {tm 0.53} 137 {tm 0.6} 169 {tm 0.61} 184 {tm 0.62} 185 {tm 0.59} 186 {tm 0.6} 201 {tm 0.48} 71 {tm 0.44} 86 {tm 0.52} 87 {tm 0.44} 88 {tm 0.4} 103 {tm 0.46} 201 {tm 0.45} 216 {tm 0.48} 217 {tm 0.44} 218 {tm 0.4} 233 {tm 0.45} 116 {tm 0.4} 131 {tm 0.63} 132 {tm 0.5} 133 {tm 0.52} 148 {tm 0.52} 218 {tm 0.56} 233 {tm 0.58} 234 {tm 0.48} 235 {tm 0.54} 250
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.5} 107 {tm 0.55} 122 {tm 0.58} 123 {tm 0.56} 124 {tm 0.54} 139 {tm 0.57} 199 {tm 0.57} 214 {tm 0.6} 215 {tm 0.57} 216 {tm 0.53} 231
T: {outFR 0} * {outFE 0} *
```

```
I: {tm 0} * {tm 0.47} 105 {tm 0.45} 120 {tm 0.49} 121 {tm 0.47} 122 {tm 0.45} 137 {tm 0.53} 171 {tm 0.51} 186
{tm 0.59} 187 {tm 0.51} 188 {tm 0.53} 203 {tm 0.94} 16 {tm 0.97} 32 {tm 0.96} 33 {tm 0.93} 48 {tm 0.77} 220
{tm 0.72} 235 {tm 0.92} 236 {tm 0.8} 237 {tm 0.82} 252 {tm 0.41} 121 {tm 0.39} 136 {tm 0.44} 137 {tm 0.41} 138
{tm 0.42} 153 {tm 0.47} 215 {tm 0.43} 230 {tm 0.48} 231 {tm 0.4} 232 {tm 0.33} 247 {tm 0.55} 121 {tm 0.52} 136
{tm 0.55} 137 {tm 0.52} 138 {tm 0.52} 153 {tm 0.56} 169 {tm 0.57} 184 {tm 0.58} 185 {tm 0.56} 186 {tm 0.57}
201
T: {outFR 0} * {outFE 0} *  {outFR 1} 2 {outFE 1} 3
I: {tm 0} * I: {tm 0} * I: {tm 0} *
T: {-} * T: {-} * T: {-} *
I: {tm 0} * {tm 0.47} 105 {tm 0.45} 120 {tm 0.49} 121 {tm 0.47} 122 {tm 0.45} 137 {tm 0.53} 171 {tm 0.51} 186
{tm 0.59} 187 {tm 0.51} 188 {tm 0.53} 203 {tm 0.94} 16 {tm 0.97} 32 {tm 0.96} 33 {tm 0.93} 48 {tm 0.77} 220
{tm 0.72} 235 {tm 0.92} 236 {tm 0.8} 237 {tm 0.82} 252 {tm 0.41} 121 {tm 0.39} 136 {tm 0.44} 137 {tm 0.41} 138
{tm 0.42} 153 {tm 0.47} 215 {tm 0.43} 230 {tm 0.48} 231 {tm 0.4} 232 {tm 0.33} 247 {tm 0.55} 121 {tm 0.52} 136
{tm 0.55} 137 {tm 0.52} 138 {tm 0.52} 153 {tm 0.56} 169 {tm 0.57} 184 {tm 0.58} 185 {tm 0.56} 186 {tm 0.57}
201
T: {outFR 0} * {outFE 0} *  {outFR 1} 2 {outFE 1} 3
I: {tm 0} *  {tm 0.57} 89 {tm 0.51} 104 {tm 0.58} 105 {tm 0.54} 106 {tm 0.57} 121 {tm 0.54} 168 {tm 0.55} 183
{tm 0.58} 184 {tm 0.58} 185 {tm 0.52} 200
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.52} 77 {tm 0.42} 92 {tm 0.69} 93 {tm 0.68} 94 {tm 0.66} 109 {tm 0.54} 162 {tm 0.63} 177 {tm
0.63} 178 {tm 0.49} 179 {tm 0.22} 194
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.6} 124 {tm 0.55} 139 {tm 0.67} 140 {tm 0.61} 141 {tm 0.63} 156 {tm 0.64} 199 {tm 0.64} 214
{tm 0.68} 215 {tm 0.63} 216 {tm 0.58} 231
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.47} 104 {tm 0.35} 119 {tm 0.47} 120 {tm 0.45} 121 {tm 0.45} 136 {tm 0.52} 169 {tm 0.52} 184
{tm 0.54} 185 {tm 0.51} 186 {tm 0.49} 201
T: {outFR 0} * {outFE 0} *  {outFR 1} 2 {outFE 1} 4
I: {tm 0} *  {tm 0.48} 25 {tm 0.5} 40 {tm 0.72} 41 {tm 0.51} 42 {tm 0.52} 57 {tm 0.47} 182 {tm 0.43} 197 {tm 0.47}
198 {tm 0.47} 199 {tm 0.45} 214
T: {outFR 0} * {outFE 0} *
I: {tm 0} *  {tm 0.55} 105 {tm 0.53} 120 {tm 0.56} 121 {tm 0.55} 122 {tm 0.54} 137 {tm 0.57} 169 {tm 0.58} 184
{tm 0.58} 185 {tm 0.56} 186 {tm 0.57} 201
T: {outFR 0} * {outFE 0} *  {outFR 1} 2 {outFE 1} 5
I: {tm 0} *
T: {-} *
```