University of Oslo
Department of Linguistics

Modeling Word Senses With Fuzzy Clustering

Erik Velldal

Cand. Philol. Thesis in Language, Logic and Information

November 5, 2003

Abstract

This thesis describes a clustering approach to automatically inferring soft semantic classes and characterizing senses of a set of Norwegian nouns. The words are represented by way of their distribution in text, identified as local contexts in the form of lexical-syntactic relations. Through a shallow processing step the context features are extracted for lemmatized word forms in syntactically tagged corpora. The corresponding frequency counts of noun–context co-occurrences are weighted with a statistical association measure, and the distributional profile of a given word is represented in the form of a feature vector in a semantic space model. A hybrid approach is taken when clustering the word vectors; a bottom-up hierarchical method is used to initialize various types of fuzzy partitional clusterings. With the purpose of capturing the notion of typicality the clusters are construed as fuzzy sets, and the words are assigned varying degrees of membership with respect to the various classes. Words are assigned graded memberships in clusters on the basis of their resemblance towards a class prototype. The goal is to automatically uncover semantic classes, where the various memberships of a given word in these fuzzy clusters can be used to characterize its various senses.

Acknowledgements

A lot of people have played a part in helping me complete this thesis. First of all I want to thank my supervisor Jan Tore Lønning for all his patience, assurance and advice. I also want to thank Ruth Vatvedt Fjeld at the Section for Norwegian Lexicography and Dialectology at UiO for letting me use the SNLD corpus and for always being so genuinely encouraging. Thanks are due to everyone at the Text Laboratory, and especially Anders Nøklestad and Lars Nygaard for help on managing the corpus data. Lars has furthermore given me relentless assistance on all sorts of technical issues, including setting up the context data base for this project. Ragnhild Holberg also deserves special thanks for proofreading most parts of the thesis. Thanks to ATCQ for recording The Low End Theory which I listened to incessantly while trying to complete this report. Finally, and most of all, I want to thank Iselin Engan for being there for me in every possible way (smask!) and hauling me over the finish line.

Contents

Chapter 1

Introduction

1.1 Problem Statement and Thesis Outline

Ambiguous word meaning is a core issue in many of the most fundamental and unresolved problems within natural language processing (NLP). Many words can take on a wide variety of different meanings, ranging from subtle and vague indeterminacies to the related, but distinct senses of polysemous words, and the completely disparate senses of homonyms. This situation makes for some major pitfalls for many NLP applications. For instance, in the setting of machine translation (MT) and question-answering (QA) systems, the task of choosing the right sense of a given word can represent a do-or-die decision. Semantic knowledge might provide helpful guidance in both parsing and generation processes.

Correspondingly, in order to deal with the challenges posed by meaning ambiguities, there is a great need for having computational lexicons equipped with broad-coverage semantic information. There exists many hand-crafted repositories of semantic information in the form of machine-readable dictionaries and thesauri, where words are categorized according to different semantic classes and different senses. However, the *productivity* of natural language also means that words continually receive new, extended or altered meanings, and in many applications there is a need for *specialized* semantic lexicons, adjusted to fit a specific subject or domain. The manual compilation and maintenance of such semantic resources is a time-consuming and labor-intensive process, and in practice their coverage is therefore often severely restricted. All these factors indicate a great need for methods that can *automatically* infer semantic relations directly from data.

In the project described in this thesis we seek to automatically categorize a set of Norwegian nouns, in order to reflect their various senses and relations of semantic similarity. We present various methods for bootstrapping semantic classes and word senses on the basis of corpus data. The properties or attributes by which we seek to characterize the words, are their respective *contexts of use*; the nouns are described by the linguistic environment of their occurrences in text. Hence, a fundamental assumption underlying the project is that the linguistic context of a word may have something essential to tell us about its meaning. Such *distributional approaches* to modeling word similarity have gained increasing interest over the last years, in pace with the tremendous increase in available electronic texts and corpora, as well as computing resources and software. Thus, computer-based corpus linguistics seem to revitalize many ideas from the tradition of empirical linguistics.

1.1.1 The Distributional Hypothesis

> *Meaning is use.* Wittgenstein (1953)
>
> *The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.* Harris (1968)
>
> *You shall know a word by the company it keeps.* Firth (1968)

The empiricist claims above are so often and routinely cited in works taking a distributional perspective on lexical semantics that they have become somewhat like maxims or dictums for the entire approach. The idea that the meaning of a word can be deduced from its lexical context is often known as '*the distributional hypothesis'* or '*the contextual theory of meaning'*, – formulations of Harris and Firth respectively.

Distributional information undoubtly plays an important role in human's ability to process language. This is especially obvious when we try to make sense of a new or novel word. Consider the word *retawerif* in the following sentence:

(1.1.1) His head was throbbing after drinking too many shots of *retawerif* at the party last night.

As if by reflex, we immediately assume that *retawerif* is some sort of strong alcoholic beverage. Although we have (most probably) never encountered this particular word before, the meanings and connotations of the accompanying (as well as absent) words, indicate its meaning quite clearly. Also when interpreting already familiar words, the surrounding context is a major source of cues and constraints. Consider the word form *shots* in sentence (1.1.1) above. Although this word might actually take on any one of a range of different senses, we effortlessly select the appropriate meaning that corresponds to a serving of liquor. If we had to describe the meaning of the word in isolation however, we would have to include a host of other senses, such as: a snapshot, photograph or movie scene; a swing, stroke or throw; an attempt or effort; a guess; a missile or bullet; an injection; an action of firing a projectile; a marksman or someone who shoots; and a range of other more or less conceivable alternatives. Many homonyms are even ambiguous with respect to part of speech, but seeing that our concern here is only with categorizing nouns, these ambiguities will not be that relevant. Nonetheless, we see that not only are the *co-occurring words* important determinants of the meaning that we assign a given target, but also the *relations* that we find to hold between them. Of course, this does not mean that distributional information and linguistic context is all there is to word meaning. Influence of body language, speech cues such as stress and prosody, encyclopedic knowledge, and personal background, to name but a few things, are all important factors when interpreting the meaning of words. All the same, and particularly in relation to *written* language of course, we expect that the meaning of words must also somehow reside in their distribution of use.

We will not dwell upon the claim that context can provide essential cues when decoding the meaning of words. This is all quite intuitive and uncontroversial, as is also the fact that many words are ambiguous and have multiple meanings. However, in relation to NLP, these simple insights have some important implications.

If the meaning of words can be discerned and characterized by properties of the environment in which they appear, then these properties can be used to compare and contrast different words. This means that, if we accept the distributional hypothesis, we can use *distributional similarity* as a measure of *semantic similarity.* If two words tend to keep the same sort of lexical company, we can assume that they are likely to have the same sort of meaning. The *contextual pattern* that a given word exhibits, i.e. the totality of contexts that the word appears in over the course of many uses, can in this way provide a basis for modeling its senses. Clearly such a distributional approach to meaning touches on many key issues concerning both *representation* and *acquisition* of semantic knowledge for a computational lexicon.

Most work on distributional characterization of word similarity has been based on co-occurrences within *n*-grams, broad context windows, or even documents, without incorporating much linguistic information. However, the previous work of, among others, Hindle (1990), Grefenstette (1992), Pereira et al. (1993), and Pantel and Lin (2002), clearly indicate the plausibility of extracting semantic relations on the basis of more "local" contextual information concerning grammatical and syntactic relations between co-occurring words. In the first chapter to follow, we will make the concept of context more precise, and section 2.1 discusses how different definitions of context will result in different types of semantic information. In this connection, we also try to clarify the notion of *semantic similarity* that we seek to capture, as opposed to the notion of *semantic relatedness.*

The nouns that comprise the data set for this thesis, are characterized by way of their co-occurrences with various grammatical and syntactic constructions. In order to *extract* the local context features, one needs a *syntactic parser.* Unfortunately however, no such parser is available for Norwegian. We therefore implement an *ad hoc* shallow processing tool, that works on morphosyntactically tagged corpora. This "poor mans" shallow parser is outlined in section 2.3.2. Section 2.3.3 furthermore describes a *windowing tool* that extracts a broader form of word context, but this is not used in the final representations that we use for the automatic categorization task described in this report.

1.1.2 Soft Clusters in a Semantic Space

We have established that the nouns are described by way of their observed lexical-syntactic relations in text. When *representing* these distributional patterns, we use a *semantic space model.* The contextual features are seen as defining the coordinates of an abstract feature space, where each word is given a vectorial representation. Each contextual property is taken to correspond to a dimension, and the totality of contextual features make up a multi-dimensional context space. The details of the vector space model are described in chapter 3. In section 3.1 we present some details concerning the process of *selecting the data* *set*, before we move on to discussing how to ascribe significance and relevance to the various co-occurrence counts of nouns and contexts. Instead of relying on raw frequency alone, statistical *association measures* are applied in order to determine the strength of salience held by the various contextual attributes in relation to different nouns. Various examples of such weighting schemes are described in section 3.2. The association weighting of the frequency counts also provides a way of highlighting the most salient usage patterns of various words. Section 3.2 and 4.3.1 show examples of such distributional profiles for both words and classes respectively, where the local contexts are ranked according to association strength.

As mentioned, we can use the distributional correspondence of words as an indicator for semantic similarity. Furthermore, in the semantic space model, we construe the distributional profiles of words as points or vectors in a noun–context space. This means that we can let the *semantic similarity* of words be given as a function of the *spatial proximity* between their feature vectors in the model. In this way, a semantic space model gives us a way of immediately retrieving a set of words that are most similar to another given target word, by searching for the points that lie the closest to it in the space. Section 3.3 shows how standard measures of proximity in a vector space, can be used to identify semantically similar words, in the form of *nearest neighbor* relations. It is also worth noting that, although we will not actually be concerned with the issue of how word meaning is represented *mentally*, there is in fact a growing body of research indicating also the *psychological plausibility* of semantic space models (see e.g Lund et al., 1995; Lowe and McDonald, 2000; Gärdenfors, 2000; McDonald and Ramscar, 2001).

Section 3.4 rounds off the chapter with a closer look at some properties of word meaning and conceptual classes in general, in order to establish some requirements that a relevant model should accommodate.

Categorization and Clustering

Clustering is a generic term for a range of methods within statistical learning and data analysis, and comprise an important type of approach within the broader area of *pattern recognition* (PR) and *machine learning* (ML). Duda and Hart (1973) describe PR as "*a field concerned with machine recognition of meaningful regularities in noisy or complex environments*", while Bezdek (1981) describes it simply as "*the search for structure in data*". On the basis of the vectorial word representations described above, the techniques for clustering provides a range of possible approaches to the task of automatically discovering or constructing *classes* of similar words. This is what we turn to in chapter 4, which describes a selection of clustering methods in detail.

The process of clustering can be seen as a form of *category induction*, in parallel with that of *categorization* itself. Simply put, by forming groups and lumping together things which resemble each other (with respect to some criterion), clustering is a process of *generalization* and *abstraction*.

Within the methodology of statistical learning, the term *classification* is usually reserved for the situation where we categorize something according to already established categories with known and given examples of the members of the various groups. When *training* a classifier the task is to learn to assign *new* instances to *predefined* classes. This is an instance of what is called *su-*

pervised learning within ML, and requires *training data* that correctly specifies what we are trying to learn. Clustering methods, on the other hand, does not require any such predefined system of classes and labeled exemplars, and is also known as *self-organization* or *unsupervised learning* (see e.g. Bezdek and Sankar, 1992a). The process of discovering, or constructing, the classes and the process of assigning the memberships are done simultaneously and automatically.

The general objective of a cluster analysis is to partition the data into subsets, so that the similarity among members of the same group is high (homogeneity) while the similarity between the groups themselves is low (heterogeneity). The objective of the particular clustering task described in this thesis is to partition a set of nouns, as described by their contextual distribution in text, into a set of "conceptual groups" that reflect semantic similarity.

We mentioned how clustering can be understood as a process of generalization and abstraction. By recognizing common and recurring features, we can view words as *kinds* of words, rather than just individual tokens. By treating similar that which is similar, we can reason by analogy and transfer knowledge between different situations. We can make predictions and inferences from expecting that things that are alike in *some* respect might also behave similarly in yet other respects. Clustering can thereby contribute to alleviate the infamous *sparse data problem* or the *zero-frequency problem* within statistical NLP. In relation to language modeling, the problem concerns the fact that no matter how large our data sample is, there will always be many relatively common events (in the form of word co-occurrences, grammatical constructions, symbol sequences, etc.) that may be observed only very rarely or not at all (see e.g. Dagan et al., 1995). Due to the inherent productivity of language, there will always be perfectly plausible events that remain unseen and thereby unaccounted for, if a language model is based solely and directly on observed occurrences. However, by summarizing our data and generalizing information about similar situations, we can use knowledge of *observed* events when making predictions concerning *unobserved* events. To this end clustering can provide a way of deciding which events are similar and dissimilar to each other, by viewing the data in terms of *types* (classes) instead of *tokens* (individual words). This line of reasoning is the basis for *class-based language models*, and for example Brown et al. (1992) and Pereira et al. (1993) show how distributionally derived word classes can be used to estimate probabilities of unseen events.

Another related use of class-based similarity models can be to express *structural restrictions* and performing *structural disambiguation*. When arguing for why it might be useful to have a classification of words into semantically similar sets, Hindle (1990) : "*A variety of linguistic relations apply to sets of semantically similar words. For example, modifiers select semantically similar nouns, selectional restrictions are expressed in terms of the semantic class of objects, and semantic type restricts the possibilities for noun compounding.*" Accordingly, Li and Abe (1998) show how clustering can be used to improve the accuracy of disambiguating noun compounds and PP-attachments. Having words assigned to semantic classes might also aid the resolution of co-reference and anaphoric bindings.

Fuzzy Clustering

The objective of our word clustering is, as said, to find groups in the data that can in some way be seen to correspond to semantic classes, which can further be taken to reflect the various senses for a set of nouns. That is, we want the word clusters that are formed to represent *meanings* in some sense, with words categorized according to their semantic content. As we know, words are frequently seen to be homonymic, polysemous or vague, and any attempt to pin down some aspect of word meaning should take these possibilities of ambiguity into account. Words must be allowed membership in *multiple* classes if we want to reflect the fact that they may have multiple meanings. Moreover, different words can represent more or less typical instances of a given concept. Some words may represent a clear-cut instance of a given category, while others are peripheral or border-line cases. Correspondingly, the *boundaries* of conceptual categories are often, by their very nature, fleeting and undetermined.

In order to represent the memberships of words in semantic categories, we will adopt the notion of *fuzzy sets* as introduced by Zadeh (1965). Fuzzy sets are meant to deal with *vague categories* that are imprecisely defined. In contrast to classical sets, objects can "belong" to a fuzzy set with *varying degrees of membership*. The nouns in our data set will correspondingly be assigned *fuzzy memberships* across the various "sense classes". The membership values are furthermore taken to indicate the degree of *typicality* or *compatibility* that a word holds towards the concept expressed by a given class.

The fuzzy *membership functions* are based on the distance between a word and a class *prototype* in the semantic space. This means that, the strength of membership that a word holds in a given cluster reflects its resemblance to a prototypical representation of the class. Various examples of *soft* clustering schemes are applied in order to *automatically elicit* the fuzzy membership functions directly from the given corpus data. The notion of fuzzy sets is further introduced in section 3.4.2, but frequent returns to the issue of fuzzy membership functions will be made throughout chapter 4.

Although various soft clustering methods are used for eliciting the fuzzy membership functions, the overall clustering approach presented in this thesis amounts to a *hybrid* method; the terminal clusters of a *hierarchical* method is used for initializing a second pass of (various types of) *partitional fuzzy clustering*. Throughout chapter 4 we switch between describing the various clustering techniques and reviewing their application to the noun data. A general outline of hierarchical methods are given in section 4.1.1, while the particular *agglomerative* method that we use in the initial phase of the noun clustering is presented in section 4.1.3. We then move on to describe and apply three types of fuzzy methods; *fuzzy c-means*, *possibilistic c-means*, and *possibilistic prototype classification*, presented in sections 4.2.3, 4.2.5 and 4.3 respectively. All of the fuzzy methods are initialized with the *centroids*, or prototypes, computed for the classes that result from the first step of bottom-up clustering.

Fuzzy clustering techniques have predominantly been used in such application areas of pattern recognition as *image processing* and *computer vision*, and are often placed within the paradigm of *soft computing*. However, fuzzy clustering models may be very attractive in relation to modeling conceptual classes and word senses as well, by virtue of allowing for multiple and graded memberships. This is in contrast to the *hard* classes and *crisp* memberships of conventional clustering methods. Again, various properties of a clustering model that might support the task of inferring semantic categories will be discussed in section 3.4.

Chapter 2

‘Context’ in Context

We have so far casually talked about using the “context” of words without really specifying what this context is supposed to be made up of. There are many different ways to define what exactly is to count as the context of a given focus word, and the choice has a lot of import regarding the type of relations that one is likely to find. Different types of contexts provide different types of cues about the semantics of a word. Various such issues that relate to the different ways of delineating and extracting the contextual distributions of words as they occur in text, are examined in the first section of this chapter. Furthermore, if we are to intelligibly talk about word context and word meaning, we also need to explicate the meaning of “a word” itself, – a concept that is not entirely unproblematic. In section 2.2 we put the notion of *a word* under closer scrutiny and try to arrive at a working definition. While trying to clarify the concepts of words and contexts, we also look at how these entities are isolated from text. A *shallow parser* is set up for the task of extracting contextual word features from corpora of tagged text. While many issues related to the parsing process are discussed as we go along, section 2.3 describes the particularities of the implementation in more detail.

2.1 Types of Context

There is a wide range of techniques in use for extracting contextual features in applications that rely on distributional representations of words. Depending on what sort of semantic aspects the representations are meant to reflect, different definitions of word context are appropriate.

2.1.1 Topical Context

One extensively used technique is that of *sliding context windows*. Depending on a size parameter, the context is simply given by all the words spanned by the window, e.g. 100 words preceding and following the target word. Patel et al. (1998) categorizes the windows as being either *rectangular*, treating every word occurring within the window as equally important, or *triangular*, weighting the importance of a context word according to its distance from the target. The context is also often narrowed down by only including words from the same sentence as the target word appears in. If the linear ordering of the words within a context window is ignored, it is often known as a *bag of words model* (BOW) (see Manning and Schütze, 1999, ch. 7). A preprocessing step that is also commonly found in windowing approaches, is to filter out closed-class words or function words through so-called *stop lists*. The idea is that only *content* words contributes significantly to indicate the meaning of a word.

These are all still rather coarse definitions of word context, and they fall within what Miller and Leacock (2000) call *topical* context. These types of contexts seem to capture information about the general topic or theme of discourse. Various subject areas and topical domains can be seen to have their own associated sub-vocabulary. Even if we were to restrict our focus to a single newspaper, the words that are most typical of the financial section, are likely to differ from the words found in the sports pages.

When comparing words on the basis of their topical contexts then, one tends to find relationships of semantic *relatedness*, rather than semantic *similarity*. In order to illustrate the distinction between the two kinds of relationships, consider the two following sets of words and the ways in which they team up with *car*;

(2.1.1) car, train, bicycle, truck, vehicle, airplane, buss

(2.1.2) car, road, wheel, gasoline, motor, driver, license

Intuitively, the words in (2.1.1) are semantically more *similar* to *car* than those in (2.1.2), even-though the words of (2.1.2) can be said to be more closely *related*. The relations of semantic similarity can be seen to represent a special case of semantic relatedness (Resnik, 1999). In "Saussurian terms", we can say that, while the members of (2.1.2) show more of a *syntagmatic* likeness, the relationship between the words in (2.1.1) is one of *paradigmatic* similarity. Intuitively, words that are topically related, as opposed to semantically similar, are not even required to be of the same syntactic category.

If a cluster analysis is done on the basis of topical contextual representations then, we are likely to form groups of words that belong to the same semantic domain or theme, but that are not necessarily similar in meaning. The noun clustering performed in this project however, is instead geared towards capturing the distinctions and relations of semantic similarity.

2.1.2 Local Context

Indicators that are attuned to the *meaning* of a word, rather than the broader topical aspect, are more likely to be found in the *local* context. While topical context seems to give clues about the semantic domain of a word, local context gives clues about its semantic content. An example of local context is the *grammatical relations* and *dependencies* that a target holds to other words. Other local features can include information about such things as inflection, capitalization, part-of-speech or other syntactic or grammatical properties of the target or the words in its immediate vicinity. Distributional representations of words based on local context including information about grammatical relations, has been applied with good results in tasks such as word sense disambiguation (WSD) (Hearst, 1991; Resnik, 1997) and when judging word similarity (Hindle, 1990; Grefenstette, 1992; Lin, 1998).

The idea that local context, and grammatical relations in particular, can provide strong indications about the semantic content of the words involved, is also quite intuitive. When words combine in a construction they often impose some sort of semantic constraints on each-other. This is especially evident in the case of verbal predicates which often show strong preferences regarding the semantic type of their arguments. Given a predicate such as *eat*, we expect its object to be something *edible*, and the subject to be some sort of *animate living entity*. "Pete ate the carrots" sounds fine, while "The carrots ate Pete" sounds anomalous and slightly paranoid. The point is that things that are *x'ed* are often different *kinds* of things than those that *x'es*, – both of which may be altogether different from the kinds of things involved in *y'ing*. The restrictions on combinations of constituents are not necessarily categorical or absolute, but they are pervasive enough as to impose a *regularity* and *pattern-like* behavior across constructions. Words that typically find themselves in the complement position of a predicate like *eat*, might also frequently appear in combinations with other constituents such as '__ *in the oven*', '*serve* __', '__ *on the table*', '*delicious* __', or '*pass the* __'.

Given that we want our clustering analysis of nouns to result in conceptual classes that in some way reflect relations of semantic similarity, the discussion above clearly motivates the use of locally founded distributional representations. If representing words on the basis of their local context seems to reflect their semantic content rather than their thematic domain, we can reasonably expect that a clustering analysis of these representations reveals relations of similarity rather than relatedness. This intuition also seems to be confirmed in a previous study by Jonsdottir, Velldal, and Holberg (2002), where a clustering analysis of a small set of Norwegian nouns was carried out on the basis of a sample of verb–object relationships extracted from the Oslo-Bergen Corpus. The nouns where clustered using an agglomerative group-average method, – a technique that is described i section 4.1. Many of the word groups in the resulting set of clusters seem to clearly display relations of semantic similarity. The encouraging results from this experiment seem to attest to the viability of the overall approach.

The specific type of local contexts used for the noun representations for our current project, are based on a more comprehensive set of syntactic constructions and grammatical relations. The set of relational types comprises adjectival modifications, prepositional phrases, possesive modification, noun-noun conjunction, noun-noun modification, and verbal subjects, objects and indirect objects. A given contextual *feature* of a given noun, consists of one of these relational types in addition to the constituent that participates in this construction together with the noun. As an example, the Norwegian noun *kake* (cake) might have the feature of appearing as the objects of the verb *bake* (bake), and being modified by the adjective *hjemmelaget* (homemade). The final noun representations consist of co-occurrence counts for a given set of such features. The feature values thus express the number of times the features and the nouns are observed to appear together in a corpora of tagged text. In section 2.3 and chapter 3 we review the features and the feature values in more detail.

A key component of local context is of course the syntactic category of the words. We have assumed that an important aspect of knowing the meaning of a word is knowing how it is used, but an important aspect of the latter is knowing its part-of-speech. As noted in Miller and Leacock (2000), the particular contexts in which nouns, verbs and other word classes can appear, constitute distinct categories of contexts. Part-of-speech is thus an important part of the information needed to characterize a word distributionally. Since, for the purpose of this project, we only need to characterize nouns, the representations that we build can be thought of as inherently typed for syntactic category, – they are all noun contexts.

To extract the grammatical relations necessary to map the local contextual distribution of nouns, a *parser* is needed. Hindle (1990) notes that "the stumbling block to any automatic use of distributional patterns" is often the lack of a "robust syntactic analyzer". Unfortunately, this was indeed also the main obstacle for building the distributional rerpresenataitons for this project, as no parser is presently available for Norwegian. As an ad hock solution, a "poor man's", shallow parser modul, – Spartan[1], was set up. On the brighter side, the shallow parser does not have to analyze raw and unrestricted text directly. Instead, it works on top of a syntactic and morphological analysis performed by *The Oslo-Bergen Tagger*. The tagger is developed by the Text Laboratory at the University of Oslo (UiO) and the HIT-center.[2] Based on Constraint Grammar rules, a formalism developed by Fred Karlsson at the University of Helsinki (see Karlsson et al., 1995), it performs morphosyntactic disambiguated tagging of Norwegian text.[3] The analysis of the tagger sets the principal preconditions for the way Spartan identifies words and their contextual features. Spartan and the corpora tagged by the Oslo-Bergen tagger are presented in more detail in section 2.3.

As a final remark on the discussion of local and topical context types, we might note that they also seem to differ in regards to the *frequency strata* of words that they are suited for. Grefenstette (1993) compares classical windowing techniques to methods using lexical-syntactic relations for finding word similarity relations in corpora. The window based approach seems more viable for infrequent and rare words, for which an analysis restricted to syntactical information alone would not provide sufficient information to make any judgements about their semantic content. Both Miller and Leacock (2000) and Grefenstette (1993) find that local context seems to provide very precise sense indicators. Unfortunately, it is simply not always available for less frequent words. The distributional representations must be based on only a limited number of contextual features. Many infrequent words might then simply be too sparse on empirical evidence for any reliable judgement about their meaning to be made. In relation to WSD, Leacock et al. (1993) notes that representations based on local context provide for excellent precision but low recall. Miller and Leacock (2000) and Resnik (1997) suggest that an important direction of research should be towards ways of combining topical and local types when building representations.

Although only local context features are employed in the noun clustering described later, this project also provides a windowing tool, – Conwin, in order to make representations based on topical context possible. Conwin extracts content words from a context window which is delimited by a specified number of sentences. The default window size is three, for which the context of a given target word is defined as content words occurring within the previous, the current, and the following sentence. The windowing module is, like Spartan,

[1] Shallow PARsing of TAgged Norwegian text
[2] Center for Humanities Information Technologies, at the University of Bergen
[3] Further information about the tagger can be found in Hagen et al. (2000)

implemented to run on text tagged by the Oslo-Bergen Tagger, and further details about the implementation are given in section 2.3.3, after we describe the shallow parser and other issues related to the shallow processing.

As a follow-up to the current project, it could be interesting to compare the results of clustering the same set of nouns with the same methods as described later, but where the distributional representations are instead based on a wider context extracted with Conwin. When constructing semantic classes based on distributional represenations of words, one might also benefit from a division of labor between the different types of contextual representations. The core members of the classes could consist of high-frequency words clustered on the basis of reliable features of the local context. One representation does not exclude the other however, and distributional profiles based on topical contexts could also be associated with the words and the classes. When words of less frequent appearance are to be classified or compared, one can then fall back on a representation of a broader contextual distribution.

2.2 What is a word?

When given a sentence, how do we break down the words that go into it? The units of a sentence can be cut and sliced in a number of different ways, and the resulting pieces that we call "words" can correspond to a range of quite diverse entities. We will not make any claims about right or wrong for the different conceptions of a word, but only demonstrate how words are isolated in this particular project, and that there is more than one way to do it. Consider the components of sentence (2.2.1) below;

(2.2.1) *En* ***programmerer*** ***programmerte*** ***programmer***.
A **programmer** **programmed** **programs**.
'A programmer programmed programs.'

A variable number of words might be identified from sentence (2.2.1) depending on the particular methodology used. According to different approaches, two, four, seven or more words might be extracted. Consider first the case where no linguistic information is available to help us generalize about the symbols we see. If all we have to go by are the full-form "graphical" words, then we have *four* units above, – corresponding to each of the observed symbol sequences separated by white space. A more principled alternative is to try to uncover some underlying form of the word; by "ignoring" morphological distinctions one can achive a mapping of word *tokens* to a more abstract notion of word *types*. Both *stemming* and *lemmatization* are such techniques that aim at some form of morphological normalization, the former being somewhat less sophisticated, but also less computationally expensive, than the latter. As described in for instance (Carlberger et al., 2001), a stemmer uses an ad hoc set of stripping rules and exception lists to transform word forms into some sort of least common denominator for their morphological variants. This truncation process can include both prefix and suffix removal. The resulting "stem" is not necessarily an actual word itself, and should not be confused with a word *lemma* (see Carlberger et al., 2001). Applying a brute force approach such as stemming to the words in (2.2.1), we might end up with only *two* base forms, corresponding to, say, the stems *en* and *program*.

Lemmatization on the other hand, is a more linguistically informed approach and requires a complete morpho-grammatical analysis. Lemmatization reduces the inflectional and variant forms of a word to their base form, so that all conjugations of a verb are represented by its infinitive form, all inflectional variants of a noun are represented by its nominative singular form, and so on. The lemma can be seen to correspond to the dictionary look-up form of a word, and can be defined as the coupling of a word base form, with its inflectional variant forms and a designated part-of-speech. If the words of the sentence in example (2.2.1) are lemmatized on the basis of a POS-analysis, we might identify *four* words, – corresponding to the determiner *en*, the nouns *program* and *programmer*, and the verb *program*. But considering the fact that many words are homonymous and might be ambiguous with regard to syntactic category, the list of lemmas might easily grow. To illustrate the case of POS ambiguities and several possible grammatical analysis, example (2.2.2) shows the output of the Oslo-Bergen Tagger for sentence (2.2.1).[4]. The input surface forms are shown in brackets, while the suggested lemmas are listed as indented entries beneath each token.

(2.2.2)
```
"<En>"
        "en"            det   @det>
        "en"            pron  @subj
"<programmerer>"
        "programmere"   verb  @fv
        "programmerer"  subst @loes-np @obj @subj
"<programmerte>"
        "programmere"   adj   @adj>
        "programmere"   verb  @fv
"<programmer>"
        "program"       subst @loes-np @obj @subj @i-obj
"<.>"
        "$." clb <<< <punkt>
```

Only the functional tags are displayed here in addition to POS, but note that many other syntactical tags have been left out to avoid unnecessary clutter. The tag set used by the Oslo-Bergen Tagger is divided into *morphosyntactic* and *syntactic* tags. The former marks syntactic category or part-of-speech in addition to features such as gender, definiteness, tense and so on. The latter indicate syntactic functions like subject, object and the like. An overview of a subset of the tags indicating syntactic function is given in table 2.1, while some of the tags marking syntactic category are listed in table 2.2.[5]

As seen from the analysis in (2.2.2), the tagger leaves a total of *seven* possible base forms for the components of sentence (2.2.1). The ambiguities in the analysis is the result of a second possible reading of the sentence, as presented in (2.2.3) below. Note that the Norwegian word *en* can be either a pronoun (one) or a determiner (a).

---

[4]The sentence was parsed 28/8-2003 with the web interfaced version of the tagger located at http://decentius.hit.uib.no:8005/cl/cgp/test.html.

[5]Note that the syntactic tags have a prefixed "commercial at" (@), while an arrow (< or >) indicates the direction of the head that the tagged word modifies. More thorough documentation of the complete tag sets can be found in Johannessen (1998).

(2.2.3)	*En*	*programmerer*	*programmerte*	*programmer.*
	One	programs	programmed	programs.

‘One programs programmed programs.’

Tag	Description
@DET>	Right-modifying determiner.
@<P-UTFYLL	Left-modifying prepositional complement.
@SUBST>	Right-modifying noun.
@ADJ>	Right-modifying adjective.
@ADV	Adverbial
@FV	Finite verb.
@SUBJ	Subject.
@OBJ	Object.
@I-OBJ	Indirect object.
@IV	Non-finite verb.
@KON	Conjunction.
@LØS-NP	NP with no syntactic function.

Table 2.1: Tags Indicating Syntactic Functions.

Tag	Description
det	Determiner
adj	Adjective
subst	Noun
verb	Verb
prep	Preposition
pron	Pronoun
sbu	Subjunction
inf-merke	Infinitival ‘to’
konj	Conjunction
clb	Clause boundary

Table 2.2: Tags Indicating Syntactic Category.

Consider the final noun *programmer* (programs) of sentence (2.2.2). This polysemous form can be used with several and quite distinct meanings, as in the sense of a radio or television show, a sequence of coded instructions for a computer, or an announcement of events, to name a few. In much the same way that a dictionary might give separate entries for different meanings of a word, one might argue that these variations constitute distinct semantic lemmata. But as seen in (2.2.2), such semantic distinctions are ignored under “syntactic lemmatization”, and the word form is assigned a single base form *program* (program). This means that possible senses of a polysemous or even homonymous form, with respect to the same syntactic category, are conflated in the same lemma.

Word Identification in Spartan The word extraction in this project is implemented through Spartan, which ultimately rests on the identifications done by the Oslo-Bergen Tagger. When Spartan records words from tagged text, its attention is mainly restricted to the tags indicating syntactic function and the morphosyntactic tags indicating part-of-speech, as shown in tables 2.1 and 2.2. In the particular case of (2.2.2) there are no alternative lemmas of the same syntactic category suggested for any of the surface forms. When more than one lemma is offered however, i.e. we get more than one base form marked with the same POS, we simply include them all. This situation can occur when, as previously mentioned, an inflected word form is homonymous and corresponds to several different base forms. Consider the Norwegian sentence of (2.2.4) below. This sentence as multiple possible readings, two of which correspond to the translations given in (2.2.4a) and (2.2.4b). The Norwegian noun form *tanken* can correspond to the base form *tanke*, which can be translated to *thought*, or *tank*, which is ambiguous between the sense of a water tank and an army tank.

(2.2.4a) *Han orket ikke tanken.*
He stand not thought-the.
'He could not stand the thought.'

(2.2.4b) *Han orket ikke tanken.*
He stand not tank-the.
'He could not stand the tank.'

Two base forms, both nouns, are suggested for the homonymous form *tanken*; *tank* (tank) and *tanken* (thought). In this case Spartan extracts two lemmas, with the same POS, for a single surface form.

If the base forms and syntactic categories are both identical however, we only keep one of the suggested lemmas. As an example, consider the word form *aviser* (newspapers). The tagger assigns the noun analysis of *aviser* two separate lemma entries with the base form *avis* (newspaper), identical in all respects except for gender, – one being marked as masculine while the other is feminine. When parallel entries, such as those for *avis*, diverge only in features other than POS and syntactic function, the distinction is ignored and the identical base forms are treated as, well, identical. Only one base form or lemma is recorded in this case.

2.2.1 Conflation of Senses

By the analysis outlined above, a word corresponds to a base form associated with a syntactic category or part-of-speech. All surface forms or tokens with the same possible base form and the same POS, as analyzed by the tagger, and regardless of homonymous or polysemous status, will be mapped to the same type, – the same *word*. A consequence of identifying words in this manner, is that the distributional analysis of a given word is done with respect to the sum of observed word tokens, as opposed to the distinct word senses associated with these tokens. Every use of the noun *program* in all its different senses, will be summed together in a single distributional signature.

As noted by Resnik (1993) the distributional hypothesis "makes the most sense when taken at the level of word meanings or uses rather than word tokens." But as Resnik (1993) further remarks, to propose a distributional analysis of word senses rather than tokens, would be "circular under the assumption that the set of word senses is itself defined by analyzing how word tokens are distributed".

The task described (Resnik, 1993) is that of capturing word similarity. Our task is that of inducing word senses by way of clustering. The problem of sense conflation provides ample motivation for the use of a *soft* or *fuzzy* clustering scheme. Our basic problem is that of mapping contextual representations of words into categories corresponding to word senses. But since each word representation may in fact be a fusion of various senses, several such mappings may be needed in order to adequately characterize a word. The disparities of hard and soft clustering are presented in chapter 4.

2.2.2 Vocabulary Size

Morphological normalization by way of stemming and lemmatization effectively reduces the vocabulary size of a given text sample (see Manning and Schütze, 1999, ch. 6). Since all the variant forms and tokens are reduced to common lemmata or stems, the overall number of individual word types is greatly reduced. It thereby contributes to alleviate part of the "sparse data problem"; the unique constituents we need to keep track of will be fewer in number, and each of them might be more frequently observed. A phenomenon pulling in the opposite direction however, is the system of *compounding* that we find in Norwegian. The three units of a noun phrase such as *hard disk error*, would in the Norwegian equivalent be appended to form the single constituent *harddiskfeil*, as seen in example (2.2.5). This productive mechanism yields more unique words, and contributes instead to further "sparsify" the data.

(2.2.5) *harddiskfeil*
hard-disk-error
'hard disk error'

In order to get more data and make more fine-grained and accurate predictions, an idea for a future improvement might be to add a level of tokenization when recording word occurrences so that for instance *feil* (error) in (2.2.5) would be seen as a separate form, but this would have to include some analysis of compund words.

2.3 Shallow Processing

In order to get our hands on the words that we want to describe, and the cotextual features with which to describe them, two elements are required; a large body of tagged text and a way to process it. This section describes our source, – the corpora of text tagged by the Oslo-Bergen Tagger, and outlines some tools implemented to mine this source for distributional information on words. Section 2.3.2 describes Spartan, a shallow parser, while a windowing tool is presented in 2.3.3. (Appendix A gives an overview of where to locate the code for the various components of the shallow processing, in the source files that accompany this paper.)

2.3.1 Corpora

Two different corpora of written Norwegian[6] texts are used in this project, both analysed and annotated by the Oslo-Bergen Tagger. One is the The Oslo Corpus developed by the Text Laboratory, and the other is a corpus currently under development at the Section for Norwegian Lexicography and Dialectology at UiO (hereafter referred to as the SNLD Corpus). The Oslo Corpus contains about 18.5 million words and consists of texts from newspapers and magazines (9.6 mill. words), factual prose (7.1 mill. words), and fiction (1.7 mill).[7]

The SNLD Corpus is an ongoing project led by Ruth Vatvedt Fjeld at UiO, with the goal of constructing a more balanced and representative collection of contemporary Norwegian text. The final corpus is intended to comprise 20 million words, of which nearly 10 million have so far been assembled. However, due to some overlap between the sources of the two corpora, only 4 million words of the SNLD Corpus were used in this project. (Whenever the term "the corpora" is used without further qualifications in this paper, it is meant to refer to the Oslo Corpus and SNLD Corpus.)

The texts are analyzed with a sort of dependency grammar, and words are marked with functional tags indicating heads and modifiers. As noted in section 2.2 where we presented a subset of the tags used to label words in the corpora, an arrow points in the direction of the head of which a word is marked as a modifier. Consider the analysis of sentence (2.3.1) in (2.3.2).[8]

(2.3.1)	*Kunden*	*bestilte*	*den*	*mest*	*eksklusive*	*vinen*	*på*	*menyen.*
	Customer-the	ordered	the	most	exclusive	wine	on	menu-the.

'The customer ordered the most exclusive wine on the menu.'

[6]There exist two official written forms of the Norwegian language; Bokmål, which is the most widely used variant, and Nynorsk. All texts used in the project are in the Bokmål form.

[7]All numbers pertain to the *bokmål* part of the corpus. There is also an additional *nynorsk* part containing 3.8 million words.

[8]As elsewhere throughout this paper, for clarity and ease of exposition we only include a few selected tags when displaying the analysis. For a description of the tags, please refer to table 2.1 and 2.2 of section 2.2. Sentence (2.3.1) was tagged by the web interfaced version of the Oslo-Bergen Tagger at 31/8-2003.

(2.3.2)
```
"<Kunden>"
        "kunde"     subst @obj  @subj
"<bestilte>"
        "bestille"  verb  @fv
"<den>"
        "den"       det   @det>
"<mest>"
        "mye"       adj   @adv>
"<eksklusive>"
        "eksklusiv" adj   @adj>
"<vinen>"
        "vin"       subst @obj  @subj
"<på>"
        "på"        prep  @adv
"<menyen>"
        "meny"      subst @<p-utfyll
"<.>"
        "$." clb <<< <punkt>
```

We see that multiple functional tags are often assigned in ambiguous cases[9], and there are no definite indications of attachments, phrases or trees connecting the words. The task of Spartan is thus to attempt to recover such relations on the basis of the syntactic functions.

2.3.2 Spartan

In shallow parsing, as opposed to full parsing, the objective is not to construct a complete parse tree spanning the entire sentence. Rather, a shallow parser only attempts to do a partial analysis and find "local trees". The aim is at isolating chunks and phrases, and detecting basic head/modifier relations. Because the parse is not intended to be complete, difficult cases may be left open and the parser can afford to be more conservative when making decisions. The approach is non-committal and many constituents and indicated relationships are left unattached or unresolved. Furthermore, since we are only interested in the distribution of *nouns*, the shallow analysis is only geared towards finding phrases and relations involving nouns or NPs.

Spartan consists of three layered modules;

1. a set of regular expressions (reg-exps)
2. a set of utility functions and search tools
3. a set of relational rules

The first two parts consist of general tools for processing the corpora. They are used for searching and navigating the tagged text. The set of regular expressions determine how the different tags, markers, delimiters, words etc. are

[9]Note that the multiple assignments of the `@obj` and `@subj` tags in (2.3.2) is the result of a possible topicalized reading, with an inversion of the subject and the object positions.

to be identified, while the search functions typically use the "reg-exps" when looking for specific constituents in the text. Examples of such tasks can include directed search for a word labeled with a particular syntactic tag, recognizing phrase boundaries for delimiting a sentence, and so on.

Regular Expressions The reg-exps are built in successive layers, where small basic units are joined together to make larger and more complex terms. As was shown in example (2.3.2), a noun marked as a possible object by the tagger might look like this;

(2.3.3) `"vin"       subst @obj  @subj`

To give an example of a regular expression in Spartan and how it is built up from smaller pieces through interpolation, (2.3.4) shows the definition of the pattern[10] that would match a noun such as (2.3.3) and return its base form. The top expression, (2.3.4a), defines the pattern of a noun marked as a subject. This expression is further composed of smaller patterns; one describing a general noun pattern and one for identifying the subject tag. The patterns may then further branch out to different sub-expressions that recognize the increasingly smaller bits and pieces that together make up the complete constituent that we are trying to describe. The technicalities and details of the patterns displayed here is not important, and their individual purpose should be fairly evident from the names of the variables that hold them. Note however, that pattern (2.3.4c) matches and *captures* what the tagger has analyzed to be a base or canonical form of the word. If a successful match is made for the pattern (2.3.4a) as a whole, then the base form captured by (2.3.4c) is returned.

(2.3.4a) `$NOUN_SUBJ_PAT = qr/$NOUN_PAT.*$SUBJ/;`

(2.3.4b) `$NOUN_PAT = qr/$LEMMA\ssubst\s/;`

(2.3.4c) `$LEMMA =  qr/^\s+\"\*?($LETTER*)\"/;`

(2.3.4d) `$LETTER = qr/[*.\s\d\w-]/;`

(2.3.4e) `$SUBJ = qr/\@subj/i;`

Although this sort of layered structure can quickly result in a horrendously long list of terms, it facilitates easy updates, maintainability and reuse of code. This is also important for compatibility between different versions of the tagger. Whenever a small addition or change is done with respect to the tags or format used by the tagger, we can just update the relevant pattern(s). The alternative of course, would be to instead rewrite all the affected rules, functions and higher-level patterns separately.

Rules The *rule* module defines the syntactic constructions and grammatical relations that we wish to recover from the corpora. The rules build on the "reg-exps" and the search tools when declaring the structure of the sub-trees they describe. Each rule works on a buffer with a sentence-delimited chunk

[10] Spartan is written in Perl, and the reg-exp examples are here shown in the form of Perl code.

of tagged text. When a rule has been successfully applied, the output is a n-tuple consisting of a relation name followed by the words that participate in the observed relation. The general form of the n-tuples produced by the rules is <`relation-type word`$_1$ ... `word`$_{n-1}$ >, where `word`$_{n-1}$ is always a noun. As an examle, consider the rule `noun_subj_verb`, which tries to find a verb and a preposed noun subject. If applied to a buffer containing the sentence of (2.3.2), it would report the <V_S_N bestille kunde>, meaning that *kunde* (customer) is found to be the subject of *bestille* (order). The implementation of this rule, based on the regular experessions and search tools mentioned above, can be paraphrased as in (2.3.5);

(2.3.5) *verb — noun subject rule;*
start from the initial position of the sentence buffer and look to the right for a non-auxiliary verb;
if successful match, look to the left for a noun marked as subject;
if successful match, look for constituents between the noun and the verb that can make the analysis less certain, e.g. another non-noun subject, a clause boundary, etc;
if unsuccessful match, report the relation for the corresponding lemmata;
try again with the verb position as initial buffer position.

One by one the rules traverse the buffer, trying to recognize a particular grammatical relationship. The rules have no memory, – chart-like or otherwise. The different rules work independently of each-other, and they are completely oblivious to the structures or constituents identified by any other rule. Although clearly not the most principled route to follow, it provides a fast way to arrive at working prototype. It also has the property of making the application order of the rules irrelevant, and rules can easily be added or removed. The execution is not unreasonably slow either, – it takes less than thirty minutes to process the entire corpora, yielding a total of nearly 5.7 million grammatical relations.[11] Table 2.3 below shows the relations output by Spartan for sentence (2.3.2).

Relation Type	word$_1$	word$_2$	word$_3$
`V_O_N`	`bestille` (order)	`vin` (wine)	
`V_S_N`	`bestille` (order)	`kunde`(customer)	
`A_mod_N`	`eksklusiv`(exclusive)	`vin` (wine)	
`N_prep_N`	`vin` (wine)	`på` (on)	`meny` (menu)

Table 2.3: Relations found by Spartan for sentence (2.3.2).

Word Features

The next step towards getting the data that we need is to transform the relational n-tuples into atomic *features*. The n-tuples of syntactic relations and grammatical dependencies are turned into features or attributes of individual words.

[11] 3.9 million relations were extracted from the Oslo Corpus, while 1.8 million were found in the SNLD Corpus.

If we again use an example drawn from the parse of sentence (2.3.2), we see that the relations reported by the rules are on a form such as <A_mod_N exclusive wine>. This relation can be informally said to express the event of an adjective, *eksklusiv* (exclusive), modifying a noun, *vin* (wine). If we change our perspective and focus on the individual words that participate in this event, we can say that it is an attribute of *eksklusiv* that it modifies *vin*, while *vin* has the feature of being modified by *eksklusiv*. Since our concern here is solely to describe nouns, most n-tuples only give rise to a single feature, – that of the dependent noun. So in this particular example above, the noun *vin* is assigned the feature `(adj_mod_by eksklusiv)`. If however, the n-tuple represents a construction that involves two nouns, such as a noun phrase containing a prepositional phrase with a noun complement, then two features are extracted; one indicating the attribute of modifying something, and another corresponding to the attribute of being modified. This means that the final set of features is far larger than the set of relational n-tuples. A complete list of the types of word features that are used is shown in table 2.4 (note that all features pertain to nouns). To be

Feature Types	Description
`subj_of`	Verbal subject
`obj_of`	Verbal direct object
`ind_obj_of`	Verbal indirect object
`prep_obj_of`	Verbal prepositional object
`pp_mod_of`	Modifying a noun complement in a PP
`pp_mod_by`	Modified by a noun specifier in a PP
`poss_of`	Specifier of possessive construction
`poss_by`	Complement of possessive construction
`adj_mod_by`	Adjective modification
`noun_mod_of`	Modifying a noun
`noun_mod_by`	Modified by a noun
`noun_con`	Conjunction

Table 2.4: Local Context Feature Types

precise, a *feature* is a pair consisting of a feature *type*, as one of those displayed in table 2.4, together with the second lexical *constituent* that takes part of the relation. Table 2.5 shows the complete set of local context features extracted by Spartan for words in example (2.3.1) and as tagged in (2.3.2).

Target	Feature
kunde (customer)	`SUBJ_OF bestille` (order)
vin (wine)	`OBJ_OF bestille` (order)
vin (wine)	`ADJ_MOD_BY eksklusiv` (exclusive)
vin (wine)	`PP_MOD_BY meny` (menu)
meny (menu)	`PP_MOD_OF vin` (wine)

Table 2.5: Local Context of Words in Example (2.3.2)

Most of the contextual feature types listed in table 2.4 should be pretty self-explanatory, as also goes for the rules and relations that give rise to them. A few however, could probably do with some further elaboration and justification.

Noun–Noun Modification Some of the included features pertain to particularities of Norwegian constructions, such as the features `noun_mod_of` and `noun_mod_by`. These features cover constructions such as

(2.3.6) *et glass vin*
a glass wine
'a glass of wine'

In this example the tagger labels *glass* with the syntactic function `@subst>`, and on the basis of this we assign *vin* the feature `(noun_mod_by glass)`, while *glass* is attributed `(noun_mod_of vin)`.

Conjunction Another feature that might be seen to stand a bit to the side of the others, is that of `noun_con`. This attribute is used to describe nouns combined by conjunction, such as *øl og vin* (beer and wine), and does not fit as comfortably into a dependency-like relational pattern as the other features. But the conjunction constructions are nontheless excellent indicators of semantic similarity, and are therefore included as part of the local context.

Prepositional Phrases In relation to the analysis of sentence (2.3.2), we saw the feature pair `pp_mod_of` and `pp_mod_by`, which also might need some further explanation. These features are derived from the relation type `N_prep_N`, which is assigned certain constructions of the form noun–preposition–noun. An example of this is seen in the relations and features extracted for the tagged sentence in (2.3.2). The triplet `<N_prep_N vin på meny>` (`<N_prep_N wine on menu>`) yields the feature `(pp_mod_by meny)` for *vin* (wine) and the feature `(pp_mod_of vin)` for *meny* (menu). Note that when converting the "prepositional triplet" to features, the prepostion *på* (on) is here simply thrown away. The particular choice of preposition seems to some extent arbitrary, and is at least not a very discerning attribute for our current purposes.

As seen from (2.3.2), the tagger itself makes no commitments regarding the attachments of prepositional phrases. Each and every preposition is labeled as a possible adverb (@adv), but otherwise left as a dangler. The attachment of the preposition and its possible noun complement (marked by @<p-utfyll) is left unresolved. When reporting the `N_prep_N` relation, Spartan applies some simple heuristics for deciding whether it is likely that the construction is actually a prepositional phrase embedded in a noun phrase. The heuristics include simple conditions such as; there is no verb immediately preceding the candidate head noun, or this verb is not labeled as taking a particle or adverbial complement,[12] or the verb is separated from the head by a tag marking a clause boundary of some sort or a subjunction, and so forth.

[12] The verb codes of NorKompLeks (Norwegian Computational Lexicon) indicating subcategorization or argument structure are used by the Oslo-Bergen Tagger. The codes "part" or "adv" (abbreviated to "pa" and "a" by the tagger) are attached to verbs taking possible particle and adverbial complements.

Prepositional Objects Another feature type that might appear to be somewhat obscure, is that of “prepositional objects”, represented by the feature `prep_obj_of`. This feature type is used for describing nouns in constructions of the form verb–preposition–noun. Verbs which are immediately followed by an adverbial particle or preposition are treated as “two-part verbs” by Spartan. Consider the sentence of (2.3.7) below, and the corresponding analysis of the tagger shown in (2.3.8);

(2.3.7)	*Han*	*slo*	*opp*	*nummeret*	*i*	*katalogen.*
	He	hit	up	number-the	in	catalogue-the

‘He looked up the number in the catalogue.’

(2.3.8)
```
"<Han>"
        "han"      pron  @subj
"<slo>"
        "slå"      verb  @fv
"<opp>"
        "opp"      prep  @adv
"<nummeret>"
        "nummer"   subst @<p-utfyll
"<i>"
        "i"        prep  @adv
"<katalogen>"
        "katalog"  subst @<p-utfyll
"<.>"
        "$." clb <<< <punkt>
```

As previously pointed out, the tagger leaves the question of prepositional binding undecided. The noun *nummer* (number) is labeled as the possible complement (`@<p-utfyll`) of a preposition whose attachment is unresolved. In this case, the particle *opp* (up) is appended to the verb component to yield the compound unit *slå_opp* (look_up). The noun *nummer* (number) is regarded as the object of this unit, and assigned the feature `(prep_obj_of slå_opp)`.

Many of the verbs treated in this way will not be what one typically considers to be a phrasal verb or a particle verb. Nevertheless, the prepositions following the verbs often seem to offer very relevant semantic cues. Separating different instances of the same verb occurring with different prepositions by regarding the pair as a single unit, often gives intuitively meaningful distinctions. Consider the different instances of the verb *snakke* (talk), and the various particles or prepositions that they appear with, in the examples of table 2.3.2. Next to each sentence we see the relevant target noun and the verb-preposition pair that is part of the feature of type `prep_obj_of`.

We see that various combinations of verbs and prepositions in these give rise to different compounded units and thereby different features for the nouns involved, and it seems reasonable to expect that things that are being talked *with*, talked *about* or talked *at*, might be quite disparate kinds of things.

An obvious question at this point perhaps, is why there is no feature named `prep_subj_of`. The first reason for this is practical. What we have here treated as particle verb constructions or phrasal verb, are not tagged as such in the

Sentence	Target	(prep_obj_of ...)
Arthur snakket til forsamlingen.	*forsamling*	snakke_til
Arthur talked to the crowd.	crowd	talk_to
Arthur snakket på møtet.	*møte*	snakke_på
Arthur talked at the meeting.	meeting	talk_at
Arthur snakket med en venn.	*venn*	snakke_med
Arthur talked with a friend.	friend	talk_with
Arthur snakket om livet.	*liv*	snakke_om
Arthur talked about life.	life	talk_about

corpora. The nouns following such constructions do not have any object tag associated with them. The preposed subject on the other hand, is marked with the subject tag. When Spartan tries to find verb related dependencies, this is primarily done on the basis of the tags indicating subjects, objects and direct objects. The feature type `prep_obj_of` is an attempt to include information that would otherwise be lost, since no object tag is given. Another reason for the omission of a subject converse `prep_obj_of`, is that the "phrasal" treatment of verbs and prepositions does not represent the same gains when the goal is to semantically characterize nouns in subject position. Consider again the examples in 2.3.2. They are all instances of the act of talking. The variations in meaning, that stem from the use of different prepositions, seem to apply to the objects to a much stronger degree than the subjects. Arthur performs the act of talking in much the same way in all these situations. The same situation seems to arise with many similar constructions. Examples of the opposite, where the particle in a proper phrasal verb gives strong semantic implications about the subject, is of course not hard to find. The first example that we gave of this type of construction, sentence (2.3.7), is indeed such a case. To be sure, the inclusion of the `prep_obj_of` feature does not represent an attempt to deal with phrasal verbs in any principled way. The case of "three-part verbs", or even separable "two-part verbs", is ignored with even respect to the "objects".

Noise vs. Signal Many of the verb–preposition–noun constellations covered by the rule governing the `prep_obj_of` feature, will undoubtly receive a misguided analysis. This probably holds true for quite a few of the other features reported by Spartan also. This leads us over to a very important point, which applies to pretty much every piece of information quenched out of the shallow processing step. The majority of events will be observed only one or a few times. What remains after the subsequent process of feature selection, are only observations which stand out for having occurred many times. Hopefully this will contribute to secure the reliability of the extracted patterns. From the pool of millions of extracted features, only a bucketful of the most frequent ones will actually be used in the final noun representations. Most of the noise and errors, unless highly systematic and consistent, will then be sifted out simply by virtue of their low-frequency or non-uniform appearance. In the case of verb–

preposition constructions, this might even prove a way of determining which of the observed combinations actually represent genuine two-part verbs.

With the help of Lars Nygaard from the Text Laboratory at UiO, the results of processing the Oslo-Bergen Corpus and the SNLD Corpus with Spartan, were used for setting up a data base of noun contexts comprising over 8 million feature counts. These features are scattered over more than 310000 unique nouns, of which nearly 30% are hapax legomena.[13]

Normalization

In any sufficiently large annotated collection of texts, there will unavoidably remain some level of noise in the form of non-alphanumeric characters, non-words, misspellings, mistaggings, and other unpredictable oddities of various shapes and forms. Noise filtering, standardization and various issues that can collectively be termed *normalization*, are therefore important parts of the sort of shallow processing that we deal with in this chapter.

Table 2.6 gives a few examples of how some such issues are dealt with in Spartan. We see, for instance, that white space characters in constituents that the tagger has recognized as multi-word units are replaced with an underscore. Alphabetic case is ignored and converted to agree.

In relation to numbers, Spartan replaces ordinal units, which are labeled as *ordenstall* (ordinal numbers) by the tagger, with `ORDO`, while most other numbers are replaced with `NUM`. For our purposes, the interesting thing to note about two occurrences such as *19.40* and *19.45*, is not the difference of five minutes, but the similarity of format which is typically used when stating the time of the day. Instead of trying to anticipate every format of strings denoting such things as dates and times, we just convert all numeric occurrences to the symbol `NUM`, and common and recurring patterns will then manifest themselves automatically.

Tokens	Normalization
17., *syttende* (17th, seventeenth)	ORDO
4%, *23%*, *90%*	NUM%
1-åring, *103-åring* (1-year-old)	NUM-åring
50-tallet, *1700-tallet* (50's, 17th century)	NUM-tallet
23.30, *07.55*	NUM.NUM
7/10-03, *10/7-2003*	NUM/NUM-NUM
08.22.2003, *22.08.03*	NUM.NUM.NUM
*arthur Arthur *arthur*	arthur
TV 2	tv_2
i ferd med (about to)	i_ferd_med

Table 2.6: Normalization in Spartan

[13]A hapax legomenon is a word or construction that has only one observed occurrence.

2.3.3 Conwin

Although only local context features are employed in the noun clustering that we describe later on, this project also provides a windowing tool, – Conwin, in order to make representations based on topical context possible. The broader kind of contexts extracted by Conwin, might be used to supplement the local features produced by Spartan.

Just as the Spartan rules, the Conwin module is built on top of the regular expressions and search utilities described in section 2.3.2. Conwin extracts content words from a frame delimited by a number of sentences specified by the user. Lemmatized word forms of the syntactic categories verbs, nouns, adjectives and adverbs, are extracted, while other closed-class words are ignored.[14] The context is delineated as being all content words occurring within n number of sentences surrounding the target word, where n defaults to 3. If $n = 1$, only words from the current sentence are included. Even-valued specifications of n extend the sentence span in the “leftward” direction, odd-valued specifications expand to the right; for $n = 2$ it includes the preceding and the current sentence; for $n = 3$ it includes the preceding, the current and the following sentence, and so forth. Table (2.7) shows the contexts extracted by Conwin for the words in the tagged sentence of example (2.3.2) with $n = 1$. Both Spartan and Conwin

Target	Context
kunde	`bestille eksklusiv vin meny`
bestille	`kunde eksklusiv vin meny`
eksklusiv	`kunde bestille vin meny`
vin	`kunde bestille eksklusiv meny`
meny	`kunde bestille eksklusiv vin`

Table 2.7: Conwin: Window Contexs of Words in Example (2.3.2).

take text analysed by the Oslo-Bergen Tagger as input, and they can optionally be requested to assign a running id number to each processed sentence. This sentence id is reported together with the features for each target word, and the two context types might thereby be aligned and traced.

[14]It is of course a gross oversimplification to sort every word of these four syntactic categories as “content words”, while brushing off all words from any other POS as semantically “empty”. Not all nouns and verbs are substantially content bearing, and prepositions are not by definition “meaningless”. This commonly used coarse division of things nevertheless provides a convenient and effective approximation.

Chapter 3

Semantic Spaces

This chapter describes the model used to represent the set of nouns that we want to cluster. The nouns will be characterized by way of their co-occurrences with the local contexts that we described in the previous chapter. In order to quantify the distributional similarity of words, we need to give them a numerical representation. This chapter presents the *vector space model* that we use for numerically representing the contextual patterns of nouns. A vector space is defined by a system of n coordinates where objects are represented as real valued vectors in the space $\Re^n$. In our case, the coordinates correspond to contextual features, while the objects are nouns.

Let $T = \{t_1 \ldots, t_k\}$ be the set of k nouns that we want to describe. The nouns are characterized on the basis of their co-occurrence frequencies with n contextual features of a set C. Each noun $t_i \in T$ is represented by a n-dimensional feature vector $\mathbf{f}_i = \langle \mathbf{f}_{i1}, \ldots, \mathbf{f}_{in} \rangle$. Each dimension $0 < j \leq n$ corresponds to a local context $c_j \in C$. We will use $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$ to denote the set of k noun objects representing T, where an element $\mathbf{f}_{ij}$ gives the value of the jth contextual feature in $\mathbf{f}_i$. The value of each dimension in a word vector is given by (a function of) the number of times the word has been observed to appear in the corresponding context. If the feature $c_j =$ `(obj_of bake)` and $\mathbf{f}_i$ represents the noun $t_i = cake$, then $\mathbf{f}_{ij} = 8$ means that *cake* has been observed as the direct object of the verb *bake* 8 times.

The set of feature vectors can also conveniently be represented as a word-by-context table or matrix, where each row describes a noun and each column describes a feature. We will sometimes treat $\mathbf{F}$ as such a $k \times n$ feature matrix, where the columns represent contexts and the ith table row corresponds to the feature vector $\mathbf{f}_i$. The concepts of a matrix and a set of vectors will often be used interchangeably throughout this report.

The n contextual features form the axis of an abstract feature space in which each noun object is represented as a vector or a point. The points are positioned according to their values along the various contextual dimensions. Representing words as points in a vector space allows us to apply measures of geometrical distance. In section 1.1.1 we described the hypothesis that the semantic similarity of words may be reflected in their distributional similarity. Furthermore, in a vector space model the distributional similarity is expressed by some measure of *spatial proximity* of the feature vectors. In other words, semantic similarity is defined as proximity in space. Words with similar contextual distributions are represented by points that lie closer together in space. Words with dissimilar distributions fall farther apart. When coupled with the idea that contextual patterns of words are indicants of their semantic content, such word-context spaces are often called *semantic spaces*. For the purposes of this project, a semantic space can be formally defined as a triple $\langle \mathbf{F}, A, s \rangle$, corresponding to a *co-occurrence matrix*, an *association measure* and a *similarity function*. This way of defining a semantic space model may also be seen to resemble the approach taken by McDonald (1997).

The co-occurrence matrix $\mathbf{F}$ is a feature matrix, or a set of noun objects in feature space, as described above. The component values of $\mathbf{F}$ correspond to the raw count data of the observed co-occurrences of nouns and contexts. Section 3.1 looks at the process of feature selection for defining the dimension variables of $\mathbf{F}$.

The association measure A is a weighting function that maps each element $\mathbf{f}_{ij}$ of the feature vectors in $\mathbf{F}$ to a real value. We will use $\mathbf{X}$ to denote the result of $A(\mathbf{F})$. This means that $\mathbf{X}$ is a set of vectors where each $\mathbf{x}_i = \langle A(\mathbf{f}_{i1}), \ldots, A(\mathbf{f}_{in}) \rangle$. The weighting function will typically be a measure of association strength for a word–context pair, in the form of a statistical test of dependence. $A(\mathbf{f}_{ij})$ then represents the salience of the contextual feature c_j for the noun represented by $\mathbf{f}_i$. A can also be the identity function, for which $\mathbf{F} = \mathbf{X}$. Under a matrix interpretation we will sometimes refer to $\mathbf{X}$ as the *association matrix*, and when we later go on to define word similarity in the space it is done on the basis of $\mathbf{X}$ rather than directly on $\mathbf{F}$. The cluster analysis is also performed on $\mathbf{X}$. The measures of association or salience is the subject of section 3.2.

The proximity function $s : \mathbf{X} \times \mathbf{X} \to \Re$ defines the similarity of the word vectors. We already mentioned how semantic similarity, by way of distributional correspondence, can be seen as a function of distance in the semantic space. Section 3.3 presents some standard definitions of geometrical distance. We also construct a *similarity matrix* on the basis of s on $\mathbf{X}$, and see examples of rankings of the most similar words for various targets. Finally, in section 3.4, we move on to the related issue of how *semantic classes*, and the notion of *class memberships*, can be construed in a semantic space model. By discussing various general properties of concepts and semantic categories, we correspondingly specify the properties that a cluster model should be able to accommodate if we want to represent such classes.

3.1 The Co-occurrence Matrix

The first step in the construction of the semantic space is to initialize the co-occurrence matrix $\mathbf{F}$. A fundamental part of this process concerns the process of defining the n dimensions of the feature space. This is what defines the dimensionality of the space and the interpretation of the coordinates. In section 2.1 we made some important choices in this connection when defining the notion of a contextual feature. We decided that the nouns will be characterized by local contexts in the form of syntactic relations and dependencies, and not by, say, articles, documents, word windows, character n-grams, or other possible types of attributes that might reasonably constitute the context axis of a word–context space. But the problem of feature extraction remains. Recall that our context data base, resulting from the shallow processing described in section 2.3, contains over 8 million features. As this would yield an infeasible high number of coordinates, we must decide on a smaller set of dimension variables. This is an issue that has been given a lot of attention in research on semantic space models and in works on pattern recognition and statistical data analysis in general. A broad distinction can be made between approaches where the features undergo transformations and those in which they are merely selected. In the former type of approach, the issue of feature extraction can also be cast as a problem of reducing the number of dimensions in a space. In our case, however, the dimensions are defined through simple feature selection based on a frequency criterion. Before we select the actual context set however, we try to make this choice of strategy somewhat more tenable by briefly reviewing a few examples of the the two broad types of methods in the following sections.

3.1.1 Dimension Reduction

If we think of the space as initially constructed with every feature in store, or with a relatively high value for n, then our task is to reduce the dimensionality of the space to a manageable size. The reduction of dimensionality is often based on the assumption that phenomenons appearing complex and high-dimensional, may actually be governed by a few “hidden causes” or “latent variables” (see Carreira-Perpiñán, 1997). The task is to uncover these underlying variables.

There are various such techniques in use. One of the more widely used such techniques within the field of IR and semantic space research, is the method of latent semantic analysis (LSA), as advocated by Landauer and Dumais (1997). LSA aims at uncovering the inherent dimensionality of the full feature space through a technique based on singular value decomposition (SVD) of the initial frequency matrix. The resulting dimensions of the reduced space are linear combinations of the original ones.

Another alternative is the method of *random indexing* used by Sahlgren (2002; 2002). As with LSA, the resulting space has a dimensionality lower than the number of input features, but the technique of random indexing is more scalable and not as computational expensive. Under this procedure, the dimensionality of the space is fixed and there is no separate reduction step. Instead, every feature is assigned a distributed representation in the form of a random label: Each context is associated with a sparse index vector of constant length that holds a small number of randomly distributed -1 and +1 elements, with the remaining elements set to 0 (Sahlgren, 2001). When a target word is observed with a given context, the associated random label is added to the feature vector of the word. The final data matrix is given by adding up all such random index vectors corresponding to all observed co-occurrences.

Clustering as Dimension Reduction It is also worth noting that the process of clustering is actually a form of dimensionality reduction in its own right. If our original set of object data is instead seen as the feature set, the resulting clusters can be seen as the underlying variables. To illustrate this view of clustering as reduction, let us say that our final goal is actually to cluster *verbs* and that we want the groups reflect similarities of subcategorization. The verb features correspond to which nouns they take as arguments. After an initial pass of noun clustering, features that involve nouns of the same cluster can be mapped to the same dimension of the verb–argument space. Rather than recording co-occurrence information for *individual* noun arguments, we can record it instead for *classes* of nouns. In the same vein, before clustering the noun data set in our current project, we could analyze the transpose of **F**, a context–noun space, and use the resulting context clusters as our features.

Interpretability of the Dimensions In relation to spaces constructed within approaches such as LSA and Random Indexing, Sahlgren (2002) mentions a potential problem that also applies to dimension reduction through clustering, which is that no single dimension necessarily means something in particular. It may not always be clear what features a given variable represents or how it should be interpreted. The resulting models may thereby be theoretically opaque and the knowledge acquired through them may be difficult to inspect and assess (Sahlgren, 2002). This might be a particular disadvantage when working with contextual word features as we do. Much potential insight might be gained from being able to trace which dimensions, and thereby which local contexts, make two words similar, and which contexts constitute the most defining coordinates of a given class.

3.1.2 Feature Selection

In the initial and full-profile co-occurrence matrix on the other hand, what Sahlgren (2002) calls the "localist" frequency matrix, every dimension corresponds to a concrete feature. This property is preserved in the spaces that result from the second type of extraction methods that we mentioned. These procedures simply truncate the initial dimensions rather than induce new ones, – the features are selected rather than transformed. An example of such an approach is found in relation to the Hyperspace Analogue to Language (HAL) model of Lund and Burgess (1996). A reduced-dimension feature matrix is produced through analysis of *column variance*, discarding the features of lowest variance. Lowe and McDonald (2000) on the other hand, choose context words on the basis of a statistical criterion of *reliability*. An ANOVA method is used to measure the consistency of co-occurrence patterns across various partitions of a corpus. The contexts found to be most reliable are then selected to define the dimensions of the word space.

Lowe (2001) states the problem of feature selection as an instance of the bias-variance dilemma in statistical learning theory: There is a trade-off between selecting either low-frequency features that are highly indicative of content but come with unreliable statistical properties, *or* high-frequency features that yield reliable statistics but appear across the board and thereby have little discerning potential.

Levy and Bullinaria (2001) compared various such truncation techniques based on ranking the context words according to some ordering criterion, and then selecting the top n candidates. The contexts were ordered according to variance, reliability, or frequency. When evaluated for the tasks of synonym choice and semantic categorization, Levy and Bullinaria (2001) concluded that the former two criterions did not seem to confer any advantage over the simple frequency strategy. Levy and Bullinaria (2001) stress instead the importance of corpus size and including a number of features as high as feasible.

Feature Types The good results obtained in (Levy and Bullinaria, 2001) by simply using the most high-frequent context words as the feature set, is perhaps even more striking considering the fact that they also included closed class words, such as determiners and prepositions. Function words and closed class words, and other words that rank high according to frequency, are often regarded as being of little use in relation to semantic space models because of their uniform co-occurrence patterns.

There is an important difference concerning the feature types that we use within the project of this paper and the features used in the works cited above. While (Lowe, 2001; Lund and Burgess, 1996; Levy and Bullinaria, 2001; Sahlgren, 2002) all employ some form of word windows, our noun features are based on lexico-syntactic local contexts. As mentioned in 2.1, features based on word co-occurrences within grammatical relations, are probably more semantically indicative and discerning than features based on word co-occurrences alone. Consider the difference between the most frequent local contexts and the frequent words that participate in them. Table 3.1 lists some of the most frequent features extracted by Spartan from the Oslo Corpus and the SNLD Corpus. (A feature such as `(subj_of flabbergast)`, corresponding to the property of being the subject of the verb 'flabbergast', is listed as being composed of the type `subj_of` and the word `flabbergast`).

Rank	Feature	
	Type	Word
1	`adj_mod_by`	`ORDO` (any ordinal number)
2	`subj_of`	`si` (say)
3	`adj_mod_by`	`stor` (large, big, much)
4	`adj_mod_by`	`mange` (many)
5	`obj_of`	`gi` (give)
6	`adj_mod_by`	`ny` (new)
7	`subj_of`	`komme` (come)
8	`obj_of`	`ta` (take)
9	`subj_of`	`gi` (give)
10	`subj_of`	`gå` (go, walk)
991	`obj_of`	`drepe` (kill, murder)
992	`pp_mod_by`	`myndighet` (government, authority)
993	`noun_con`	`kontroll` (control)
994	`noun_con`	`mål` (goal, measure)
995	`subj_of`	`vende` (turn)
996	`adj_mod_by`	`øvre` (upper, high)
997	`pp_mod_of`	`manus` (manuscript)
998	`pp_mod_by`	`arbeide` (work)
999	`obj_of`	`hindre` (hinder, obstruct)
1000	`noun_con`	`ulempe` (disadvantage, inconvenience)

Table 3.1: Examples of the most frequent features extracted by Spartan from the Oslo Corpus and the SNLD Corpus, sorted by number of occurrences.

We see that the feature `(subj_of si)` is ranked as the second most frequent context. Compare the property of appearing in the subject position of the verb *si* (say), and the property of occurring together with the word *si* within a window of, say, ten words. The former property clearly seems to be more discriminating than the latter. Consider the feature ranked as the 991th most frequent in table 3.1. Being known to simply co-occur with the verb drepe (kill, murder) might be informative to a certain degree, but being its object is more telling. When syntactic and grammatical information is included, even the most frequent features seem to be more "semantically focused". When Levy and Bullinaria (2001) report good results with the simple frequency based approach applied to context window data, it seems all the more likely that this method should also be a viable alternative for our data set.

3.1.3 The Vocabulary Problem

In the word space experiments of Levy and Bullinaria (2001), the simple frequency based approach was also compared to an application of singular value decomposition as in LSA. The dimensionality reduction did not result in any improved performance on their particular evaluation tasks. Schütze and Silverstein (1997) notes that, in relation to information retrieval (IR), the technique of indexing by latent semantic analysis (LSI[1]) as used by Deerwester et al. (1990), is effective for dealing with the so-called *vocabulary problem*. This is actually the problem of synonymy and the fact that people use a high variety of words to express the same idea. The implication for IR is that similar documents that contain information about the same topic, may still use different sets of terms. If polysemy is a problem that reduces precision, synonymy can be seen as a problem that reduces recall.

An analogy of the vocabulary problem also carries over to our setting. Consider two target terms t_i and t_j. Let us say that the former has been observed to frequently occur with features involving the words *beer*, *bar*, and *booze*, while the latter has a high value for features with *lager*, *pub* and *liquor*. Let us say that the features are of the same type, for example `noun_con` denoting a conjunction. When comparing the feature vectors of t_i and t_j ($\mathbf{f}_i$ and $\mathbf{f}_j$), the two terms will not check out as similar. Through LSI or feature clustering, however, the respective features might have been mapped to the same dimension, and the vectors will be recognized as representing similar information. This is an important property when doing *similarity search*, where the task is to retrieve similar objects for a given query. Within IR, the task is then typically to retrieve documents that match a search query submitted by a user. In our semantic space $\langle \mathbf{F}, A, s \rangle$, the query might be thought of as a given target noun, and the task being to retrieve similar words. We will later see examples of such similarity search on $\langle \mathbf{F}, A, s \rangle$ in section 3.3.

In the case of clustering, however, which is our actual task at hand, Schütze and Silverstein (1997) notes that the potential problems of non-overlapping feature vectors are not equally pressing. When assigning an object to a cluster, it is not the distance of the object to another individual vector that settles the matter, but the sum of such distances. Cluster distance is measured between the feature vector and many individual vectors, or to a single centroid. We can think of distance expressed as a one-to-many rather than a one-to-one relation.

[1]LSI is a technique for representing documents as vectors in a latent semantic space rather than a term space.

Vectors encoding similar content may then be assigned the same clusters even if they do not have many individual features in common. This effect is perhaps most prominent in the case of centroid based clustering or re-assignment (Schütze and Silverstein, 1997), which are the sort of methods that are applied to the data set of this project. A targets distance from a cluster is then measured towards a summarizing prototype, – a single center point reflecting the averaged feature values of all members within the group. Since the clustering of our noun–context data follows such a centroid based approach, this might further indicate that we might do well without the extra step of dimensionality reduction.

3.1.4 The Noun–Context Data Set

In the last sections we have seen several points that seem to suggest that the simple frequency approach (as described in section 3.1.2) might be sufficient when defining the dimensions of our noun–context space. In order to select our final data set, all nouns and features recorded by the shallow processing described in chapter 2.3, are ranked according to the frequency of their observations. Before selecting the final features, however, we first remove the 50 most frequent contexts, as they seem overly general and uniformly distributed. We then define the feature set C as the n most frequent local contexts, with $n = 1000$. The set of excluded contexts mainly consists of subject and object relations with 'empty' and 'light' verbs, such as *gå* (go) *komme* (komme) *ta* (take), and modifications of general adjectives such as *god* (good) *ny* (new) and *ulik* (different). Due to the Zipfian distribution[2] of the co-occurrence data, frequency drops quickly in the top range of the list; the highest ranked feature has 52380 occurrences, while the feature at the 50th position has 7909 occurrences. The last of the contexts included in our feature set, at position 1050 of the frequency list, was observed exactly 1000 times.

The set of nouns T is selected in a similar way. T consists of the k most frequently observed nouns in the total data set, for $k = 3000$. We thus have a 3000×1000 co-occurrence matrix $\mathbf{F}$.

Note that the figures given for feature frequency above were relative to the entire data set. The most frequent context with respect to $\mathbf{F}$ has 6423 occurrences, while the least frequent has 330. Note also that $\sum_{j=1}^{n} \mathbf{f}_{ij}$ does not represent the total number of times that the noun t_i (represented by $\mathbf{f}_i$) has been uniquely observed, but the number of recorded local contexts that it has been observed in. A single sentence with a single occurrence of a given word, might spawn multiple co-occurrence counts for the word, corresponding to for instance adjectival modifications, verb–argument relations, and so forth. In other words, a single word token might participate in multiple local contexts simultaneously.

This also means that the notion of *co-occurrence* used in this report, is somewhat unusual. One typically talks of a co-occurrence pair (x, y) in the sense of two words w_1 and w_2 appearing together. The set of object variables

[2]Zipf's law, formulated by the Harvard linguist George Kingsley Zipf, states that for many frequency distributions, the relationship between the frequency of an event f and its rank r (according to frequency) obeys $f \propto \frac{1}{r}$ (Zipf, 1935). Since the distribution of words is observed to obey Zipf's law it is sometimes said to have an Zipfian distribution. This means that a language generally has a small number of very frequent words, an intermediate number of intermediately frequent words, and a large number of infrequent words (see Manning and Schütze, 1999, ch. 1).

and the set of dimension variables might then even coincide and consist of the same entities, recorded in a word-by-word and $k \times k$ co-occurrence matrix. The co-occurrences are also often *directional*, as for bi-grams, where w_1 linearly precedes w_2. The pair (w_1, w_2) is thus different from (w_2, w_1). In our case, however, a co-occurrence pair consists of a local context c and a noun t. Since a sense of directionality and structure is already encoded in the feature c, it makes little sense to differentiate (c, t) from (t, c). Although it might seem more appropriate to speak of a noun t occurring *in* a context c, rather than the pair t and c *co*-occurring, we often opt for the latter since it is more compact and forms part of a well-established terminology.

Tables 3.2 and 3.3 show examples of $t \in T$ and $c \in C$ throughout the frequency range (every 100th noun t and every 34th local context c is displayed – yielding 30 examples of each – sorted according to frequency).

Rank	Noun	Frequency
1	*år* (year)	13617
101	*ansikt* (face)	1805
201	*lønn* (salary, reward, maple)	1322
301	*gud* (god)	1148
401	*materiale* (material)	647
501	*mandat* (mandate)	637
601	*midt* (middle)	551
701	*etat* (department, service)	484
801	*institutt* (institute)	474
901	*aldersgruppe* (age bracket)	411
1001	*blad* (magazine, leaf)	396
1101	*arrangement* (arrangement)	301
1201	*beskyttelse* (protection, defence, cover)	288
1301	*nemnd* (committee, board)	285
1401	*utfall* (outcome, result)	277
1501	*dødsfall* (death, decease)	251
1601	*søkelys* (focus, spotlight)	198
1701	*oppsikt* (attention, sensation)	192
1801	*karna* (?)	183
1901	*fangst* (catch)	146
2001	*refleksjon* (reflection)	145
2101	*oljeselskap* (oil company)	137
2201	*oppbygging* (construction, composition)	124
2301	*tante* (aunt)	119
2401	*merknad* (note, remark, observation)	118
2501	*medlemsland* (membership countries)	116
2601	*vandring* (wandering, migration, travel)	108
2701	*opplag* (impression, stock, edition)	108
2801	*hell* (good luck, fortune, wane, inclination)	105
2901	*arbeidsdag* (work day)	105

Table 3.2: Examples of nouns drawn from the term set T, sorted according to frequency in the co-occurrence matrix $\mathbf{F}$.

	Context Feature		
Rank	Feature Type	Feature Word	Frequency
1	obj_of	gå (go, walk, run, leave, pass)	5598
35	prep_obj_of	se_på (watch, look_at)	3796
69	obj_of	omfatte (include, comprise)	2768
103	adj_mod_by	dårlig (bad, poor, ill)	2699
137	poss_by	utvalg (committee, range, selection)	2317
171	obj_of	slå (hit, strike, knock)	2312
205	obj_of	nå (reach, catch, make)	1820
239	adj_mod_by	nasjonal (national)	1778
273	adj_mod_by	vesentlig (essential, considerable)	1610
307	adj_mod_by	statlig (public, governmental)	1469
341	noun_con	mor (mother)	1306
375	adj_mod_by	rett (right, law, court, dish)	1255
409	subj_of	le (laugh)	1188
443	subj_of	lese (read)	1150
477	pp_mod_of	tiltak (effort, initiative, measure, precaution)	1133
511	noun_con	utdanning (education)	1132
545	adj_mod_by	hard (hard, tough, heavy)	1106
579	subj_of	forklare (explain)	968
613	subj_of	tilby (offer)	943
647	pp_mod_of	strid (battle, struggle, controversy, dispute)	860
681	obj_of	vekke (suggest, call, wake, excite)	850
715	adj_mod_by	elektrisk (electric)	849
749	pp_mod_of	tjeneste (service, favour)	779
783	adj_mod_by	britisk (British)	776
817	subj_of	uttale (express, state, pronounce)	748
851	pp_mod_of	ansikt (face)	680
885	obj_of	glemme (forget)	662
919	pp_mod_of	rolle (role)	598
953	pp_mod_of	vann (water)	591
987	pp_mod_of	undersøkelse (investigation, inquiry, examination)	584

Table 3.3: Examples of local contexts drawn from the feature set C, sorted according to frequency in the co-occurrence matrix $\mathbf{F}$.

3.2 The Association Matrix

The feature matrix **F** consists of co-occurrence counts of words and local contexts in corpora. But the raw frequencies alone may not always be very informative. Consider the word *vin* (wine) which has a total count of 1895 in our context data base, dispersed across 805 unique contexts. The noun has been observed as direct object of the verb *kjøpe* (buy) 14 times, and as the object of *helle* (pour) 8 times. If we use frequency as our ordering criterion, `(obj_of kjøpe)` is ranked as a more important and typical feature of *vin* than the feature `(obj_of helle)`. Intuitively however, the property of being bought does not seem to be as indicative of what it means to be "wine-like", as the property of being poured. While an impressive number and variety of things can be bought, only a few things can readily be poured.[3] Raw co-occurrence frequency then, is not a good criterion for relevance. Instead we need to weight our frequency counts by a function that reflects how *salient* a given context is in relation to a given target word. To this end we apply an *association measure*. Typically such measures are applied to lexical co-occurrence pairs in the form of n-grams, in order to identify *collocations*. In our case, however, we want to capture associations not as a relation between two target nouns or lexical co-occurrences, but in the sense of what Church and Hanks (1989) calls *lexico-syntactic co-occurrence constraints*.

In general terms, an association measure takes form of a statistical test for determining the degree of dependence between a pair of events. The basis of the tests is typically the *null hypothesis* that the occurrences of t and c are independent. We then compare, in some way or another, our actual count data to the results that we could expect by chance under the null hypothesis. The deviation corresponds to the degree of correlation.

Three different association measures are implemented[4] in this project; *mutual information*, the *log likelihood ratio*, and the *log odds ratio*. The weighting function A of $\langle \mathbf{F}, A, s \rangle$ applies such a measure to each component of the co-occurrence matrix **F**. The result is the association matrix **X**, reflecting salience scores instead of frequencies. But before we turn to describe the particular salience tests we first introduce some handy notation and formulate the empirical basis of the tests.

3.2.1 Data Representation

For the purpose of association testing, a given local context c and a noun t are treated as the outcome of the two-valued random variables C and T. The value variables correspond to the presence or absence of c and t respectively. Our observations can then be cross-classified with respect to the possible combinations of events, which can be set up in a standard two-way contingency table. We can think of each component $\mathbf{f}_{ij}$ of the feature vectors in **F** to give rise to their own 2-dimensional cross-classification table. This is illustrated in table 3.4 below, where the $k \times n$ co-occurrence matrix is collapsed into a 2×2 table for

[3] In the context data base resulting from the shallow processing step described in section 2.3.2, the feature `(obj_of kjøpe)` is listed with a total of 4146 occurrences with 1908 different combinations, while `(obj_of helle)` is observed 484 times with 320 unique nouns.

[4] Refer to appendix A to see where to find the code for the various association measures in the source files that accompanies this paper.

component $\mathbf{f}_{ij}$, where $t_i = t$ and $c_j = c$. The expression $f(c,t)$ refers to the

	$T = t$		$T \neq t$		
$C = c$	$f(c,t)$	$+$	$f(c,\neg t)$	$=$	$f(c)$
	$+$		$+$		
$C \neq c$	$f(\neg c,t)$	$+$	$f(\neg c,t)$	$=$	$f(\neg c)$
	$=$		$=$		
	$f(t)$		$f(\neg t)$		

Table 3.4: Contingency Table of Observed Frequencies

co-occurrence frequency of the pair t and c, $f(\neg t)$ refers to the frequency of any noun–context pair not involving t, and so forth. The marginal frequencies of c and t are given by the row sum $f(c)$ and the column sum $f(t)$ respectively. Let N be the total number of co-occurrences and the sum of the feature matrix $\mathbf{F}$. We see that $f(t) + f(\neg t) + f(c) + f(\neg c) = N$. Table 3.5 summarizes our data in the same way as table 3.4, but using a more flexible and compact notation similar to that of Pedersen (1996). The asterisks can be thought of as wild cards or as giving the row or column indices that are summed over. On the basis of the

	$T = t$		$T \neq t$		
$C = c$	$O_{(11)}$	$+$	$O_{(12)}$	$=$	$O_{(1*)}$
	$+$		$+$		
$C \neq c$	$O_{(21)}$	$+$	$O_{(22)}$	$=$	$O_{(2*)}$
	$=$		$=$		
	$O_{(*1)}$		$O_{(*2)}$		

Table 3.5: Contingency Table of Observed Frequencies

empirical counts in table 3.5, we can now compute the *expected* co-occurrence frequency of c and t under the (null hypothesis) assumption of *independence* as

$$E_{(ij)} = \frac{O_{(i*)}O_{(*j)}}{N} \tag{3.2.1}$$

Indices i and j of $E_{(ij)}$ correspond to the same levels and factors as in the cross-classification of table 3.5, so that the expected frequency of, say, the pair $(\neg c, t)$ is $E_{(21)} = (O_{(2*)}O_{(*1)})/N$.

The maximum likelihood estimates (MLE) for our observations are given in table 3.6. These are simply the distributions that maximize the probability of our data. We have now formulated all the prerequisites and can turn to the actual association measures.

3.2.2 Association Measures

Mutual Information A widely adopted basis for association weighting within the NLP-community is the information theoretic measure of mutual information (MI). Fano (1961) defined the notion of (pointwise) mutual information between two events c and t as in equation (3.2.3). The introduction of mutual information in computational linguistics is usually credited Church and Hanks (1989),

	$T = t$	$T \neq t$	
$C = c$	$P(t, c) = \frac{O_{(11)}}{N}$	$P(\neg t, c) = \frac{O_{(12)}}{N}$	$P(c) = \frac{O_{(1*)}}{N}$
$C \neq c$	$P(t, \neg c) = \frac{O_{(21)}}{N}$	$P(\neg c, \neg t) = \frac{O_{(22)}}{N}$	
	$P(t) = \frac{O_{(*1)}}{N}$		

Table 3.6: Maximum Likelihood Estimates for C and T

and has since been used in a number of works (Hindle, 1990; Brown et al., 1992; Pantel and Lin, 2002; Tugwell and Kilgarriff, 2001; Yarowsky, 1992).

$$I(c, t) = \log_2 \frac{P(c, t)}{P(c)P(t)} = \log_2 \frac{P(c|t)P(t)}{P(c)P(t)} \tag{3.2.2}$$

$$= \log_2 \frac{P(c|t)}{P(c)} = \tag{3.2.3}$$

$$= \log_2 \frac{O_{(11)}}{E_{(11)}} \tag{3.2.4}$$

The notion of mutual information can be interpreted in a number of different ways. $I(c, t)$ can be seen to express the amount of information, provided by the occurrence of c about the occurrence of t. Alternatively, it can be interpreted as the reduction of uncertainty about the occurrence of one event given the occurrence of the other (see Manning and Schütze, 1999, ch. 5).

Church and Hanks (1990) informally describes mutual information as comparing the probability of observing c and t together with the probabilities of observing c and t independently. As also seen from (3.2.2), mutual information of c and t is thus a likelihood ratio of their joint probability and the product of their marginal probabilities (see Manning and Schütze, 1999, ch. 5). In the case of perfect independence where the co-occurrence of c and t is as expected by chance, we have that $P(t, c) = P(t)P(c)$ and the mutual information is zero. In the case of perfect dependence on the other hand, c and t always occur together and $P(c, t) = P(c)$. The mutual information is then $I(c, t) = \log_2 P(c)/P(c)P(t) = \log_2 1/P(t)$. We see that the lower the marginal probability of seeing t, the larger the association score. This is the cause of the well know tendency of the mutual information measure to overestimate the correlation of rare events. This is especially an undesirable trait when working with corpus data because of the Zipfian distribution of words, which means that the majority of words will occur only very few times. In order to compensate for this frequency bias, MI-based association measures often include some sort of heuristic correction, such as squaring the numerator of equation (3.2.4) (see Evert, 2001). Pantel and Lin (2002) multiplies the entire score with the discounting factor

$$\frac{O_{(11)}}{O_{(11)}+1} \times \frac{\min\left(O_{(1*)}, O_{(*1)}\right)}{\min\left(O_{(1*)}, O_{(*1)}\right)+1} \tag{3.2.5}$$

Log Likelihood Ratio An alternative association measure designed to overcome some of these problems related to data sparseness, is the test statistic simply known as *log likelihood ratio*, $-2\log\lambda$, as advocated by Dunning (1993). Dunning (1993) writes of the log likelihood ratio, also know as the *G-score* or G^2, that it "allows the direct comparison of the significance of rare and common phenomena."

The basis of the G^2 test is the null hypothesis H_0 that the occurrences of t and c are independent. H_0 assumes that the probability of seeing c is unaffected by knowing whether $T = t$. Given our actual observations, we then compare the likelihood of this hypothesis to the likelihood of a hypothesis H_1 that instead assumes dependence. The likelihood ratio indicates how much more likely one hypothesis is over another (Manning and Schütze, 1999). The two different hypothetical population models are stated as

$$H_0 : p_0 = p(c|t) = p(c, \neg t) = p(c) \qquad \text{(independence)} \tag{3.2.6}$$

$$H_1 : p_1 = p(c|t) \neq p2 = p(c|\neg t) \qquad \text{(dependence)} \tag{3.2.7}$$

where the conditional probabilities of H_1 are estimated as $p_1 = \frac{O_{(11)}}{C_{(*1)}}$ and $p_2 = \frac{O_{(1*)}-O_{(11)}}{N-O_{(*1)}}$.

G^2 is a so-called goodness of fit statistic that measures the correspondence between our observed counts and the counts expected if the null hypothesis of independence is true (see Pedersen, 1996). The deviation corresponds to degree of correlation. Dunning's log likelihood ratio can be formulated in many different ways, but the most concise alternative is perhaps the the so-called entropy form given in (Evert, 2001) and (Pedersen, 1996);

$$-2\log\lambda = 2\sum_{ij} O_{(ij)} \log_2 \frac{O_{(ij)}}{E_{(ij)}} \tag{3.2.8}$$

If the assumption of independence is true, the quantity $-2\log\lambda$ is asymptotically approximated by the X^2 distribution (Dunning, 1993). Some of the claims about the ability of log likelihood to more accurately handle low frequency data however, are questioned by experiments in Evert and Krenn (2001). Dunning (1993) notes that G^2 also tends to overestimate independence.

Log Odds Ratio The final association measure that we will look into is the log odds ratio employed in the semantic space experiments of Lowe (2001). The odds ratio θ gives the ratio of the odds for some event to occur, where the odds themselves are also a ratio. Given the local context c, the odds of finding t rather than some other noun, can be stated as $P(c,t)/P(c,\neg t)$. Given instead that any other context than c is present, the chance of seeing t rather than some other noun, is $P(\neg c, t)/P(\neg c, \neg t)$. The ratio of these two odds indicates how much the chance of seeing t increase in the event of c being present.

When measured by the log odds ratio, the magnitude of association between a noun and a context is independent of their marginal probabilities. On the basis of the values in tables 3.5 and 3.6, the odds ratio is estimated as

$$\theta(c,t) = \frac{P(c,t)/P(c,\neg t)}{P(\neg c,t)/P(\neg c,\neg t)} \tag{3.2.9}$$

$$= \frac{O_{(11)}/O_{(12)}}{O_{(21)}/O_{(22)}} \tag{3.2.10}$$

$$= \frac{O_{(11)}O_{(22)}}{O_{(12)}O_{(21)}} \tag{3.2.11}$$

If the probability of seeing t increases when c is present, then $\theta(c,t) > 1$. If $\theta(c,t) = 1$ then c makes no difference to the probability of seeing t, which means that the noun and the context are distributionally independent. By taking the natural logarithm of the odds ratio, $\log \theta$, the score is made symmetric with 0 being the neutral value that indicates independence (Lowe and McDonald, 2000).

In all results reported in this paper the log odds ratio is used as the basis of the association measure A in the semantic space given by $\langle \mathbf{F}, A, s \rangle$. The weighting function A thus applies the $\log \theta$ test to every component of $\mathbf{F}$ as we described above, resulting in the association matrix $\mathbf{X}$. It is often difficult to evaluate the effect that a particular choice of A has on the overall model, except for by manual inspection of the salience scores. The results of $-2\log\lambda$ and $\log\theta$ seemed quite similar, and somewhat better than the results obtained with mutual information (with various additional weighting schemes to balance the frequency bias, such as the discounting factor in equation (3.2.5)). This judgment was made entirely on an intuitive basis by manually comparing lists of noun–context "collocation pairs" ranked according to association strength.

3.2.3 Negative Correlations

The case of *negatively correlated* pairs is usually ignored when measuring word–context associations. This is done both because negative associations are not considered a salient property of what we want to model (Lowe and McDonald, 2000) and due to unreliability of these estimates for sparse corpus data (Dagan et al., 1995). In the implementation of association weighting in this project, all such cases are explicitly adjusted; unobserved or negatively correlated co-occurrence pairs (c,t) are assumed to have zero association. The version of G^2 formulated in (3.2.8) is two-tailed in the sense that the direction of dependence as predicted by H_1 is not stated. Both negatively and positively correlated event pairs thus receive a high score. With respect to the hypothesis of dependence formulated in (3.2.7), we can filter out the negatively correlated pairs by checking if $p1 < p2$. The exact same pairs c and t result in $\log\theta(c,t) < 0$ and $I(c,t) < 0$. For all such negatively correlated pairs, we assign the context c a salience score of 0 for the noun t.

3.2.4 Local Truncation

By "zeroing out" the weights that correspond to negatively correlated events, we also obtain an effect similar to what Schütze and Silverstein (1997) describe as *local truncation*. This is simply the process of converting non-zero values of a vector to 0, and thereby locally projecting the feature vector onto a different subspace. In our case, the association weighting and truncation step is carried out simultaneously through setting negative correlations to 0. The association vectors in **X** thus have fewer non-zero elements than the co-occurrence vectors in **F**. The projection is "local" in the sense that different dimensions are set to zero in different feature vectors, since different features are analyzed as being negatively associated for different nouns. Note by the way that the feature selection we did when defining **F** in section 3.1 can be seen as a form of *global* truncation.

While the total number of non-zero components of **F** is 416237, with an average of 139 per vector, this is reduced to a total of 343776 in **X**, with 115 as the vector average. These truncations might not seem very dramatic, but the effect varies along with vector size. Nouns with initially sparse frequency vectors are usually not affected at all. With vectors of many non-zero elements however, the truncation is quite radical. The projections of the most frequent nouns in the sample, such as *del* (part) and *år* (year) have only 1/5th of the number of non-zero values as their original frequency vectors. In the case of *del* and *år*, the vectors were locally truncated from 776 and 764 non-zero values to 259 and 148 respectively. The truncation ratio is exactly the same whether we perform the association weighting by mutual information, the G^2 test or log odds ratio. Since the exact same pairs of nouns and contexts check out as negatively correlated in all three tests, as described in 3.2.3 above, the same number of non-zero values are set to zero.

Schütze and Silverstein (1997) report good results using such local projections in relation document clustering. In (Schütze and Silverstein, 1997) the term weighting and projection are separate processes. A constant truncation factor is used, which means that all vectors end up having a pre-determined number of non-zero values. Schütze and Silverstein (1997) use truncation of cluster centroids to speed up the proximity calculations while clustering, and the truncation is therefore somewhat more drastic in terms of the number of non-zero values retained. Computing the pairwise proximities between feature vectors is the major bottleneck in clustering, and when using a proximity measure such as the cosine, calculating the proximity of two feature vectors takes time proportional to the number of distinct features in the smaller vector. Schütze and Silverstein (1997) found that, while truncation significantly speeds up the analysis, the quality of the resulting clusters where just as good as those obtained by full-profile clustering. The truncation obtained through the association weighting on our noun-by-context matrix, has the additional effect of reducing noise, since the least reliable features in the co-occurrence patterns are usually the ones that are excised. It might be an interesting idea in later experiments though, to further truncate both the association vectors in **X** and the cluster centroids, by setting the elements with the lowest salience scores to zero.

3.2.5 Ranking by Salience

Given the association matrix **X**, the local contexts can be ranked according to salience scores for the various target nouns. Table 3.7 and 3.8 give examples of such a list for the words *konflikt* (conflict) and *teori* (theory). While the co-occurrence vector of *konflikt* in **F** has 342 non-zero values, the corresponding salience weighted vector in **X** only has 235. The vector representing *teori* (theory) is only reduced from 234 non-zero elements in **F** to 216 in **X**.

		Context Feature		
Rank	Frequency	Feature Type	Feature Word	Association
1	86	`obj_of`	`løse` (solve)	4.64
2	139	`prep_obj_of`	`komme_i` (come_in)	4.08
3	38	`obj_of`	`oppstå` (arise)	3.92
4	48	`subj_of`	`løse` (solve)	3.79
5	46	`subj_of`	`oppstå` (arise)	3.71
6	17	`obj_of`	`unngå` (avoid)	3.20
7	28	`prep_obj_of`	`føre_til` (lead_to)	2.78
8	16	`adj_mod_by`	`indre` (inner)	2.76
9	50	`obj_of`	`skape` (create)	2.54
10	16	`adj_mod_by`	`alvorlig` (serious)	2.53
11	8	`prep_obj_of`	`prege_av` (mark_by)	2.45
12	9	`pp_mod_by`	`hensyn` (condsideration, regard)	2.32
13	19	`adj_mod_by`	`åpen` (open)	2.30
14	6	`subj_of`	`ende` (end)	2.17
15	8	`subj_of`	`handle` (act, buy)	2.06
16	9	`subj_of`	`bryte` (break)	1.97
17	7	`pp_mod_of`	`grad` (extent, degree)	1.96
18	4	`subj_of`	`true` (threaten)	1.91
19	21	`pp_mod_of`	`side` (side, page)	1.91
20	7	`adj_mod_by`	`dyp` (deep)	1.90

Table 3.7: The 20 most salient local contexts of the noun ***konflikt*** (**conflict**).

The display of such context rankings can be useful and interesting in their own right, as it summarizes the most common and distinguishing usage patterns of a word at a quick glance. Tugwell and Kilgarriff (2001) use a related technique in the *word sketch workbench* designed to aid lexicographers compiling dictionaries. After we perform the cluster analysis of the noun vectors in **X** we will also see examples of such context profiles for entire classes of words. Of course, if our main purpose was to construct such "word sketch"-like displays, we would not throw away information about, say, what particular preposition is used within a NP and so on, and we would include a much larger set of contexts.

In relation to the lexicographic task of identifying collocates in a broad sense, Lowe (2001) writes that it "emphasizes the 'second order' nature of semantic space measures of similarity: they reflect regularities across multiple 'first order' association measures, one for each vector element." The "second order" proximity measures are what we turn to in the next section.

		Context Feature		
Rank	Frequency	Feature Type	Feature Word	Association
0	17	`subj_of`	`forklare` (explain, account for)	3.88
1	75	`adj_mod_by`	`økonomisk` (economical)	3.74
2	12	`adj_mod_by`	`vitenskapelig` (scientific)	3.60
3	5	`noun_con`	`erfaring` (experience, practice)	3.30
4	8	`obj_of`	`presentere` (present, introduce)	3.25
5	13	`obj_of`	`utvikle` (develop, evolve, grow)	3.00
6	6	`pp_mod_of`	`utgangspunkt` (point of departure)	2.98
7	5	`pp_mod_of`	`kunnskap` (knowledge)	2.81
8	6	`adj_mod_by`	`administrativ` (administrative)	2.80
9	4	`subj_of`	`stemme` (agree, correspond)	2.71
10	5	`subj_of`	`tilsi` (indicate, justify)	2.71
11	5	`obj_of`	`støtte` (support, back up,)	2.70
12	6	`obj_of`	`styrke` (strenghten)	2.65
13	5	`subj_of`	`beskrive` (describe)	2.51
14	4	`adj_mod_by`	`tradisjonell` (traditional)	2.49
15	3	`subj_of`	`bekrefte` (confirm, acknowledge)	2.44
16	3	`subj_of`	`oppfatte` (understand, interpret, perceive)	2.24
17	2	`pp_mod_of`	`motsetning` (opposition, opposite, contrast)	2.20
18	3	`pp_mod_of`	`forskjell` (difference, distinction)	2.17
19	4	`obj_of`	`nevne` (mention)	2.17

Table 3.8: The 20 most salient local contexts of the noun ***teori*** (**theory**).

3.3 The Proximity Matrix

In relation to numerical pattern recognition, Bezdek (1998) states that there are two fundamental types of data; *object data*, given as feature vectors, and *relational data*, expressing proximities. In our case, the object data are the vectorial representations of the contextual distributions of nouns, as given by $\mathbf{X}$. Our relational data are the similarities of these noun objects as defined by a measure of the proximity measure s. It is this latter data type that is the focus of this section.

3.3.1 Proximity Measures

The notion of proximity can be construed both as a relation of *distance* and as a relation *similarity*, and we will often use the two concepts interchangeably. There is no essential difference between the perspectives of distance or similarity for our purposes, as long as we remember that the inverse relationships holds for the two different conceptualizations. When we want to remain neutral with respect to the particular perspective, we will simply talk of "proximities". In this section we present some standardly used functions for measuring the proximity of vectors in space.

A *distance function* d on a space $\Re^n$ must obey the following conditions for all points $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ in $\mathbf{X}$:

(3.3.1) Minimality: $d(\mathbf{x}, \mathbf{y}) \geq 0$ and $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$

(3.3.2) Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

(3.3.3) Triangle Inequality: $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$

A standard metric for measuring distance is given by the Euclidean norm of the difference of the vectors;

$$d_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{y_i})^2} \tag{3.3.4}$$

The Euclidean distance is actually an instance of the family known as Minkowski metrics, defined as

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|^p} \tag{3.3.5}$$

In the case of Euclidean distance we have $p = 2$, and other common metrics include the Manhattan or City-Block distance for which $p = 1$, and Supremum distance where $p = \infty$.

The vector elements of words with different frequencies will tend to have different magnitudes. This will bias the Euclidean distance so that longer vectors will generally have a better chance to be positioned closer to a given target (see Manning and Schütze, 1999, ch. 15). Long word vectors would be more similar to each other by virtue of length, not semantic similarity. It seems unreasonable that relative frequency should be a factor in determining semantic similarity however. This tendency would actually have a stronger impact if we were to compare words on the basis of their frequency vectors in $\mathbf{F}$ rather than their association vectors in $\mathbf{X}$. The association weighting by A with the log odds ratio as described in section 3.2, means that the we are working with salience scores rather than raw co-occurrence counts. However, words with initially long frequency vectors, will also tend to have longer association vectors, even though the individual frequency counts have been "normalized" for chance co-occurrence, element-wise. The distance would also be dependent on the range of the particular association measure used in A.

A commonly used measure which avoids some of these problems is the *cosine* of the angle between the vectors. The cosine is defined as

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i}{\sqrt{\sum\limits_{i=1}^{n} \mathbf{x}_i^2} \sqrt{\sum\limits_{i=1}^{n} \mathbf{y}_i^2}} \tag{3.3.6}$$

with a value ranging from 0.0 for orthogonal vectors, to 1.0 for vectors that point in the same direction.[5] Because of this constant range the cosine avoids the arbitrary scaling caused by dimensionality and the range of the association measure A (Lowe, 2001).

Under another interpretation the cosine is also known as the *normalized correlation coefficient*. It is then seen as the measure of correlation between the elements of two vectors, scaled by their individual magnitudes (see Manning and Schütze, 1999, ch. 8). At any rate, in terms of the "proximity dichotomy" just mentioned, the cosine is a measure of similarity.

When applied to to *normalized* vectors, the cosine and the Euclidean distance give the same rank order (see Manning and Schütze, 1999). A vector is said to be normalized if it has unit length:

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} \mathbf{x}_i^2} = \sum_{i=1}^{n} \mathbf{x}_i^2 = 1 \tag{3.3.7}$$

The cosine is sometimes called the normalized inner product, since its denominator involves the lengths of the vectors. In the case of normalized vectors then, the cosine measure can be reduced to the inner product alone:

$$\cos(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i \tag{3.3.8}$$

Since normalizing the vectors means they all have the same length, this is often done in order to avoid the problems related to comparing vectors of frequent and rare words, as mentioned above. By using the simple inner product or dot product, we can also implement the distance measure more efficiently.

As said, when the vector arguments are normalized, the Euclidean distance and the dot-product give the same answer with respect to which vectors are closest together and farthest apart in the space. These are also the proximity measures that we will use in the cluster analysis and when defining the semantic space. We normalize each association vector in $\mathbf{X}$ according to (3.3.7) by dividing each of its components by the vector's length normalized. The similarity function s in $\langle \mathbf{F}, A, s \rangle$ is then specified to be the cosine, as computed by the dot-product.

[5] All the word vectors are positioned in what corresponds to the first quadrant of the plane, in the sense that they only have positive coordinates.

The Similarity Matrix A much-used construct when dealing with vector space models, is that of a *proximity matrix.* The total set of pairwise proximities according to s on the set of association vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, defines a $k \times k$ *proximity matrix* $\mathbf{S}$. More specifically, since s is a measure of similarity, $\mathbf{S}$ is a *similarity matrix.* A component $\mathbf{S}_{ij}$ represents the similarity of the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

Clustering algorithms often differ with respect to taking a proximity matrix (e.g $\mathbf{S}$) or a data matrix (e.g $\mathbf{X}$ or $\mathbf{F}$) as input arguments, and we will see examples of both approaches in the next chapter. The latter approach also requires that a proximity function such as s is specified.

3.3.2 Nearest Neighbors

There are many ways in which the similarity matrix can be put to immediate use to get semantic information about words. In many approaches to modeling word similarity, the proximity matrix itself is in fact the end product. Given $\mathbf{S}$, we can quickly retrieve the k nearest neighbors (kNNs) of any given target. The concept of kNN simply refers to the k points closest in the space to a given target point. In our setting, a list of kNNs for a given target noun consists of the k nouns that are closest to it in the word-context space, ranked according to similarity. Hindle (1990) derives such lists of nouns on the basis of verb object and subject relationships, which clearly displayed relations of semantic similarity. We will see examples of kNN word lists derived from our noun–context space in a moment.

RNNs On the basis of kNNs, one can define the more strict relationship of *RNNs* (Reciprocal/Respective Nearest Neighbors). Two points are RNNs if each is the nearest neighbor of the other. Hindle (1990) and Lin (1998) use the notion of RNNs to identify words that are substitutable or near-synonyms. On the basis of the similarity matrix $\mathbf{S}$ computed for our context space $\langle \mathbf{F}, A, s \rangle$, we are able to find a total of 421 pairs of such RNNs. Table 3.9 displays a randomly picked subset of the pairs (as in (Lin, 1998), every 10th pair of the total set of RNNs is selected.). Most of the retrieved noun pairs in the list are quite similar in meaning. The list also illustrates the commonly observed phenomenon that antonymic or complementary words are rendered as similar under a distributional characterization, such as the RNNs; *økning* (increase) / *reduksjon* (decrease); *slutt* (end) / *begynnelse* (beginning); and *liv* (life) / *død* (death) (not shown in table 3.9, similarity score 0.478). However, this might be more of an issue when analyzing verbs and adjectives than when dealing with nouns.

A special group of cases in the set of RNNs has been sifted out and are shown separately at the bottom of the table. These are word pairs whose similarity is due to the way words are lemmatized in the corpora that we use. Nouns such as *rose* (rose) and *ros* (praise, commendation) are homographs in some of their common inflections. Consider the plural form *roser* in the sentences below.

(3.3.9)	*Livet*	*er*	*ingen*	*dans*	*på*	*roser.*
	Life-the	is	no	dance	on	roses.

‘Life is not a bed of roses.’

(3.3.10) *Livet*	*er*	*ingen*	*dans*	*på*	*roser.*
Life-the	is	no	dance	on	commendations.

'Life is no dance on commendations.'

In the Norwegian sentence in (3.3.9)–(3.3.10), both lemmas *rose* and *ros* are extracted for the word form *roser*. Although no human language user would think of the praise or commendation sense of *roser* in this context, the correct meaning and lemma is ambiguous to the tagger. Consequently, such base forms will share a high number of features. The process of extracting words is described in more detail in section 2.2. Many of the retrieved RNNs consist of such base forms that result from homographic word forms with multiple lemma analysis per single token in the tagged corpora. In many cases, the second unit of the pair ('Word 2' in table 3.9) is very marginal in normal language use.

kNN-Lists As said, one of the immediate uses of the similarity matrix is to define sets of similar words in the form of kNN lists. The tables 3.10 − 3.18 below show the kNNs for a sample of target nouns in $\langle \mathbf{F}, A, s \rangle$, with $k = 10$. We see that some of the same problems hold for these lists as for the RNNs discussed above. As an example, the fact that the word *bøk* (beech) is found to be similar to the target *fortelling* (story) in table 3.12, is due to the ambiguous lemmatization of plural forms of *bok* (book) in the tagged corpora. In such cases we mark the "odd" entry as, in this particular eaxmple,"*bøk* (beech Pl = *bok*)", indicating that the plural forms of *bøk* can also correspond to the base form *bok* (book). Certain other entries are marked with "?" indicating "non-words" resulting from spelling mistakes or erroneous lemmatization. (Refer to section 2.2 for details of how words are identified in the tagged corpora.)

Dagan et al. (1999) use such kNN sets of similar words for *nearest neighbors averaging*, a method for estimating the probabilities of rare joint events. Co-occurrence probabilities for a target and a second variable are estimated by averaging the corresponding probabilities for a set of similar instances. To give an example, let us say we want to estimate the probability of seeing the noun *Norway* in a prepositional phrase '*government of* __'. In a *class-based* approach, the estimates can be based on the class that the word belongs to, say COUNTRY[6]. This is similar to the approach taken by Pereira et al. (1993). In a *similarity-based* approach as that of Dagan et al. (1999) on the other hand, each word can be seen to define their own class. The class simply consists of the words that are most similar to a given target (Lee and Pereira, 1999), such as the set shown in table 3.11 for *norge* (Norway). The word sets must then be delineated either by specifying a similarity threshold or the number of members k.

Grefenstette (1992) too extracts such similarity relations with the SEXTANT system on the basis of overlapping syntactic context. He argues that such ranked word similarities can be used for improving precision and recall in information retrieval by expanding query terms with other close words, and as a means for constructing domain-specific dictionaries or corpus-defined thesauri.

[6]The classes obtained through a clustering analysis will not actually be symbolically labeled, but we will sometimes use such symbolic labels for the sake of convenience when discussing them. When referring to labeled classes we will adopt the convention of using a SMALL CAPS FONT.

Word 1	Word 2	Similarity
år (year)	*måned* (month)	0.679
folk (people)	*nordmann* (Norwegian)	0.610
spørsmål (question)	*problemstilling* (problem)	0.587
hånd (hand)	*hand* (hand)	0.871
kamp (fight, game)	*turnering* (tournament)	0.498
medlem (member)	*deltaker* (participant)	0.477
pasient (patient)	*klient* (client)	0.579
kirke (church)	*menighet* (church community, congregation)	0.460
slutt (end)	*begynnelse* (beginning)	0.737
prosjekt (project)	*program* (program)	0.502
ressurs (resource)	*kapasitet* (capacity)	0.511
sønn (son)	*datter* (daughter)	0.658
økning (increase)	*reduksjon* (decrease)	0.712
kunst (art)	*litteratur* (literature)	0.416
beløp (amount, sum)	*sum* (sum)	0.550
frihet (freedom)	*handlefrihet* (freedom of action, latitude)	0.394
besøk (visit)	*opphold* (stay)	0.438
utforming (design, arrangement)	*gjennomføring* (implementation, carrying out)	0.527
elv (river)	*bekk* (river, stream)	0.417
tilgang (supply, access)	*tilgjengelighet* (accessibility, availability)	0.476
olje (oil)	*gass* (gass)	0.409
ende (end, bottom)	*hjørne* (corner)	0.435
arbeidsmarked (labor marked)	*arbeidsliv* (employment sector)	0.466
pensjonsalder (retirement age)	*aldersgrense* (age limit)	0.489
jørn (Jørn)	*ingrid* (Ingrid)	0.706
meri (?)	*oter* (? otter)	0.718
koffert (suitcase)	*veske* (bag, purse)	0.536
per (Per)	*jon* (Jon)	0.423
skepsis (skepticism, disbelief)	*misnøye* (discontent, dissatisfaction)	0.365
selvstendighet (independence)	*uavhengighet* (independence)	0.425
tall (number)	*talle* (manure)	0.753
kurs (course)	*kur* (cure)	0.579
kone (wife)	*kon* (cone)	0.692
fordel (advantage)	*fordeler* (electrical distributor)	0.569
vind (wind)	*vinde* (winch, reel)	0.848
narkotika (narcotic)	*narkotikum* (narcotic)	0.933
motstand (resistance, opposition)	*motstander* (opponent)	0.347
skrift (writing)	*skrifte* (testimony, reprimand)	0.623
rute (square, window)	*ruter* (diamond playing card)	0.671
kinn (chin)	*kinne* (butter churn)	0.673
rose (rose)	*ros* (praise, commendation)	0.633
tomt (property, real estate)	*tomte* (leprechaun, gnome)	0.804

Table 3.9: Random examples of Reciprocal Nearest Neighbors in the Semantic Space.

Rank	Neighbor	Similarity
1	*hode* (head)	0.432
2	*hår* (hair)	0.389
3	*rygg* (back)	0.387
4	*munn* (mouth)	0.385
5	*hake* (chin)	0.375
6	*skjegg* (beard)	0.370
7	*ansikt* (face)	0.369
8	*nakke* (neck)	0.360
9	*arm* (arm)	0.358

Table 3.10: The 10 nearest neighbors of the target noun ***nese*** (**nose**).

Rank	Neighbor	Similarity
1	*danmark* (Denmark)	0.579
2	*sverige* (Sweden)	0.567
3	*tyskland* (Germany)	0.562
4	*russland* (Russia)	0.550
5	*kina* (China)	0.533
6	*bergen* (Bergen)	0.512
7	*frankrike* (France)	0.511
8	*land* (land, country)	0.508
9	*england* (England)	0.499
10	*finland* (Finland)	0.498

Table 3.11: The 10 nearest neighbors of the target noun ***norge*** (**Norway**).

When studying the words in tables 3.10 – 3.18 we see that many of the retrieved sets of nearest neighbors are encouragingly coherent. The kNN lists clearly demonstrate that quite precise sense distinctions can be discerned in the context-space. As noted by Pantel and Lin (2002) however, a serious drawback related to such word lists and similarity matrices, is that the listed words may be similar to different senses of the target due to polysemy. Such nearest-neighbors induced word sets might be fine if the goal is restricted to capture distributional similarity as such. However, we soon run into problems once we add the expectation that semantic similarity should follow. In section 2.2.1 we noted that different senses of a word are conflated in a single representation when recording the co-occurrence information from corpora. In relation to the vector space representation of such distributional patterns, Resnik (1993) remarks;

> If each token is associated with a single point in semantic space, then words having multiple senses will occupy a point determined by the relative frequencies of the individual senses. Although in many cases multiple word senses share relevant properties – for example the *newspaper* and *term paper* senses of *paper* – in other instances the single point in semantic space represents an amalgam of properties that may not preserve the relationships associated with component word senses.

Rank	Neighbor	Similarity
1	*bok* (book)	0.419
2	*setning* (sentence)	0.409
3	*roman* (novel)	0.406
4	*dikt* (poem)	0.401
5	*tekst* (text)	0.398
6	*bøk* (beech, Pl = *bok*)	0.370
7	*novelle* (short story)	0.369
8	*historie* (story, history)	0.364
9	*verk* (piece)	0.360
10	*tale* (speech)	0.358

Table 3.12: The 10 nearest neighbors of the target noun ***fortelling*** (**story**).

Rank	Neighbor	Similarity
1	*norm* (norm)	0.350
2	*tradisjon* (tradition)	0.321
3	*variant* (variety)	0.320
4	*form* (form)	0.319
5	*stil* (style)	0.317
6	*struktur* (structure)	0.317
7	*ramme* (frame)	0.316
8	*trekk* (feature, property)	0.314
9	*fellesskap* (community)	0.311
10	*ram* (ram ?)	0.311

Table 3.13: The 10 nearest neighbors of the target noun ***mønster*** (**pattern**).

This means that a simple list of similar words may reflect multiple meanings of a polysemous target. Some of the word sets shown below seem quite semantically coherent, such as those retrieved for the nouns *nese* (nose), *norge* (Norway), and *konflikt* (conflict). Yet others again seem to be more diluted. Consider the neighbors retrieved for the word *vann* (water) displayed in table 3.16. Among the possible interpretations and similarity relations, we can gleam the senses of water as an "elementary substance" or one of the four elements (earth, air), as a beverage (milk, beer, coffee) or a nutrient (food), a liquid substance (blood), or a body of water (ocean). Among the nouns listed as similar to the *middag* (dinner), we might see traces of its sense as food (sausage, sauce, fruit, cake), a type of meal (breakfast, meal), or a social event (party). As an example of a rather vague word with many metaphorical uses, consider the target *mønster* (pattern). This vagueness is clearly reflected in its neighboring words, as shown in table 3.13. Although the neighbors such as *form* (form), *tradisjon* (tradition), *stil* (style) and *struktur* (structure) can all easily be seen to relate to *mønster* (pattern), they do so in rather different ways.

We see that vague or polysemous targets cause problems for the semantic coherency of these kNN sets. The same problems arise, of course, in relation to

Rank	Neighbor	Similarity
1	*frokost* (breakfast)	0.558
2	*måltid* (meal)	0.416
3	*mat* (food)	0.398
4	*pølse* (sausage)	0.321
5	*sjokolade* (chocolate)	0.301
6	*fest* (party)	0.292
7	*kake* (cake)	0.286
8	*frukt* (fruit)	0.285
9	*saus* (sauce)	0.282
10	*kak* (cak ?)	0.278

Table 3.14: The 10 nearest neighbors of the target noun ***middag*** (**dinner**).

Rank	Neighbor	Similarity
1	*sang* (song)	0.469
2	*melodi* (melody)	0.439
3	*tone* (tone)	0.439
4	*lyd* (sound)	0.420
5	*tekst* (text)	0.410
6	*vers* (verse)	0.406
7	*låt* (tune)	0.395
8	*kunst* (art)	0.389
9	*bilde* (picture)	0.386
10	*plate* (record, disc)	0.383

Table 3.15: The 10 nearest neighbors of the target noun ***musikk*** (**music**).

Rank	Neighbor	Similarity
1	*luft* (air)	0.597
2	*kaffe* (coffee)	0.495
3	*sjø* (sea, ocean)	0.434
4	*øl* (beer)	0.423
5	*mat* (food)	0.420
6	*melk* (milk)	0.401
7	*glass* (glass)	0.400
8	*jord* (earth, soil)	0.394
9	*blod* (blood)	0.391
10	*vatn* (water)	0.377

Table 3.16: The 10 nearest neighbors of the target noun ***vann*** (**water**).

polysemous neighbors. Because we tend to automatically select the appropriate sense of the neighboring words, such lists of kNNs often seem more consistent than they really are. This becomes especially clear when translating the entries from one language to another. The Norwegian noun *hake* can be used to mean a variety of things; a chin, a hook, a check mark, or an inconvenience. All these

Rank	Neighbor	Similarity
1	*problem* (problem)	0.537
2	*krise* (crisis)	0.481
3	*strid* (fight, discord, controversy)	0.470
4	*uenighet* (disagreement)	0.453
5	*motsetning* (contrast, difference, opposition)	0.438
6	*vanskelighet* (difficulty, trouble)	0.432
7	*kris* (?)	0.426
8	*misforståelse* (misunderstanding)	0.425
9	*brudd* (break, rupture)	0.420
10	*krangel* (quarrel)	0.412

Table 3.17: The 10 nearest neighbors of the target noun ***konflikt*** (**conflict**).

Rank	Neighbor	Similarity
1	*innsikt* (insight)	0.564
2	*respekt* (respect)	0.553
3	*kunnskap* (knowledge, information)	0.527
4	*holdning* (attitude, stance)	0.490
5	*bevissthet* (awareness, consciousness)	0.479
6	*tillit* (trust, confidence)	0.455
7	*kjennskap* (knowledge)	0.437
8	*erkjennelse* (acknowledgement, (re)cognition)	0.413
9	*kommunikasjon* (communication)	0.410
10	*trygghet* (security, safety, confidence)	0.409

Table 3.18: The 10 nearest neighbors of the noun ***forståelse*** (**understanding**).

different senses of *hake* are relatively common. However, when translating it in the context of other words listed as similar to *nese* (nose) in table 3.10, it can in fact be very hard to recognize any other meaning than that of the lower part of the face and jaw. Analogous examples can be found in most of the sets of similar words displayed here. For this reason, the English translations that are offered for various word lists and clusters throughout this paper, should only be regarded a rough guide, and not, of course, as exhaustive and accurate equivalents of the Norwegian source.

Moreover, this phenomenon of "selective reading" is, naturally, also apparent among the English translations, and often in a way which parallels the ambiguity of the Norwegian original. Consider the list of words that are judged similar to *forståelse* (understanding) in table 3.18. When presented with the word *holdning* (stance) in this context, we recognize its meaning as an intellectual or emotional attitude more readily than its meaning as physical posture. When *plate* (disc) is suggested as similar to *musikk* (music) in table 3.15, we probably find the meaning as a phonograph recording much more salient than any other given kind of flat body.

We have here discussed lists of kNNs (with a fairly high value for k) with the purpose of shedding light on some problematic issues related to the representation of semantic classes. Many of these issues also applies, which we will see, to *hard clustering models* as described in the sections to follow. The main emphasis in most work on sematic space models is typically the similarity relations that can be found to hold between individual words. For the purposes of this thesis, the semantic space model rather provides a framework for formulating the categorization task.

Although the kNN delimited word sets might be useful for many purposes, such as nearest neighbor averaging, the examples given in the last section clearly show that the similarity relations expressed within such sets are far too heterogeneous to make them suited for representing semantic categories. In the next section we further explore alternative ways to derive and represent the notion of a semantic class and more properly characterize the meaning of a word. Nonetheless, taken together with the set of RNNs displayed in table 3.9, the examples of kNNs drawn from the proximity matrix $\mathbf{S}$ for $\langle \mathbf{F}, A, s \rangle$ go a long way towards demonstrating the potential of semantic knowledge that is inherent in the noun–context space.

3.4 Meaningful Classes

We have established that the lexical distribution of a noun is represented as a vector in a multidimensional space given by the triplet $\langle \mathbf{F}, A, s \rangle$, corresponding to the co-occurrence matrix, the association measure and the similarity function. When measuring the distributional correspondence of words, we measure the proximity of their association vectors in $\mathbf{X} = A(\mathbf{F})$.

Given that words are thought of as such vectors or points positioned in the context space, the previous section showed how sets of semantically similar words can be identified as points in the vicinity of a given target. By the same token, a semantic category or concept can in this model be taken to correspond to a more densely populated region or a cluster of points in the space. In this section we try to further approach the notion of meaning and conceptual classes. We will try to clarify some of the properties that we want to hold for the purportedly meaningful classes that we seek to discover or construct through the cluster analysis. Intuitively, if we want our categories to represent meanings, we would want our classes to display the same properties that we think hold for meanings. In the following sections we briefly review some of the previous work done in relation to cluster based modeling of word similarity. Simultaneously, we discuss some general properties of "concepts" or semantic classes, and thereby try to arrive at some general properties that would be desirable also in a model of such classes. Along the way, we look at the specifics of how classes and class memberships are represented in our semantic space model.

3.4.1 Class-Based Similarity

Hard Cluster Models In section 3.3.2 we saw how, due to the possible multiple meanings of words, a diverse range of relations and senses will often be expressed in the set of kNNs for a given target. All in all, such simple word lists do not, of course, make for a well-founded model of semantic classes or concepts.

With the objective of modeling senses, much of the same criticism also applies to approaches based on *hard* or *crisp* clustering models. A class is then given simply as a set of words, and each word is assigned membership in one class only. Note that, although we briefly review some examples of word clustering in this section, a more thorough and technical description of clustering methods is given in chapter 4.

Brown et al. (1992) derives sets of hard word clusters through a bottom-up hierarchical method. The main aim of Brown et al. (1992) is at constructing a *class-based n-gram model*. This means that the sets of conditional probabilities $P(w_n|w_1,\ldots,w_{n-1})$ for a given string history in the language model is computed as $P(w_n|c_n)P(c_n|c_1,\ldots,c_{n-1})$, for words w and classes c (Brown et al., 1992).

In a similar spirit, Li and Abe (1998) aim at syntactic disambiguation through *class-based estimation of joint probabilities*. Words are clustered into "discrete" groups on the basis of co-occurrence information about noun and verb pairs. Li and Abe (1998) use a bottom-up algorithm based on the minimum description length principle (MDL), that clusters both nouns and verbs into classes simultaneously. On the basis of the resulting clusters, the joint probability of a noun n and a verb v is computed as $P(n,v) = p(C_n, C_v)P(n|C_n)P(v|C_v)$, where C_n and C_v are the classes to which n and v uniquely belong (Li and Abe, 1998).

As said, within a hard clustering model, such as that of (Brown et al., 1992) and (Li and Abe, 1998), each word only belongs to a single class. Obviously, such a categorization scheme is not able to accommodate the multiple meanings that a word may hold. Either the classes must contain a blend of various senses of different words, or a lot of possible meanings must be ignored and suppressed in their interpretation.

Types of Similarity Resnik (1993) points out a pervasive problem that often attaches to distributional methods for discovering word classes; it can be difficult to come up with a common description of the type of information that the classes convey. In many cases, the only trait that is common to all classes is that of distributional similarity itself, and the classes often encode syntactic information in addition to semantic aspects (Resnik, 1993). The clusters found by Brown et al. (1992) for example, include words that are grouped together on the basis of common stems, number, tense and inflections, in addition to showing relations of a "*semantic flavor*" (Brown et al., 1992). Pantel and Lin (2002) report some "part-of-speech confusion" in the clusters obtained with their CBC algorithm (described below). This is because their feature vectors for words conflate all tokens of a base form regardless of syntactic category (Pantel and Lin, 2002). In our case, however, as noted in section 2.1.2, the feature vectors of the context space can be seen as inherently typed for POS. Since they only encode information about nouns, in addition to only relying on the lemmatized forms, we are not susceptible to find clusters that encode such syntactic properties. Furthermore, as discussed in section 2.1, depending on what contextual properties one chooses to focus on, semantic relationships of somewhat different kinds seem to be revealed. In broad terms, the sort of meaning distinctions that we seek to capture in our noun–context space, are relations of semantic *similarity* as opposed to semantic *relatedness*, – a distinction we noted in section 2.1.1.

Although there are undoubtedly many difficulties related to pinning down which types of similarities are actually expressed within distributionally derived word classes, a part of the problem is perhaps more far-reaching, and can be seen to apply to conceptual categories in general. One of the difficulties here concerns the "similarity of the similarities". It is often hard to find a "common denominator" for the type of the similarity relations that hold between members of a class, – not only across different categories but even within one and the same class. Are the similarities that hold between the members of the concept COLOR, the same as those within the concept VEHICLE? Is *red* similar to *purple* in the same way that *car* is similar to *boat*? Is *car* similar to *boat* in the same way that *motor-bike* is to *train* ? One of the few general and common descriptions we can offer for the relations that hold within such conceptual classes, is perhaps the notion of *family resemblances* as introduced by Wittgenstein (1953) in his critique of classical categories. In his renowned comparison of various examples of games, Wittgenstein found that "*we see a complicated network of similarities overlapping and crisscrossing: sometimes overall similarities, sometimes similarities of detail*" (Wittgenstein, 1953). The members of a conceptual category often resemble each other in the same way that the members within a family do, where no common feature is necessarily shared by all.

Multiple Memberships It seems clear that any serious attempt at categorizing words semantically must allow for *multiple class memberships*. The semantics of a word might then more properly be characterized by, not a set of words, but a set of classes. From a class perspective, this means that the groups are not necessarily *disjoint* or *mutually exclusive*.

As we shall see, the property of multiple memberships and non-exclusive classes, can be thought implemented in at least two ways. One way is to think of the classes as *overlapping*, with the possibility of words having *disjunctive memberships* in multiple classes. Under this approach, a word is a full-fledged member of any number of crisp classes. Alternatively, words can belong to multiple classes with *varying degrees of memberships*. In the terminology of cluster analysis, this corresponds to the distinction between *disjunctive* and *soft* clustering models. We will briefly describe the works of Pereira et al. (1993) and Pantel and Lin (2002), as examples of the soft and disjunctive approaches respectively.

Pereira et al. (1993) have previously applied a soft clustering technique to a set of nouns on the basis of verb-object co-occurrence data. Their soft word clustering is based on a *deterministic annealing* procedure adopted from statistical mechanics. Instead of representing the co-occurrence patterns of words as vectors in a space, Pereira et al. (1993) construe the contextual profiles as probability distributions over a set of events. 1000 nouns are hierarchically clustered and assigned probabilistic memberships in the resulting groups. The membership probabilities are determined on the basis of the relative entropy between a word distribution and a cluster centroid distribution (Pereira et al., 1993). Pereira et al. (1993) then use the clusters for a class-based model of word co-occurrence and estimating joint probabilities.

Although there have been much work in relation to automatic and unsupervised categorization of similar words, such as (Brown et al., 1992; Pereira et al., 1993; Li and Abe, 1998), the intention is usually to uncover distributional similarity as such, rather than to identify word senses. The goal is typically to smooth probability distributions through class-based averaging (see Lee and Pereira, 1999). Approaches that instead focus primarily on semantic aspects, are, by contrast, typically cast within a *supervised* or *semi-automatic* framework with the goal of word sense disambiguation. Instead of attempting to induce the classes automatically as in clustering, the methods are often based on manually defined sets of semantic categories, e.g. Roget's Thesaurus in (Yarowsky, 1992) and WordNet in (Resnik, 1997). In fact, Pantel and Lin (2002) write that "*to the best of our knowledge, there has been no previous work in automatic word sense discovery from text*". Pereira et al. (1993) do, however, say that they aim at deriving "*hidden sense classes*", and to "*model senses as probabilistic concepts or clusters c with corresponding cluster membership probabilities* $p(c|w)$ *for each word* w" (Pereira et al., 1993).

Pantel and Lin (2002) try to detect senses of a word, by a method they call *clustering by committee* (CBC). Words are characterized by feature vectors encoding dependency relations extracted from a corpus, such as adjectival modifications, verbal objects, etc. Similarly to the association weighting we described in section 3.2, the feature values are given by the mutual information score for the feature and the word, weighted by the discounting factor given in equation (3.2.5). The similarity between feature vectors is then computed by the cosine coefficient, as given in (3.3.6).

The *committees* in CBC are tight and small clusters that are initially constructed for the purpose of defining representative cluster feature vectors (Pantel and Lin, 2002). The committees are formed through bottom-up clustering of the words in the kNNs of the words in the data set, with $k = 10$. The finally chosen committee clusters are sought to be well scattered in the feature space, and consist of small and tight groups of words. A committee centroid is given by computing the "*mutual information vector*" for the sum of the committee members, analogously to how weighted feature vectors are computed for individual words (Pantel and Lin, 2002).

CBC then assigns each word to the clusters that have a committee centroid closer than some given threshold. The overlapping features of a word and a centroid are removed from the word vector upon assignment. By removing the intersecting features of a word and a class before possibly assigning the word to another cluster, CBC tries to discover the less frequent senses of a word as well as avoid finding duplicate senses (Pantel and Lin, 2002). The feature vectors of the clusters on the other hand, remain constant and are not updated. The output of the CBC procedure is a list of clusters for each word. Each cluster that a word belongs to is taken to represent a sense of the word. Although Pantel and Lin (2002) include the similarity score of a (possibly revised) feature vector and a centroid vector when displaying the cluster memberships of words, they do not employ a notion of graded belonging and seem instead to subscribe to the idea of disjunctive memberships.

Typicality In the models produced by hard and disjunctive clustering, all members of a class have *equal status*. In reality, however, we often judge certain members of a conceptual category as more central and representative, while others are perceived as peripheral. As always, some are simply more equal than others. A much studied characteristic of conceptual categories displaying family resemblances, is that the members show varying degrees of typicality (e.g. Rosch and Mervis, 1975). Objects with greater family resemblance to a category, are judged to be more typical (e.g. Rosch and Mervis, 1975). Few would be reluctant to agree that *car* belongs in the category VEHICLE? But what about *surfboard*? Or *horse*, or *satellite*?

While some words are clear-cut instances of a given concept, others are borderline cases. This is an important factor that both hard and disjunctive clustering schemes fail to account for. In order to reflect this in a model of semantic classses, the memberships of words can be graded with respect to typicality. In this way, both *horse* and *car* can be seen to belong to a class VEHICLE, but the latter with a greater strength of membership than the former.

Prototypes Within prototype theory, the typicality of a given exemplar is usually thought of as given by its similarity towards a class *prototype*. The information contained in a given prototype represents an abstraction across the specific instances of the category. The prototype can be thought of as a central, but constructed, representative of the group, expressing the most salient overall features of the members. The typicality of members within a class can then be defined and graded on the basis of their resemblance to this group prototype.

When a word class is construed as a cluster of points in the semantic space, one way to represent it is in the same way that we represent the individual words, – by a feature vector. By way of some average or summary of the members, we can define the center of the region. This *centroid* can then serve as a class abstraction, representing the prototypical contextual profile of the cluster. Moreover, when a class is given such a vectorial representation, we can measure the distance between a word and a class in the same way that we do for two individual words. The most straight-forward way to define the centroid $\mathbf{v}_i$ for a given class b_i, is by

$$\bar{\mathbf{v}}_i = \frac{\sum_{\mathbf{x}_j \in b_i} \mathbf{x}_j}{|b_i|} \tag{3.4.1}$$

where $|b_i|$ is the size of the cluster b_i[7]. The centroid is here simply the mean of the points in the group, – sometimes referred to as the *center of mass* or *center of gravity* in the cluster.

But this is far from the only way to define the notion of prototypes. Another alternative is to use *medoids*, – members that are considered central or particularly representative of their class (see Manning and Schütze, 1999, ch. 14). As opposed to centroids, medoids are actual exemplars of the group, – not some averaged abstraction. There are problems related to both these approaches when the goal is to define prototypes (see Pantel and Lin, 2002). When defined as by equation (3.4.1), peripheral members may get too much influence on the centroid. However, since individual words will have their own idiosyncrasies, medoids may not serve as good prototypes either. A sort of compromise might be to define the centroids on the basis of medoids. Pantel and Lin (2002) do

[7]Throughout this paper, we use b and B to denote, respectively, a hard or crisp cluster and a set of such clusters. Think of 'b' as a mnemonic for boolean classes or bins.

something similar by defining cluster feature vectors on the basis of the so called committees, – the small and tight clusters constructed in the initial step of CBC.

One should keep in mind, however, that entirely different approaches to representing classes are in use as well. Within *exemplar based* methods for example, one does not rely on any form of abstraction at all, and every stored instance of a group can potentially represent the class. Within the field of machine learning, this distinction in class representation is often known as one between *centroid based learning* and *instance based* or *memory based learning* (MBL) (see Mitchell, 1997, ch. 8). To measure the similarity between a given target and a class in an *instance based* approach, one might need to compare it to every single exemplar of the group. When determining class memberships, one can also use some sort of majority voting, as in the k-nearest neighbor method. A point is then assigned membership in the class to which most of its k nearest neighbors belong. Every approach has its "pros and cons", and each might be suitable for different tasks. We will not go into any such details here, but a great advantage of the centroid based approach is that of computational simplicity; when determining the class membership of a given target, we just compute the pairwise similarities towards the class abstractions, rather than towards every other individual word.

Given a definition of class centers as in equation (3.4.1), vector space models also offer an intuitive way to formalize the notion of typicality or graded membership in multiple classes. The "strength of belonging" can be modeled by the distance between a data point and the centroids (see e.g. Manning and Schütze, 1999, p. 499). Note however, that the centroid definition in (3.4.1) presupposes that the class b_i is defined by "boolean" or crisp memberships, so that its members can in fact be counted. We will later see how this definition can be "softened" and generalized to cover classes with non-discrete memberships.

Fuzzy Borders Let us again switch to the perspective of classes rather than words for a moment. In order to accommodate the typicality effects described above, we must acknowledge that the boundaries of the semantic classes are not always crisp and precise. If the notion of category membership or belonging is construed as a *graded* relation, it follows that the class borders should be thought of as *vague* and *fuzzy*. As already pointed out, within hard and disjunctive clustering schemes, membership is a boolean "all-or-none" relation, and class borders are "absolute". Again we can conclude that, if we want to represent conceptual categories, we should look towards soft clustering models, where the class boundaries need not be precisely determined.

Let us sum up the discussion in this section so far. For the purpose of modeling semantic categories, in a way that can account for the possible multiple meanings of words, we must allow words to have multiple memberships in several classes. Moreover, we want the memberships to be graded with respect to typicality. This typicality can be based on the resemblance of a word towards a class prototype. The next section suggests a framework that can be adopted for expressing the sort of graded memberships and typicality relations that we are after, based on the concept of *fuzzy sets*. We then move on to outline the type of clustering algorithms that we will apply to the noun data, which we describe in the next chapter.

3.4.2 Fuzzy Sets

Fuzzy sets were introduced by Zadeh (1965) for the purpose of describing classes that do not have precisely defined criteria for membership. The notion of a *fuzzy set* seems to lend itself nicely to formalize the sort of prototype resemblances that we discussed in the previous section. A fuzzy set is a class with a continuum of grades of memberships; A fuzzy set ζ on $\mathbf{X}$ is characterized by a membership function u_ζ, which associates with each $\mathbf{x}_j \in \mathbf{X}$ a real number in the unit interval $[0, 1]$ Zadeh (1965). The value of $u_\zeta(\mathbf{x}_j) = u_{\zeta j}$ represents the "*grade of membership*" that $\mathbf{x}_j$ holds in ζ, where unity corresponds to the highest degree of membership Zadeh (1965). In the case of an ordinary (crisp) set on the other hand, the two-valued characteristic function is restricted to either 1 or 0, corresponding to whether the object belongs to the set or not.

Applied to our setting, we can think of ζ as representing a "fuzzy concept". The strength of membership u_{ij} expresses the degree of resemblance that a word $\mathbf{x}_j \in \mathbf{X}$ holds toward some ideal representation of ζ.

To make things less cumbersome, we will use the notation u_ζ to denote both the characteristic function and the set itself. This means that, for a set of c classes, we write u_i to denote the ith class. Although we use these meanings interchangeably, the correct sense should always be clear from the context (at least in cases where confusion would be harmful).

Fuzz vs. Probabilities Although the grades of memberships associated with a fuzzy set are often interpreted as probabilities, this was explicitly not the intention in Zadeh (1965). Although both probabilities and fuzziness concern a notion of *imprecision*, fuzziness is meant to deal with imprecision in the absence of sharp boundaries, rather than epistemic uncertainty as a result of incomplete information. The graded memberships express the strength with which something belongs to an ill-defined set. We can think of the difference as the *certainty* by which some *binary* property is known to hold, and the *degree* to which some *graded* property is known to hold (Ruspini and Francesc, 1998). These are both expressions of lack of certainty, but the philosophical source of the uncertainty might differ (Bezdek, 1981). Bezdek and Sankar (1992b) illustrate the difference with the following (paraphrased) example: Let L be the set of all liquids, and let the fuzzy subset $\mathcal{L}$ be the set of all *potable* liquids. Suppose a wanderer has been in the desert for a week without drink, and comes upon two bottles marked A and B. The weary traveler is informed that $Pr(A \in \mathcal{L}) = 0.91$ while $u_\mathcal{L}(B) = 0.91$. This means that B is considered to be potable to a degree of 0.91. In other words, B is regarded to be fairly similar to for instance pure water, which is considered a perfectly potable liquid (i.e $u_\mathcal{L}(\text{Pure Water}) = 1.0$). While B might contain swamp water it would not contain liquids such as hydrochloric acid. The value of $Pr(A \in \mathcal{L})$ on the other hand, means that there is about 1 in 10 chance of the contents of A being lethal. When asked to choose a bottle, most subjects would opt for B.

Another difference is manifested upon *observation*. Suppose we discover that A contains, say, gasoline and B contains coffee brewed 6 hours ago. The posterior $Pr(A \in \mathcal{L})$ drops to 0 while $u_\mathcal{L}(C)$ remains 0.91. While probability concerns likelihood, fuzzy memberships represent similarities of objects in respect to imprecisely defined properties (see Bezdek and Sankar, 1992b).

Fuzz vs. Possibilities A concept that is closely related to the notion of fuzzy membership, is that of *possibility.* The theory of possibility is a non-classical theory of uncertainty, distinct from probability theory, and was introduced by Zadeh (1978a) for modeling "*flexible restrictions*" based on vague information, described by way of fuzzy sets. Dubois and Prade (1998) writes that, "*fuzzy sets, viewed as possibility distributions, act as flexible constraints on the values of variables referred to in natural language sentences*". In fact, Zadeh (1978b) proposed a meaning representation language, PRUF (Possibilistic, Relational, Universal, Fuzzy), for natural languages based on a possibilistic framework. The basic idea of PRUF is that referential meaning can be given as a fuzzy correspondence between the terms of a vocabulary and a universe of discourse.

Although the notion of possibility and fuzziness are closely related, they concern rather different perspectives regarding uncertainty and imprecision. We will not go into much detail of the notion of possibility, but just give a brief example to illustrate its difference from the notion of fuzziness. Zadeh (1978a) views a *possibility distribution* π_i to be determined by the fuzzy set u_i. Given a variable y defined on $\mathbf{X}$, the possible range of the y is restricted by means of the fuzzy set u_i, and $\pi_{ij} = u_{ij}$. For the fuzzy set u_i, the membership value given by u_{ij} expresses the degree of resemblance that a word $\mathbf{x}_j \in \mathbf{X}$ holds towards an idealization of the set u_i – the prototype $\mathbf{v}_i$. In other words, given a known and precise value of a variable $y = \mathbf{x}_j$, the fuzzy membership u_{ij} represents an estimate of the extent to which $\mathbf{x}_j$ is compatible with the concept represented by u_i. By contrast, for an ill-known piece of data y, the possibility distribution π_i can be used to describe and restrict the possible values that y may take in $\mathbf{X}$. Dubois and Prade (1993) point out that π_{ij} estimates the possibility that the variable y is equal to $\mathbf{x}_j$, given the incomplete state of knowledge that y is in u_i. For an exposition of common misunderstandings concerning fuzzy set theory, possibility theory and probability theory, see (Dubois and Prade, 1993).

The Similarity Based Interpretation The perspective on fuzziness that we have put forth in this section corresponds to what Ruspini and Francesc (1998) call a *similarity based* interpretation, as opposed to a *probabilistic* interpretation. A fuzzy set is under this interpretation considered a formal tool to describe resemblance between objects and prototypical examples, – "*a numeric measure of conceptual adequacy*" (Ruspini and Francesc, 1998). The similarity based interpretation of fuzziness seems to be very well in sync with our description of semantic categories exhibiting prototypical and peripheral members. Note however, that there are also many other types of perspectives on fuzzy sets, such as the *preference-based* interpretation, connected to the notion of goal-dependent utility. For an overview of different interpretations of the meaning of fuzzy membership functions, see (Bilgiç and Türkşen, 1999).

In the same way as there is similarity-based interpretation of fuzzy sets, we find a similarity-based interpretaion of possibility distributions, and at this point the two notions can perhaps be seen to be more converging. To make this view clearer, we might contrast what is said about the notion of possibility above, to what Dubois and Prade (1993, sec. 3.2) write in relation to "*possibility as similarity*", – paraphrased here in order to make the notation consistent with that of this paper; "There is a whole trend in fuzzy set theory according to which the degree of membership $u_j(\mathbf{x}_j)$ reflects the similarity between $\mathbf{x}_j$ and an ideal prototype $\mathbf{v}_j$ of u_j (for which $u_j(\mathbf{v}_j = 1)$. This interpretation of partial membership is clearly related to the relation of distance, and not to probability. Then if a variable y is attached a possibility distribution $\pi = u_j$, $y = \mathbf{x}_j$ is all the more possible as $\mathbf{x}_j$ looks like $\mathbf{v}_j$, is close to $\mathbf{v}_j$."

Membership Functions and Protoypes The concept of fuzzy sets with graded memberships seems to provide a very fitting frame for modeling the properties of semantic word classes that we discussed in the previous section. We also touched upon the idea that, in a semantic space model, graded memberships might be expressed on the basis of a point's distance to a class centroid. It seems that an appropriate approach then, would be to construct fuzzy membership functions on the basis of the distance between a word vector and a class prototype. As an example of such a membership function, Zimmermann and Zysno (1985) suggest using

$$u_i(\mathbf{x}_j) = \frac{1}{1 + f(d(\mathbf{x}_j, \mathbf{v}_i))} \tag{3.4.2}$$

where f is some function of the distance d between a prototype $\mathbf{v}_i$ and an object $\mathbf{x}_j$. The membership function devised by Zimmermann and Zysno (1985) (reformulated in equation (3.4.2) to fit our particular problem setting), is motivated by their empirical studies where human subjects are asked to compare a given object with a certain prototype or imaginable ideal (see Bilgiç and Türkşen, 1999).

Given such a fuzzy membership function u_i, it might also be incorporated in the calculation of the class prototype $\mathbf{v}_i$ itself. Instead of letting all class members have an equal say when defining the centroid, and instead of relying on a single prototypical class exemplar, the contribution of each exemplar $\mathbf{x}_j$ when forming the prototype can be weighted by its membership u_{ij}. The centroid calculation given by equation (3.4.1) can thus be seen to be a special instance of a more general definition of prototypes as

$$\mathbf{v}_i = \frac{\sum_{j=1}^{k} u_{ij}\mathbf{x}_j}{\sum_{j=1}^{k} u_{ij}} \tag{3.4.3}$$

where the varying degrees of membership u_{ij} determines the varying degrees of influence that the words have on the prototype (see e.g. Bezdek, 1981).

Elicitation Methods The basic idea so far, is that similarity, and thereby membership in a fuzzy conceptual class, can be expressed as a function of distance in the space. However, the specific method for *eliciting* the membership functions u_i, is of course also a fundamental question. Bilgiç and Türkşen (1999) note how different interpretations of fuzziness call for different methods for constructing the membership functions, and vice versa. In the empirical studies that motivates the model for vague concepts formulated by Zimmermann and Zysno (1985), as mentioned above, *people* are used as the source of the membership function (Bilgiç and Türkşen, 1999). In our case, however, we need to generate such membership functions *automatically*, from a set of given training data. One such type of automatic elicitation methods, is represented by the approach of *fuzzy clustering*, – a special instance of the soft clustering methods. A fuzzy clustering model assumes that words have partial or distributed memberships in fuzzy subsets on $\mathbf{X}$, as opposed to the crisp classes assumed by conventional clustering models. For a set of c fuzzy classes we then seek to automatically characterize the value of each u_i for each word $\mathbf{x}_j \in \mathbf{X}$.

In this section we have reviewed a few general properties that seem to hold for semantic classes, and thereby established a few features that would correspondingly be desirable in a model of such classes. In the next chapter we describe and apply various fuzzy clustering methods to our noun–context data and see whether they can accomodate these properties. We present a variety of clustering methods in more detail, including both hard and fuzzy algorithms. As we present the various methods, we gradually develop the clustering schemes that we apply to the noun data set. The clustering of the association vectors in $\mathbf{X}$ is done by initially applying a (hard) bottom-up hierarchical procedure, in order to construct good initial prototypes $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$. We then apply various fuzzy clustering methods to $\mathbf{V}$ and $\mathbf{X}$, in order to construct a set of fuzzy classes of nouns. We describe and apply three different fuzzy approaches, – the *fuzzy c-means* method, *possibilistic c-means*, and the *possibilistic prototype classifier*, and compare their results.

Chapter 4

Clustering

As touched upon in section 3.4.1 of the previous chapter, there are various types of clusters or partitions that may be defined on a given data set. The most common kind of partitioning is probably what is known as a *hard* clustering. The memberships within a hard partition are *crisp*, in the sense that every object belongs to only one cluster. Other clustering schemes yields a *soft* partition, where an object might belong to several clusters with varying degrees of certainty. A third type of memberships are found within *disjunctive* clusterings, in which objects are assigned multiple memberships in overlapping clusters. This latter type is less often seen, and will not be covered here.

There are also algorithmic variations regarding how these partition or clusterings are defined. The crisp and singular memberships are typical of the *hierarchical* methods, while graded memberships are more commonly found within the *iterative* or *partitional* approaches. We will see examples of such soft partitioning schemes from the particular family of algorithms known as *c-means*, presented under the heading of Partitional Clustering in 4.2.

Although the various clustering procedures can be split up and divided in many different ways, these distinctions provide the main lines that can often be found to separate the methods; the type of *memberships* they produce, and the type of *procedure* by which the clusters are formed.

$$\text{Memberships}\begin{cases}\text{Hard}\\ \text{Soft}\\ \text{Disjunctive}\end{cases} \quad \textit{and} \quad \text{Procedure}\begin{cases}\text{Partitional}\\ \\ \text{Hierarchical}\begin{cases}\text{Agglomerative}\\ \text{Divisive}\end{cases}\\ \\ \text{Hybrid}\end{cases}$$

The following sections present the particularities of some selected clustering models in more detail, while also reviewing the results of applying them to the data set of context profiles for nouns. Recall that the noun data set is given by the association matrix $\mathbf{X}$ that is defined for the semantic space $\langle \mathbf{F}, A, s\rangle$. As described in section 3.2, the association vectors in $\mathbf{v}_i \in \mathbf{X}$ are obtained by applying the weighting function A, which is based on the logg odds ratio, to the co-occurrence vectors in $\mathbf{f}_i \in \mathbf{F}$.

The word clustering scheme that we develop has an initial phase of *agglomerative* clustering on $\mathbf{X}$, followed by varieties of *fuzzy c-means* and *possibilistic* *c-means* clustering, in addition to a novel approach based on one-pass fuzzy classification. This means that, in all the three clustering schemes that we apply, we take a *hybrid* approach. An initial step of *hard* and *hierarchical* clustering is followed by a second step of *fuzzy* (soft) and *partitional* clustering. The output of step one is a set of preliminary prototypes which provide the input for step two. The different methods are described in turn over the next sections. We first give an overview of hierarchical methods in general, and then describe the particular approach taken in the initial phase of the noun clustering. We then turn to a general presentation of the partitional methods, before we finally review the application of various types of fuzzy partitional procedures to the noun data and the output of phase one.

4.1 Hierarchical Clustering

The hierarchical algorithms can be subdivided into *divisive* and *agglomerative* methods. The former kind starts by regarding all k objects as part of a single cluster, and then splits the groups top-down into smaller and smaller clusters. The divisions ultimately result in k singleton clusters, if no other stopping criterion is defined. Agglomerative methods follow the opposite strategy, and merges objects in a bottom-up fashion. At the outset, each object constitutes a cluster of its own. The clusters are then merged until there only remains one cluster containing all the objects. At each stage of the analysis, the hierarchical procedures attempt to find the optimal step, as defined by some measure of cluster similarity, and split or merge the pair of clusters that will produce the most coherent partitioning (see Everitt et al., 2001, ch. 4).

The trace of the successive divisions or fusions performed by the hierarchical methods, yields a nested sequence of partitions. This is often visualized as a binary *tree structure* known as a *dendrogram*. Each node in the tree represents a cluster, and its children show which two clusters are joined or separated. Nodes at higher levels of the tree represent increasingly general clusters of decreasingly similar objects.

A hierarchical method actually produces several partitions then, – one for each level of the tree. Note however, that this complete structure of nested partitions does not define any particular clustering of the data. In order to get a final set of clusters, the tree must be cut according to some specified number of root nodes or a similarity threshold (see e.g. Manning and Schütze, 1999, ch. 14). Other algorithms, such as the partitional methods presented later, produce what can be called a "*flat*" clustering (Manning and Schütze, 1999, ch. 14), – the classes do not form any branching structure, and all objects and classes are thought to be "on the same level".

Because hierarchical methods are often found to produce high quality clusterings, they can be attractive even if the data is not thought to posses any kind of inherent "taxonomic" or stratified structure. If the ordered structure resulting from a hierarchial procedure is not thought to be relevant, it can simply be "flattened out"; the subtrees that remain after cutting the tree are just collapsed into flat and disjoint clusters. In the case of our noun clustering, a hierachical method is used only for the purpose of finding good initial prototypes. Since we are only interested in the cluster centroids, the hierachical structure itself is ignored.

The most widely used variety of the hierarchical approaches is probably the agglomerative one (Everitt et al., 2001), which is also the approach taken in the initial phase of the noun clustering. The specific bottom-up clustering of the noun data set is described in section 4.1.3, but a more general presentation of agglomerative methods is first given in the next section.

4.1.1 Agglomerative Methods

A pseudo-code outline of a general agglomerative algorithm is given in table 4.1, yielding a set B of hard hierarchical clusters for the data set $\mathbf{X}$. The *stopping criterion* Λ is typically just a test that returns true as long as the number of root nodes (i.e. the number of clusters) is below some specified threshold $1 \leq \lambda < k$ (where the default $\lambda = 1$ means no cut-off).

Parameters:
$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$
sim: $P(\mathbf{X}) \times P(\mathbf{X}) \rightarrow \Re$
stopping criterion Λ

for $i = 1$ **to** k **do**
  $b_i \leftarrow \{\mathbf{x}_i\}$

$B \leftarrow \{b_1, \ldots, b_k\}$
$j \leftarrow k + 1$

until $\Lambda(B)$ **do**
  $(b_h, b_i) \leftarrow \underset{(b_m, b_n) \in B \times B}{\arg\max} \{sim(b_m, b_n)\}$
  $b_j \leftarrow b_h \cup b_i$
  $B \leftarrow (B \setminus \{b_h, b_i\}) \cup \{b_j\}$
  $j \leftarrow j + 1$
return B

Table 4.1: Agglomerative Hierarchical Clustering

The basic operation of the bottom-up procedures, is to merge the two clusters $(b_i, b_j) \in B \times B$ that are found to be most similar at each stage of the analysis. In section 3.3 we saw how to measure the similarity between individual objects or points in space. When measuring the similarity of clusters on the other hand, we need to define a similarity relation that holds between *collections* of objects. Agglomerative methods are *greedy* in the sense that the pair considered most similar, according to some criterion, is the pair chosen to merge at each step of the analysis (see Cutting et al., 1992). The different ways of defining such *inter-group proximities* are what set the various (hard) bottom-up methods apart. (The inter-group proximity function is written as *sim* in the general algorithm of table 4.1).

Method	AKA	Inter-Cluster Similarity Defined As	Comment
Single Linkage	Nearest-Neighbors, Minimum Method	Minimum distance between any members of each group.	Good local coherence but the so-called chaining effect often produces straggly, elongated and unbalanced clusters (Manning and Schütze, 1999; Everitt et al., 2001). Handles non-elliptical shapes, but sensitive to noise and outliers (Steinbach et al., 2003).
Complete Linkage	Furthest-Neighbors, Maximum Method.	Maximum distance between any members of each group.	Tends to produce tight and compact clusters with equal diameters (Everitt et al., 2001).
Centroid Linkage	UPGMC†	Similarity between mean vectors.	The larger cluster will dominate the merged cluster. Subject to reversals.
Pair-Group Average	UPGMA†	Average pair-wise similarity between each member of one cluster to each member of another cluster.	Compromise between single-link and complete link. Relatively robust.
Weighted Pair-Group Average	WPGMA†	Like UPGMA, but weighted by the size of the respective of clusters.	Produces more even-sized and balanced clusters. Weighted versions can be defined for all the agglomerative strategies.
Within-Groups Average	WGAC GAAC (Group Average Agglomerative Clustering)	Like UPGMA, but average similarity is computed between all objects in the union of the clusters, rather than between the groups, so that within cluster variance is minimized.	

† = acronym in Sneath and Sokal (1973), key: ***W***eighted; ***U***nweighted; ***P***air-***G***roup; ***M***ethod; ***A***rithmetic average; ***C***entroid average

Table 4.2: Agglomerative Clustering Methods.

Inter-Group Proximity Two main routes towards defining the inter-cluster proximities are pointed out by Everitt et al. (2001). One way is to let the proximity between two groups be given by some summary of the individual proximities between their members. Secondly, each group might be described by some single representative, – a prototype, and the inter-group proximity can be defined as the proximity of these representatives. These different routes can, of course, be seen to parallel the different ways of representing a class, that we touched upon in section 3.4.1. A selection of the most standardly used agglomerative strategies were implemented[1] as part of this project. The respective strategies are summarized in table 4.2 along with some brief remarks on the behavior that is commonly reported for the various methods.

It is usually considered a desirable quality of an inter-group proximity measure that it obeys the *ultrametric property* (4.1.1). Everitt et al. (2001) states this condition as

$$\forall\, i, j, k \in B : h_{ij} \leq \max(h_{ik}, h_{jk}) \tag{4.1.1}$$

where h_{ij} is the distance between clusters i and j. Distance is here defined as the height at which two clusters are joined in the dendrogram (see Everitt et al., 2001, ch. 4). If ultrametricity is not obeyed, so-called *reversals* or *inversions* may occur in the dendrogram. This happens when the fusion levels of the hierarchy do not form a monotonic sequence, so that a later fusion might take place at a lower level of dissimilarity than an earlier one (see Everitt et al., 2001, ch. 4). If on the other hand, the similarity function is monotonic, the similarity between sibling nodes in the tree is guaranteed to decrease as one climbs towards the root. Manning and Schütze (1999, ch. 14) thus states that monotonicity is a necessary property if closeness in the tree is to be interpretable as conceptual similarity. Such inversions may occur with for instance the centroid method of table 4.2.

When measuring the similarity between clusters, one also needs to measure the similarity between individual points. In the centroid-based strategy, these points are the prototypes themselves, but in the other methods they correspond to the actual data points. For all the different methods presented in table 4.2 then, the specific similarity function that they are built on top of, can be said to constitute an additional algorithmic parameter. However, for the particular implementation of agglomerative clustering in this project, the approach is somewhat different. In all of the agglomerative strategies given here, except for the centroid method, the function parameter is only implicitly present. This is because we define the methods to work on a precomputed *similarity matrix*. Rather than supplying the data set $\mathbf{X}$ and a similarity function s as input arguments, we instead pass the similarity matrix $\mathbf{S}$. Nonetheless, since $\mathbf{S}$ is of course calculated on the basis of s on $\mathbf{X}$, the similarity function can still be considered as a parameter of the overall procedure. For the agglomerative method finally applied to the nouns, $\mathbf{S}$ is computed for the normalized vectors in $\mathbf{X}$ with the cosine as the similarity measure s (see section 3.3 for further details).

Within-Groups Average Clustering In order to produce an initial set of cluster centers, we apply a procedure based on agglomerative *within-groups av-*

[1] See appendix A for an overview of which of the attached source files implements the various components of the clustering procedures described in this chapter.

erage clustering method (WGAC[2]), followed by a pruning step. The additional pruning procedure merges clusters that are very similar and very small, before the centers are computed. The entire process is more fully described in section 4.1.3 below. The within-groups average method is often reported to yield tight and coherent clusters, and experiments on the noun data set seemed to confirm this impression. The WGAC method is also well behaved with respect to reversals as described above, and is known to produce a monotonic sequence of partitions (see Everitt et al., 2001). Moreover, since we have no a priori reasons for assuming that the clusters should be balanced and even in size, there is no reason for using the weighted versions of any of the methods listed in 4.2.

The within-group average similarity of a cluster $b_j \subseteq \mathbf{X}$ is defined as

$$W(b_j) = \frac{1}{|b_j|(|b_j| - 1)} \sum_{\mathbf{y} \in b_j} \sum_{\mathbf{y} \neq \mathbf{z} \in b_j} s(\mathbf{y}, \mathbf{z}) \tag{4.1.2}$$

Using the WGAC method, the similarity of two clusters b_h and b_i is thereby computed as the average pairwise similarities within their union, $sim(b_h, b_i) = W(b_h \cup b_i)$. In other words, we apply a measure of *intra*-cluster similarity for measuring *inter*-cluster similarity, by considering the two clusters as one. At any given stage, the two clusters b_h and b_i that maximize the value of $W(b_h \cup b_i)$, are the two clusters that are merged. However, Manning and Schütze (1999, ch. 14) point out that, "*some care has to be taken in implementing the group-average agglomerative clustering*": Since computing the average similarities directly has quadratic complexity $O(k^2)$, and this is done for each of the k fusions, the overall complexity is $O(k^3)$ which may be prohibitive for large data sets. In the next section we briefly review some implementational details concerning the agglomerative method, before we describe the actual noun clustering in section 4.1.3.

4.1.2 Optimizations

Precomputed Similarity As said, we define the bottom-up algorithms to take the precomputed similarity matrix $\mathbf{S}$ as input, rather than the data set $\mathbf{X}$ and the similarity function s. Under this approach, the inter-cluster similarity measures access the pairwise similarities of the individual members by simple look-up. Instead of actually computing the pairwise similarities of the individual objects in the clusters, they simply refer to the similarity matrix. This is obviously also a good way to cut back on redundant computations, since every such pairwise similarity need only be computed once.

As previously mentioned, the complexity of computing the initial similarity matrix is $O(k^2)$. However, we can further save in on computational expenses by noting the property of *symmetry*. Since $\mathbf{S}_{ij} = \mathbf{S}_{ji}$, only one of $s(\mathbf{x}_i, \mathbf{x}_j)$ or $s(\mathbf{x}_j, \mathbf{x}_i)$ needs to be calculated (where s is the cosine). We thereby get away with $\frac{k(k+1)}{2}$ rather than k^2 steps when constructing the initial similarity matrix. The same reduction of complexity can be implemented, of course, when

[2] Although *group average agglomerative clustering* or GAAC is often used to denote this method, as in (Manning and Schütze, 1999), this is often ambiguous as to which of the Unweighted Pair-Group Average and Within-Groups Average method is really intended. To avoid confusion, we therefore prefer UPGMA to denote the former method and use WGAC for the latter.

computing symmetric similarity relations for *any* set of objects, such as the set of clusters B at each stage of the agglomerative clustering.

Automatic Memoization A similar rationale forms the basis of another optimization scheme added to the implementation, – the simple but effective technique of *automatic memoization*. The term "memoization" was coined by Michie (1968), and is a method for *caching* the results of a given function; the value that is computed by a function for some given arguments is stored in a cache, and at subsequent calls to the same function with the same arguments, the value is just retrieved rather than recomputed. This is useful in situations where the same functions are called many times with the same arguments. *Automatic* memoization is, in the words of Hall and Mayfield (1993), "*a method by which an existing function can be changed into one that memoizes*." By memoizing the inter-cluster proximity measures, we are able to speed up the clustering significantly. During memoization we also take advantage of the fact that the inter-cluster proximities are symmetric, so that the caching and retrieval is done independent of argument order.

Symmetric Caching Take the WGAC method as an example. When memoizing the within-groups average measure of equation (4.1.2), we index the computed similarity scores by both of the cluster arguments b_h and b_i. From the outset, the agglomerative methods such as WGAC are *global* in the sense that each pairwise inter-group proximity must be considered each time a pair is merged (see Cutting et al., 1992). Nevertheless, by observing the fact that $sim(b_h, b_i) = sim(b_i, b_h) = W(b_h \cup b_i)$, only one of the argument orders needs to be computed. Moreover, since we never consider merging a cluster with itself, the number of comparisons required for the maximization step at each level of the tree can be reduced to $\frac{|B|(|B|-1)}{2}$. Because of the caching of previous computations, we only need to calculate $|B| - 1$ new similarity relations in each stage of the analysis. In other words, we only need to update the cluster similarity matrix (i.e. the memoization cache) for the elements in the set $B \setminus \{b_h, b_i\}$ towards $\{b_j\}$. This means that, of the total number of $\frac{|B|(|B|-1)}{2}$ similarity relations that need to be considered before each fusion in the tree, only $|B| - 1$ of the values represent new computations. The remaining values have been previously calculated and are therefore found by simple look-ups. If the hierarchical merging continues until only a single root node remains, we get a total of $f = k - 1$ fusions, which amounts to $\frac{f(f-1)}{2}$ such updates.

The technique of memoization is a "time versus memory trade-off" (Hall and Mayfield, 1993). We exchange space for speed. For large data sets, tabulating the results of the similarity computations that we do during clustering will produce very large structures. This results in rather stark memory requirements for the system that the memoized clustering algorithm runs on when applied to large data sets. For our modest noun data set, the initial full-profile similarity matrix alone would be a 3000×3000 table, i.e. a 9000000 components. If the caching can not be done in core and the system must resort to swapping, the trade-off might no longer fall to our benefit. However, as we will see, the property of symmetry can obviously be utilized for reducing storage as well.

Symmetrical Matrices When implementing the proximity matrix and the "memo cache" for symmetric functions, we define a novel data type specialized for square symmetrical matrices. In a square symmetrical matrix, such as the k by k similarity matrix $\mathbf{S}$, it holds that $\mathbf{S}_{ij} = \mathbf{S}_{ji}$ for every i and j. Unnecessary storage can thus be avoided by conflating all these "identical" components. We implement such a symmetrical matrix $\mathbf{M}'$, that represents a $k \times k$ full-profile matrix $\mathbf{M}$, as a linear array. This representation is based on the *row-major order* format, which means that the component $\mathbf{M}_{ij}$ in a two dimesional matrix corresponds to the element $\mathbf{M}'_{j+(ik)}$ of a one-dimensional array, counting from zero. Furthermore, we want the two row-column components indexed by (i, j) and (j, i) in $\mathbf{M}$ to correspond to a *single* element of $\mathbf{M}'$. This is achieved by computing the index of $\mathbf{M}_{ij} = \mathbf{M}_{ji}$ in $\mathbf{M}'$ as

$$(\max(i,j) + k\min(i,j)) - \frac{\min(i,j)(\min(i,j)+1)}{2} \tag{4.1.3}$$

We thereby manage to reduce the size of the matrix from $k \times k$ elements in $\mathbf{M}$ to $\frac{k\times(k+1)}{2}$ elements in $\mathbf{M}'$.

Other Approaches Another optimization scheme that concerns the within-groups average method is described by Cutting et al. (1992) and Manning and Schütze (1999). By utilizing the properties of normalized vectors in combination with the cosine function, this particular version of the algorithm computes the average similarity within a hierarchical cluster in constant time on the basis of the sum vectors of its children. The optimization reduces the complexity of WGAC to $O(k^2)$, but its applicability hinges on the use of a vector space model with the cosine as measure of similarity for normalized vectors. The technique of memoization with symmetric caching is more flexible and can be used for all the inter-cluster proximity measures in table 4.2, and independently of the underlying model.

4.1.3 Finding Initial Prototypes

In the sections that follow this one, we apply various partitional clustering methods to the noun data as given by $\mathbf{X}$, – the association matrix computed for the semantic space $\langle \mathbf{F}, A, s \rangle$. All the partitional methods that we use take a set of cluster centers $\mathbf{V}$ as an additional input argument which is dependent on proper initialization. In order to produce the initial centers, we use a clustering scheme based on WGAC with an additional pruning step to deal with small and similar clusters.

Buckshot There are various other such strategies in use for finding initial centers. One much cited strategy is the *Buckshot* method which is described by Cutting et al. (1992). Buckshot uses some cluster routine such as, say, the hierarchical within-group average method, to find c centers on the basis of a subset of the data. The subset consists of $\sqrt{ck}$ points drawn randomly from the data, clustered until $|B| = c$. The corresponding cluster centers are then typically passed to a partitional method such as k-means.

The main motivation for using the Buckshot scheme is to reduce the running time of an expensive clustering routine. In the case of within-group average clustering, the quadratic time complexity $O(k^2)$ of the optimized version used by Cutting et al. (1992) is reduced to $O(ck)$ using Buckshot (Cutting et al., 1992), which is the same complexity as that of k-means. Cutting et al. (1992) suggests using it in settings such as online reclustering, where speed is paramount.

In our case however, the main concern is accuracy rather than speed, and the initial clustering to compute prototypes is a onetime operation. Moreover, it is in no way given that the random subset used by Buckshot actually contains points that are representative of the possible underlying clusters in the data. Another drawback with using Buckshot is that it is non-deterministic; different centers may be found each time the procedure is run for the same data set, since it relies on a randomly drawn sample (Cutting et al., 1992). It also requires that the number of clusters c is specified in advance.

Fusion Ratio In order to secure good initial prototypes, we instead apply within-groups average clustering of the *entire* noun set $\mathbf{X}$, but with a cut-off for the *ratio* of objects merged, or, equivalently, a threshold for the number of singleton root nodes. The version of the agglomerative algorithm that we define for the noun clustering, thus relies on a somewhat peculiar type of condition for termination. Instead of stopping when the number of clusters $|B|$ is above some specified threshold λ, such as $\lambda = c$ in Buckshot, we terminate the clustering when the ratio of singleton clusters in B reaches a specified threshold $\rho \in [0, 1]$. Given a function *singletons* defined as

(4.1.4) $$singletons(B) = \{b_i \mid b_i \in B \wedge |b_i| = 1\}$$

and a threshold ρ, we define our stopping criterion Λ for the agglomerative algorithm in table 4.1 as

(4.1.5) $$\Lambda(B) \begin{cases} T, & \text{if } |singletons(B)| \geq k\rho \\ F, & \text{otherwise} \end{cases}$$

This means that we only need to perform, at maximum and in the worst case, $(k\rho) - 1$ mergers. The rationale behind the ratio criterion is that it ensures that only the objects that show the strongest degree of similarity are clustered. Recall that the greedy WGAC method is guaranteed to produce a monotonic sequence of partitions. When forming the initial prototypes we thereby only rely on the most "confident" merging decisions. Note that the singletons ratio does not specify the number of classes c directly, since we do not know the history of the other merges. This means that although a threshold or cut-off is employed, the number of clusters c is undetermined.

Pruning the Partition Tree In order to further secure the distinctiveness of the prototypes, a pruning procedure is applied to the clusters resulting from the bottom-up run. The entire pruning procedure is outlined in table 4.3, where the predicate *rec* is defined to be true if its arguments are RNNs (reciprocal nearest neighbors;

(4.1.6)
$$rec(b_i, b_j) \begin{cases} T, & \text{if } b_i = \underset{b_k \in B, k \neq j}{\arg\max}\{sim(b_k, b_j)\} \text{ and } b_j = \underset{b_k \in B, k \neq i}{\arg\max}\{sim(b_i, b_k)\} \\ F, & \text{otherwise} \end{cases}$$

After first removing the singletons, the RNNs among the remaining set of clusters are computed. When measuring the inter-cluster similarity, we use the within-groups average criterion of equation (4.1.2) as we did in the bottom-up step. We then merge all clusters that have a similarity above δ in addition to being RNNs. We do this recursively until no RNN clusters remain that have a similarity greater than δ. This merging is done to ensure that final clusters are well scattered in the space, and to reduce the chance of discovering duplicate senses.

Parameters:
$B = \{b_1, \ldots, b_m\}$
sim: $P(\mathbf{X}) \times P(\mathbf{X}) \rightarrow \Re$
maximum RNN similarity δ
minimum cluster size σ

$B \leftarrow B \setminus singletons(B)$

$R \leftarrow \{(b_i, b_j) \mid rec(b_i, b_j) \wedge sim(b_i, b_j) > \delta\}$

until $R = \emptyset$ **do**
  for all $(b_i, b_j) \in R$ **do**
    $b_k \leftarrow b_i \cup b_j$
    $B \leftarrow (B \setminus \{b_i, b_j\}) \cup \{b_k\}$
  $R \leftarrow \{(b_i, b_j) \mid rec(b_i, b_j) \wedge sim(b_i, b_j) > \delta\}$

$S \leftarrow \{b_i \mid b_i \in B \wedge |b_i| < \sigma\}$
$B \leftarrow B \setminus S$

return $B = \{b_1, \ldots, b_c\}$

Table 4.3: Pruning the Partition Tree

As the final step of the pruning, remaining groups with a size smaller than σ are discarded. We throw away the smallest clusters since they are less likely to yield good and representative prototypes. Note that the elements of the discarded groups are not reassigned to other clusters during the pruning, since this would dilute the final prototypes. The groups that now remain in the set B, can perhaps be seen to resemble the notion of 'committees' employed by Pantel and Lin (2002), as described in section 3.4.1.

Notice that the number of clusters c is also determined through this initial phase of agglomeration and pruning. As previously mentioned, a hierarchical clustering method does not by itself assume any particular number of clusters. The number of clusters c is rather the product of our particular cut-off criterion and pruning procedure just described, together with their associated parameters.

Clustering the Nouns In the actual clustering of the 3000 noun vectors in $\mathbf{X}$, we first apply the within-groups average method with $\rho = 0.5$ for the cut-off function Λ defined as in equation (4.1.5). The bottom-up clustering thus builds a partition tree until half of the root nodes are singletons. In other words, we cluster the data set until half of the clusters of individual words remain unmerged.

We then apply the pruning procedure with $\delta = 0.35$ and $\sigma = 3$. After discarding all singletons in the first step of the pruning, we obtain a set B of 358 clusters. We then recursively merge RNN clusters that have a within-groups average similarity greater than $\delta = 0.35$, and prune away every $b \in B$ for which $|b| < \sigma = 3$. After the final merging and trimming, we have a set of hard clusters $B = \{b_i, \ldots, b_c\}$, where c=167 and about one third of the words in the initial data set are included. Note however, that due to additional pruning after the partitional clustering to follow later, the number of clusters c might still be subject to change and is not ultimately fixed.

Finally, for the pruned set of clusters B we compute the corresponding set of centers $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$. A center is computed as the average feature vector of the members, as given by equation (3.4.1), i.e. $\mathbf{v}_i = \sum_{\mathbf{x}_j \in b_i} \mathbf{x}_j / |b_i|$. This also means, of course, that we ignore the internal hierarchical structure of the clusters and treat them simply as flat "bins" or "buckets", rather than trees. The vectors of $\mathbf{V}$ provides the set of prototypes that we will use, together with the association vectors of $\mathbf{X}$, as input to various types of soft partitional clusterings.

4.2 Partitional Clustering

Partitional or 'flat' clustering methods creates a non-nested, one-level grouping of the data. Most such non-hierarchical techniques work by repeatedly reallocating objects within a partition. The procedures are usually initiated with a set of (often randomly defined) prototypes $\mathbf{V}$ for a pre-determined number of clusters c. The partition is then iteratively refined until some stopping criterion is satisfied. Typically this criterion is given by some globally defined objective function of partition quality. The reassignments continue until this goodness measure reaches some threshold or it ceases to improve between iterations. Because partitional clusterings often work by maximizing some goodness function, or alternatively minimizing a *cost function*, they are often termed *objective function clustering* (Bezdek, 1981) or *optimization methods* (Everitt et al., 2001).

The *K-means* method is perhaps the "vanilla flavor" of the wide choice of iterative clustering algorithms. When considered as part of a larger family of algorithms, known simply as *c-means* clustering, the procedure is also known as *hard c-means* (HCM). It is the c-means family that provides the framework for the particular type of partitional clustering that is applied to the contextual distributions of nouns in this project. We previously mentioned in section 3.4.1 that the deterministic annealing (DA) procedure formed the basis of the soft noun clustering performed by Pereira et al. (1993). Masulli and Rovetta (2002) shows that entropy-constrained clustering by DA, can be included within a broad definition of the c-means family. The partitional method described here can thus be seen as relating to the approach of Pereira et al. (1993) within a more general framework. We start by describing optimization methods in terms of the conceptually simple HCM procedure (section 4.2.2, before we go on to cover some of its soft or fuzzy generalizations (sections 4.2.3 and 4.2.5). But before we do anything at all, we briefly introduce some terminology related to fuzzy sets and partitional clustering.

4.2.1 Fuzzy Sets and c-Partitions

Hard c-Partitions An important notion within c-means clustering, is that of a *c-partition.* The c-partition represents the clusters and the associated membership values defined on the data set. Stated simply, a c-partition of a finite set such as $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, is a set of ck values, arrayed as a $c \times k$ *membership matrix* $\mathbf{U}$ (see Bezdek, 1998). The ith row of $\mathbf{U}$, say $\mathbf{U}_{(i)} = (u_{i1}, \dots, u_{ik})$, characterizes the ith partitioning subset of $\mathbf{X}$. In order to avoid the trivial partitions corresponding to k singletons or one all-inclusive cluster, the range of c is constrained to

$$2 \leq c < k \tag{4.2.1}$$

A c-partition must also obey the following two conditions;

$$\sum_{i=1}^{c} u_{ij} = 1, \quad 1 \leq j \leq k \tag{4.2.2a}$$

$$0 < \sum_{j=1}^{k} u_{ij} < k, \quad 1 \leq i \leq c \tag{4.2.2b}$$

Condition (4.2.2a) requires that the total membership values for each $\mathbf{x}_j \in \mathbf{X}$ sum to unity, and (4.2.2b) means that no subset is empty and no subset is all of $\mathbf{X}$ (Bezdek, 1981).

We say that $\mathbf{U}$ represents a *hard* c-partition of $\mathbf{X}$ if and only if its elements satisfy the additional constraint that

$$u_{ij} \in \{0, 1\}, \quad 1 \leq i \leq c, \quad 1 \leq j \leq k \tag{4.2.3}$$

The requirement of (4.2.3) means that memberships are crisp, and u_{ij} is either zero or one according to whether $\mathbf{x}_j$ is a member of the ith cluster or not. Taken together with condition (4.2.2a), this means that each $\mathbf{x}_j$ is a member of exactly one of the c subsets. Given the constraints in (4.2.1), (4.2.2) and (4.2.3), Bezdek (1981) defines the *hard c-partition space* for the finite set $\mathbf{X}$ as
(4.2.4)

$$M_c = \left\{ \mathbf{U} \in \mathcal{V}_{ck} \;\middle|\; \forall\, i, j : u_{ij} \in \{0, 1\}, \quad \forall\, j : \sum_{i=1}^{c} u_{ij} = 1, \quad \forall\, i : 0 < \sum_{j=1}^{k} u_{ij} < k \right\}$$

where $\mathcal{V}_{ck}$ denotes the vector space of $c \times k$ real matrices over $\Re$.

Fuzzy c-Partitions In much the same way that Zadeh (1965) generalized conventional sets to fuzzy sets, as described in section 3.4.2, hard c-partitions are generalized to *fuzzy c-partitions* (see Bezdek, 1981). A fuzzy partition must also adhere to the general conditions stated in equation 4.2.2 above. But each element u_{ik} of a membership matrix $\mathbf{U} \in M_{fc}$ is required to be in the range $[0, 1]$, rather than being constrained to the set $\{0, 1\}$. In section 3.4.2 we introduced the indicator function of a fuzzy set $u_i : \mathbf{X} \to [0, 1]$, where $u_i(\mathbf{x}_j) = u_{ij}$ defines *the grade of membership* of $\mathbf{x}_j$ in the set $\mathbf{U}_{(i)}$. (As said, we will sometimes use u_i directly to denote the set itself.) In a fuzzy partition it is then possible for an object to have *partial* memberships arbitrarily distributed among c *fuzzy* *subsets* that partitions $\mathbf{X}$. Because of the so-called *probabilistic constraint* given by (4.2.2a) on the columns of $\mathbf{U}$, the total memberships of each $\mathbf{x}_j \in \mathbf{X}$ across the c classes must still sum to 1. This constraint comes from generalizing the conception of a crisp c-partition, and is meant to avoid the trivial solution of all memberships being zero (see Krishnapuram and Keller, 1993). Of course, this also means that $\sum_{i=1}^{c}\sum_{j=1}^{k} u_{ij} = K$. On the basis of (4.2.2) and the additional constraint that

$$u_{ij} \in [0,1], \quad 1 \le i \le c, \quad 1 \le j \le k \tag{4.2.5}$$

the *fuzzy c-partition space* M_{fc}, as first introduced by Ruspini (1969), is defined as the set
(4.2.6)

$$M_{fc} = \left\{ \mathbf{U} \in \mathcal{V}_{ck} \,\middle|\, \forall\, i,j : u_{ij} \in [0,1], \quad \forall\, j : \sum_{i=1}^{c} u_{ij} = 1, \quad \forall\, i : 0 < \sum_{j=1}^{k} u_{ij} < k \right\}$$

with M_c being a subset of M_{fc} (Bezdek, 1981).

Hardening By applying a so-called *hardening* function H, any fuzzy c-partition can be *defuzzified* to produce a hard partition. The hardening function H simply sets the maximum coordinate u_{ij} of each $\mathbf{x}_j \in \mathbf{X}$ to 1, while every other u_{lj} for $l \neq i$, is set to 0 (see Bezdek, 1998). Every object then uniquely belongs to whichever cluster had the highest membership value.

α-cuts If the concept of hardening is seen as a way of connecting hard and fuzzy c-partitions, the concept of an *α-cut* is what connects classical and fuzzy sets. For a given fuzzy set u_i defined on $\mathbf{X}$ and a number α in $[0,1]$, the corresponding α-cut of u_i, is the crisp set u_i^{α} that consists of all members of u_i with a membership degree equal to or greater than α (see Klir and Yuan, 1998). If this last condition is instead formulated more strictly as only "greater than", it is called a *strong* α-cut and denoted $u_i^{\alpha+}$. The α-cut of a fuzzy set u_i is defined as

$$u_i^{\alpha} = \{\mathbf{x}_j \in \mathbf{X} \mid u_i(\mathbf{x}_j) \ge \alpha\} \tag{4.2.7}$$

while the strong α-cut is given by

$$u_i^{\alpha+} = \{\mathbf{x}_j \in \mathbf{X} \mid u_i(\mathbf{x}_j) > \alpha\} \tag{4.2.8}$$

It is also worth observing that we can define a disjunctive clustering of $\mathbf{X}$ by way of taking the α-cut of every $u_i \in \mathbf{U}$, where $\mathbf{U} \in M_{fc}$. Let $\mathbf{U}^{\alpha}$ denote the partition given by computing u_i^{α} for each $u_i \in \mathbf{U}$, where $\mathbf{U}$ is a fuzzy partition of X. $\mathbf{U}^{\alpha}$ would then correspond to a disjunctive clustering of X, – a set of possibly overlapping crisp clusters on X with disjunctive and binary memberships.

4.2.2 Hard c-Means

We now turn to the actual procedures and algorithms by which a c-partition can be defined on a given data set. We start the presentation of c-means algorithms by describing the widely used *hard c-means* method (HCM). A general outline of the method is given in table 4.4 below, which is also similar to a close relative of HCM, – the hard ISODATA[3] algorithm as designed by Ball and Hall (1967). A brief and clear introduction to the ideas of the c-means algorithm can also be found in (Jantzen, 1998).

The HCM procedure yields a hard c-partition on $\mathbf{X}$, as defined by equations (4.2.2) and (4.2.3). For each of the c crisp sets in the partition, a corresponding center or prototype is computed. Each data point is then assigned to the cluster with the nearest center. The center $\mathbf{v}_i$ of a cluster u_i in a hard c-partition $\mathbf{U}$ is simply the mean vector $\bar{\mathbf{v}}_i$, – defined in equation (3.4.1) as $\bar{\mathbf{v}}_i = \sum_{\mathbf{x}_j \in u_i} \mathbf{x}_j / |u_i|$.

HCM is usually initialized by selecting random prototypes in $\Re^n$ for a specified number of clusters c. After each point has been allocated to its closest cluster, as shown in table 4.4, the centroids are updated to reflect their new members. The process continues in this iterative fashion, alternately recomputing the prototypes and reallocating the objects, until some stopping criterion is satisfied. As stated in table 4.4, HCM terminates when the difference between prototypes $\mathbf{V}^t$ and $\mathbf{V}^{t-1}$ of successive partitions, as defined by E_t, is less than a specified threshold ϵ. One can alternatively, or additionally, specify that the procedure must halt after a maximum number of iterations τ.

Parameters:
$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$
number of clusters $1 \leq c < k$
termination threshold $0 < \epsilon$
distance function $D_{\text{err}} : \Re^{cn} \times \Re^{cn} \to \Re$

$t \leftarrow 0$
$V^t \leftarrow$ `initiate-prototypes`

do

  $t \leftarrow t + 1$

  for all $u_{ij} \in \mathbf{U}^t$ **do**

$$u_{i,j} \leftarrow \begin{cases} 1, & \text{if } \mathbf{v}_i = \min\arg_{\mathbf{v}_l \in V^{t-1}} \{\|\mathbf{x}_j - \mathbf{v}_l\|\} \\ 0, & \text{otherwise} \end{cases}$$

  for all $\mathbf{v}_i \in \mathbf{V}^t$ **do**

    $\mathbf{v}_i \leftarrow$ update with equation (3.4.1) and $\mathbf{U}^t$

  $E_t \leftarrow D_{\text{err}}(\mathbf{V}^{t-1}, \mathbf{V}^t)$

until $E_t > \epsilon$

return $(\mathbf{U}^t, \mathbf{V}^t)$

Table 4.4: Hard c-Means Clustering

Sum of Squared Errors Computing the centers according to equation (3.4.1) represent a necessary condition for minimizing the *within-group sum of squared*

[3]ISODATA is an acronym for *iterative, self-organizing data analysis techniques A* (Ball and Hall, 1967).

errors (WGSS), – an extensively used cost function that is also known as a *minimum variance objective* (see Bezdek, 1981). Let $\mathbf{V}$ be a c-tuple of n-dimensional prototypical centers, i.e $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_c) \in \Re^{cn}$ and $\mathbf{v}_i \in \Re^n$ is the centroid vector of cluster $u_i \in \mathbf{U}$. The WGSS objective functional $J_W : M_c \times \Re^{cn} \rightarrow \Re^+$ is defined in (Bezdek, 1981) as

$$J_W(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c} \sum_{j=1}^{k} u_{ij}\, \mathrm{d}(\mathbf{v}_i, \mathbf{x}_j)^2 = \sum_{i=1}^{c} \sum_{\mathbf{x}_j \in u_i} \|\mathbf{x}_j - \mathbf{v}_i\|^2 \tag{4.2.9}$$

where the distance measure $\mathrm{d}(\mathbf{v}_i, \mathbf{x}_j)$ is the Euclidean norm metric $\|\cdot\|$.

A centroid or a prototype can be thought of as a model of the points within its cluster, where the within-group squared distances express the discrepancy between the data and the model. The error incurred by representing each point $\mathbf{x}_j$ by a prototype $\mathbf{v}_i$, is the squared distance between them. The overall error contributed by a given cluster is a measure of local density, and J_W will be small when the clusters are tight and the points are close to their cluster centers (Bezdek, 1981). The output of a c-means algorithm is a pair $(\mathbf{U}, \mathbf{V})$, corresponding to the terminal partition matrix and the prototypes.

Batch Mode Note also that the algorithmic outline in table 4.4 describes the *batch* version of HCM, not the *sequential* or *incremental* version, which has the structure of a competitive learning model (see Bezdek, 1998). In an incremental set-up, the prototypes are continuously updated upon assignment of each individual point. This again, has the effect of making the procedure order dependent, which is often not desirable. Variations in the order of input of the elements of $\mathbf{X}$ may then yield different partitions.

4.2.3 Fuzzy c-Means

In much the same way that we defined a fuzzified generalization of a hard c-partition in 4.2.1, so is the *fuzzy c-means* method (FCM) a fuzzy extension of HCM. The FCM model allows each point to belong to several clusters with a graded membership, and defines a fuzzy partitioning $\mathbf{U} \in M_{fc}$ on $\mathbf{X}$. The least-squared errors criterion $J_m : M_{fc} \times \Re^{cn} \rightarrow \Re^+$ that FCM attempts to minimize is a generalization of J_W (4.2.9) and defined in (Bezdek, 1981) as

$$J_m(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c} \sum_{j=1}^{k} u_{ij}^m\, \mathrm{d}(\mathbf{v}_i, \mathbf{x}_j)^2 \tag{4.2.10}$$

where $\mathrm{d}(\mathbf{v}_i, \mathbf{x}_j) = \|\mathbf{x}_j - \mathbf{v}_i\|$ and $\|\cdot\|$ is any inner product induced norm on $\Re^n$, and $m \in (1, \infty)$ is a weighting exponent. The squared distance between a point $\mathbf{x}_j$ and a prototype $\mathbf{v}_i$ is weighted by the mth power of the membership value of $\mathbf{x}_j$ in cluster u_i (Bezdek, 1981).

Membership in the FCM model are updated according to

$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\mathrm{d}(\mathbf{v}_i, \mathbf{x}_j)}{\mathrm{d}(\mathbf{v}_l, \mathbf{x}_j)} \right)^{2/m-1} \right]^{-1} \tag{4.2.11}$$

while the prototypes are given by

$$\mathbf{v}_i = \frac{\sum_{j=1}^{k} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{k} (u_{ij})^m} \tag{4.2.12}$$

Equation (4.2.11) and (4.2.12), form *necessary* conditions for reaching a global minimum of J_m (Bezdek, 1981). The conditions are not *sufficient* however, but the idea is that local extrema of the objective function represents good clusterings (see Bezdek, 1981).

Parameters:
$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$
number of clusters $1 \leq c < k$
termination threshold $0 < \epsilon$
distance function $D_{\text{err}} : \Re^{cn} \times \Re^{cn} \rightarrow \Re$
maximum number of iterations τ
weighting exponent $1 < m < \infty$
initial prototypes $\mathbf{V}^0$

$t \leftarrow 0$

do
  $t \leftarrow t + 1$
  for all $u_{ik} \in \mathbf{U}_t$ **do**
    $u_{ik} \leftarrow$ update with (4.2.11) and $\mathbf{V}^{t-1}$
  for all $\mathbf{v}_i \in \mathbf{V}_t$ **do**
    $\mathbf{v}_i \leftarrow$ update with (4.2.12) and $\mathbf{U}^t$
  $E_t \leftarrow D_{\text{err}}(\mathbf{V}^{t-1}, \mathbf{V}^t)$
until $E_t > \epsilon$ **or** $t \leq \tau$
return $(\mathbf{U}^t, \mathbf{V}^t)$

Table 4.5: Fuzzy c-Means Clustering

Parameters The weighting parameter m, also known as the *fuzzifier* or *volume control*, controls the extent of "fuzziness" or shared memberships between the clusters of points in $\mathbf{X}$. The FCM model was introduced by Dunn (1973) for the special case of $m = 2$, and then generalized for any $m \in [1, \infty)$ by Bezdek (1973). In theory, FCM converges to a hard c-means solution as $m \rightarrow 1$ (Bezdek, 1981). When $m \rightarrow \infty$ the memberships of each $\mathbf{x}_j$ become uniformly distributed over each u_i, producing partitions that approach $\bar{\mathbf{U}} = 1/c$ (Bezdek, 1998). Similarly, the prototypes $\mathbf{v}_j$ coincide as $J_m \rightarrow 0$. There are no general guidelines as to which m gives the best results, but most users choose $m \in [1.1, 5]$ with $m = 2$ being an "*overwhelming favorite*" (Bezdek, 1998). The larger m is chosen to be, the less influence points with uniformly low memberships will have on determining the centers (Windham, 1982).

Most of the FCM algorithmic parameters are common to HCM, with the main difference being the fuzzifier m. However, this too can be thought of as implicitly present in HCM, with the constant value of 1. Note also that instead of specifying an initial set of prototypes V^0 as in table 4.5, FCM can be initialized with U^0 and by correspondingly shifting the steps in the iterations in 4.5 by one half-cycle. As in HCM, the choice of ϵ, – the lower threshold for E, controls the length of iteration as well as the quality of the terminal estimates. ϵ is usually specified in the interval $[0.01, 0.0001]$, and Bezdek (1998) warns that limit cycles may occur if it is set too small. The choice of ϵ might also be influenced by the particular metric D_{err} that one specifies for measuring E_t.

Although reported to rarely occur in practice, the situation of *singularity* occurs during the FCM procedure if one or more of the distances are zero at any iterate. In this case, the membership function of (4.2.11) can not be calculated. When this happens, we assign zero to each non-singular case, and distribute the memberships uniformly over the singular classes, subject to the constraint $\sum_{i=1}^{c} u_{ij} = 1$ (Bezdek, 1998).

Alternate Optimization As layed out in 4.5, the iterative fuzzy c-means algorithm implements a search scheme known as *alternate optimization* (AO), based on iteration through the necessary conditions for $\mathbf{U}$ and $\mathbf{V}$ at local extrema of J_m (Bezdek, 1998). Keeping the cluster centers $\mathbf{V}$ fixed, the memberships u_{ij} that minimize J_m are given in the update step (4.2.11). In the next turn, keeping instead the membership matrix $\mathbf{U}$ fixed, an optimal prototype that minimize the cost function is the "weighted mean" of the members, as defined in (4.2.12). It is proved in (Bezdek, 1980; Bezdek et al., 1987) that any iterate sequence of FCM, beginning from any initialization in $\Re^{cn}$ (or M_{fc}) for $\mathbf{V}^0$ (or $\mathbf{U}^0$), converges to a local minimum or saddle point of J_m.

Bezdek (1998) points out that AO schemes are essentially split gradient descent methods, and as such can become trapped in local extrema and are dependent on good initializations. There is no general agreement about a good initialization scheme, but some common variations are using the c first distinct points in the data, using c points randomly drawn from $\Re^n$, or in the case of FCM, using the output of HCM. For the application of FCM on the noun data, which we turn to next, we initialize the procedure with the set of preliminary prototypes $\mathbf{V}$ output from the agglomerative clustering and pruning described in section 4.1.3.

4.2.4 FCM: Results and Discussion

In this section we describe an application of a version of the FCM method to noun data in $\mathbf{X}$ and the prototypes $\mathbf{V}$ from the bottom-up pass. Recall that, as described in section 3.2, the association vectors in $\mathbf{X}$ were computed for the semantic space $\langle \mathbf{F}, A, s \rangle$, by applying the association measure A on the feature matrix $\mathbf{F}$. For the FCM partitioning of $\mathbf{X}$, we define a modified version of the algorithm formulated above. Instead of computing the weighted center $\mathbf{v}_i$ for a cluster u_i as in (4.2.12), we only include the members $\mathbf{x}_j$ of the strong α-cut $u_i^{\alpha+}$. The prototypes are thus updated according to

$$\mathbf{v}_i = \frac{\sum_{\mathbf{x}_j \in u_i^{\alpha+}} (u_{ij})^m \, \mathbf{x}_j}{\sum_{\mathbf{x}_j \in u_i^{\alpha+}} (u_{ij})^m} \tag{4.2.13}$$

This additional "sifting" contributes to reduce the influence of outliers and noise when updating the prototypes in each iteration. It also has the effect of making the center computations more efficient, since only a small subset of the vectors in $\mathbf{X}$, depending on the value for α, needs to be added up when forming the center for a given cluster u_i. The more distant vectors, with a correspondingly lower membership, are kept out of the update. The memberships are computed as usual according to equation (4.2.11), and we use Euclidean distance as the distance function d.

It is important to observe that the partition produced by this modified version, of course, only represents an approximation of the FCM model as stated in section 4.2.3. All the same, clustering with the α-cut based prototype updates, yielded better results on our noun data than using the standard FCM method. The benefit gained from the "restricted" prototype updates that we defined in equation 4.2.13 is probably due to the large number of clusters and dimensions of our data set. For each strong α-cut $u_i^{\alpha+}$ used in the center computations, we specify $\alpha = 0.01$, where $1/c = 1/167 = 0.006$ represents the maximally uniform membership distribution across c clusters.

The best results for our noun data seem to be found when specifying the weighting exponent m to be 1.2. For the distance function D_{err} we use the *Supremum norm*, which is simply the maximum distance between two components of $\mathbf{V}^{t-1}$ and $\mathbf{V}^t$. Due to the additional instability introduced by the modified update form in equation (4.2.13), the threshold ϵ, which specifies the minimum difference between iterations, is set to the relatively high value of 0.01. It takes 33 iterations for the modified FCM procedure to terminate when partitioning $\mathbf{X}$ for these parameter values.

The result of running the modified FCM method on $\mathbf{X}$ is a partition matrix $\mathbf{U}$ and a set of prototypes for the corresponding contextual profiles $\mathbf{V}$. However, during the iterative reassignments under FCM, some of the clusters become very similar, while others have become very diluted and have very uniform and low membership values. The resulting partition is therefore pruned in a way that resembles what we did after the initial bottom-up pass described in section 4.1.3. The pruning step that we formulated in table 4.3 for the partition tree, was based on two factors; proximity and size. The pruning step that we now define for the fuzzy c-partition, is based on proximity and a measure of *average fuzziness.*

Fuzziness As an answer to the rhetorical question – "*how fuzzy is a fuzzy set?*", – Bezdek and Sankar (1992b) suggest using, for example, the average membership values within the set as a quantification of the degree of uncertainty that it possesses. The question of amount of fuzziness can be seen as related to a similar question in information theory, concerning the amount of information contained in a given message. Bezdek and Sankar (1992b) further remarks that, "*for fuzzy sets, quantification of the* amount *of imprecision captured depends on the extend to which the supporting objects (as individuals or as a group)* do *or* do not *possess the concept or property represented by the fuzzy set*". In our case, we want a fuzzy set u_i to represent a semantic concept of some sort. u_i would then seem rather unintelligible and incoherent if no object within the set can be said to represent whatever property it is that the set expresses. By identifying clusters with uniformly lower membership values, we can filter out such ill-defined groups.

Since we are dealing with fuzzy sets in the context of a c-partition within a FCM-model, where every word potentially has a share of the membership values in each of the partitioning subsets, it would be of little use to base our measure of average fuzziness on the entire stock of members within a set. Instead, we determine the fuzziness of a set of words $u_i \in \mathbf{U}$, by measuring the average membership values among its most representative members. For a given set u_i, we only include the *10 strongest members* as its most typical exemplars. The average membership strength is then computed for these top 10 objects alone. A *high* degree of fuzziness corresponds to *low* average membership values. We then use this quantity as the basis for pruning the partition in order to filter out the clusters with the greatest degree of imprecision.

Pruning the c-Partition Among the most similar clusters of the partition, – clusters which have the same objects ranked as their most typical members, we find that these also usually have relatively low membership values associated. We try to discard such "duplicates" in much the same way as we did when pruning the partition tree according to the procedure in table 4.3, – by checking for RNNs and specifying some threshold for proximity. While we previously defined RNNs on the basis of within-group average similarity (see equation (4.1.6)), we now use the measure of (Euclidean) distance d between cluster centers. Also similarly to what we did after the bottom-up pass, each pair of RNNs u_i and u_j with a distance $\mathrm{d}(u_i, u_j) < \delta$ are pruned. Instead of merging such RNNs as in 4.3 however, we here simply discard one cluster of the pair. This also means that a larger share of the membership values of the elements in $\mathbf{X}$ will be reallocated and distributed to the remaining clusters, in accordance with the requirement of equation (4.2.2a), which states that each column of $\mathbf{U}$ must sum to unity. When deciding which of the clusters within a pair of RNNs should be discarded, we use as our criterion the measure of average membership among the strongest members of the sets, as described above. The most fuzzy cluster is the one that is discarded.

In a similar fashion as the method described in 4.3, we proceed by recursively pruning clusters until no remaining RNNs are closer than δ. Finally, instead of filtering out the clusters that have a *size* lower than σ, as we do in the procedure of table 4.3, we now remove clusters with *average memberships* less than σ. Of the initial set of 167 centers supplied as input to FCM, 138 remains after the pruning.

Senses in the FCM model Tables 4.6 – 4.10 show examples of sense classes obtained through the FCM clustering. Instead of presenting a random selection of clusters, we show clusters retrieved relative to given targets. For the purpose of displaying the fuzzy word groups, we use a threshold of 0.025 for the lower limit of membership that a word must hold to a given class for it to be included. Furthermore, when displaying a given class, we only show its 10 strongest members, together with their fuzzy membership values. The top caption of each table, shows the target noun for which the clusters are retrieved, e.g. "Target noun: ***kirke*** (**church**)" in table 4.6 below. Above each word group that appears in the table, is the corresponding model index number of the cluster, e.g. "c:46". This id number is followed by the degree of membership that the target noun has in the respective cluster, e.g. "M = 0.0591".

Target noun: ***kirke*** (**church**)			
c:46 M = 0.0591		c:68 m = 0.0561	
0.9998	*by* (city)	0.9763	*hus* (house)
0.9997	*bye* (?)	0.9557	*leilighet* (apartment, flat)
0.2472	*hovedstad* (capital city)	0.1997	*gård* (estate, farm)
0.2272	*landsby* (village)	0.1836	*gate* (street)
0.1777	*bygd* (small town)	0.1626	*hotell* (hotel)
0.1635	*land* (land, country)	0.1496	*hytte* (cottage, hut)
0.1219	*storby* (big city)	0.1492	*villa* (private house)
0.1038	*afrika* (Africa)	0.1091	*butikk* (shop, store)
0.0984	*verden* (world)	0.1005	*park* (park)
0.0905	*europa* (Europe)	0.0985	*bolig* (residence, house)
c:30 m = 0.0388			
0.9943	*klubbe* (mallet, Def Sg/Pl = *klubb*)		
0.9934	*klubb* (society, club)		
0.1914	*parti* (group, party)		
0.1431	*regjering* (government)		
0.1306	*bonde* (farmer)		
0.1213	*høyre* (right, conservative party)		
0.1180	*arbeiderpartiet* (Labour Party)		
0.1169	*politiker* (politician)		
0.1147	*rederi* (shipping company)		
0.1086	*produsent* (producer)		

Table 4.6: Strongest cluster memberships of *kirke* (church)

Many of the clusters and the sense suggestions shown in the tables 4.6 – 4.10 seem immediately intuitive and plausible. The clusters retrieved in table 4.8 can be seen to reflect the meaning of the target *skole* (school) as an educational center and more generally as an institution. The meaning of *school building* however, corresponding to the systematic polysemy often observed between terms denoting *institutions* and *buildings*, is not captured. The reverse situation holds for the senses found for *kirke* (church), shown in table table 4.6.

The cluster assignments shown for *reaksjon* (reaction) in table 4.7, can be seen to reflect its sense as critical response, resistance, and opposition (c:105), as well as the more general sense of consequence and effect (c:15), and the process or act of change and activity (c:133). Yet other cluster–target pairs can be seen to reflect the passage from specific to general, rather than different meanings. The groups found for the target *berlin* (Berlin) range from other specifically named cities (c:51), to general terms for inhabited places (c:46), and finally named countries (c:17).

Target noun: ***reaksjon*** (**reaction**)			
c:133 m = 0.0368		c:15 m = 0.0363	
0.1434	*tilpasning* (adjustment)	0.9983	*virkning* (effect)
0.1298	*aktivitet* (activity)	0.9954	*effekt* (effect)
0.1064	*organisering* (organization)	0.5523	*konsekvens* (consequence)
0.0995	*regulering* (regulation)	0.0939	*skadevirkning* (damage, harm)
0.0946	*samordning* (coordination)	0.0880	*utslag* (outcome, result)
0.0937	*fordeling* (distribution, division)	0.0522	*betydning* (meaning, consequence)
0.0934	*overføring* (transfer, transmission)	0.0460	*gevinst* (profit, gain, prize)
0.0931	*utbygging* (development)	0.0405	*risiko* (risk)
0.0921	*kommunikasjon* (communication)	0.0391	*forskjell* (difference)
0.0890	*overgang* (transition, change)	0.0386	*problem* (problem)
c:105 m = 0.0283			
0.9863	*kritikk* (criticism, review)		
0.9830	*beskyldning* (accusation, charge)		
0.1095	*anklage* (accusation)		
0.0617	*innvending* (objection)		
0.0537	*spark* (kick)		
0.0458	*henvendelse* (request, inquiry)		
0.0409	*angrep* (attack, charge)		
0.0383	*oppfordring* (invitation, appeal)		
0.0380	*søkelys* (focus, spotlight)		
0.0358	*anmodning* (request)		

Table 4.7: Strongest cluster memberships of *reaksjon* (reaction)

Target noun: ***skole*** (**school**)			
c:81 M = 0.0580		c:52 M = 0.0307	
0.9828	*opplæring* (training, education)	0.9548	*institusjon* (institution)
0.9142	*utdanning* (education)	0.9140	*myndighet* (authority)
0.8665	*undervisning* (teaching)	0.3600	*organisasjon* (organization)
0.1384	*utdannelse* (education)	0.2353	*etat* (department, service)
0.1076	*grunnskole* (elementary school)	0.1947	*organ* (organ)
0.0981	*voksenopplæring* (adult education)	0.1836	*bedrift* (business)
0.0918	*etterutdanning* (further education)	0.1286	*instans* (instance)
0.0580	*skole* (school)	0.0976	*arbeidsgiver* (employer)
0.0536	*forskning* (research)	0.0967	*departement* (department, ministry)
0.0524	*spesialundervisning* (special education)	0.0953	*aktør* (player, agent)

Table 4.8: Strongest cluster memberships of *skole* (school)

Target noun: ***berlin*** (**Berlin**)			
c:51 m = 0.0509		c:46 m = 0.0348	
0.9929	*bergen* (Bergen)	0.9998	*by* (city, town)
0.9874	*oslo* (Oslo)	0.9997	*bye* (?)
0.8691	*stavanger* (Stavanger)	0.2472	*hovedstad* (capital city)
0.8412	*trondheim* (Trondheim)	0.2272	*landsby* (village)
0.2781	*london* (London)	0.1777	*bygd* (small town)
0.1932	*hordaland* (Hordaland)	0.1635	*land* (land, country)
0.1853	*kristiansand* (Kristiansand)	0.1219	*storby* (big city)
0.1452	*københavn* (Copenhagen)	0.1038	*afrika* (Africa)
0.1304	*paris* (Paris)	0.0984	*verden* (world)
0.0929	*bydel* (part of town)	0.0905	*europa* (Europe)
c:17 m = 0.0267			
0.9531	*sverige* (Sweden)		
0.9452	*danmark* (Denmark)		
0.9110	*norge* (Norway)		
0.7642	*tyskland* (Germany)		
0.5909	*finland* (Finland)		
0.4724	*frankrike* (France)		
0.4200	*nederland* (Netherlands)		
0.3189	*england* (England)		
0.3120	*russland* (Russia)		
0.2838	*sveits* (Switzerland)		

Table 4.9: Strongest cluster memberships of *berlin* (Berlin)

Target noun: ***rapport*** (**report**)			
c:33 m = 0.0632		c:117 m = 0.0427	
0.9775	*forslag* (proposal, suggestion)	0.9886	*årsberetning* (annual report)
0.9775	*utkast* (draft)	0.9870	*årsregnskap* (annual accounts)
0.3214	*lovutkast* (draft bill, (law))	0.0562	*regnskap* (account)
0.1977	*lovforslag* (bill (law))	0.0427	*rapport* (report)
0.1071	*innstilling* (recommendation)	0.0376	*vedtekt* (regulation, bylaw)
0.1021	*plan* (plan)	0.0311	*grenseverdi* (limit value)
0.0879	*anbefaling* (recommendation)	0.0308	*søknad* (application)
0.0682	*mandat* (mandate)	0.0276	*note* (note)
0.0638	*reguleringsplan* (regulation plan)	0.0262	*resultatregnskap* (income statement)
0.0632	*rapport* (report)	0.0254	*læreplan* (curriculum)
c:24 m = 0.0408		c:1 m = 0.0370	
0.9998	*studie* (study)	0.9984	*bok* (bok)
0.9983	*studium* (study)	0.9906	*bøk* (beech, Pl = *bok*)
0.9280	*undersøkelse* (investigation)	0.7133	*dikt* (poem)
0.1966	*analyse* (analysis)	0.3675	*roman* (novel)
0.1197	*evaluering* (evaluation)	0.1936	*tekst* (text)
0.0807	*kartlegging* (mapping)	0.1435	*brev* (letter)
0.0776	*forskning* (research)	0.1202	*vers* (verse)
0.0672	*beregning* (estimate, calculation)	0.1036	*novelle* (short story)
0.0637	*utredning* (report, exposition)	0.0909	*verk* (work, piece,creation)
0.0635	*måling* (measurement)	0.0898	*tekster* (? 'subtitler', Pl = *tekst*)
c:27 m = 0.0301		c:57 m = 0.0283	
0.9825	*vurdering* (estimate, evaluation)	0.9996	*tall* (number)
0.9675	*gjennomgang* (presentation)	0.9996	*talle* (manure, Def Sg/Pl = *tall*)
0.9284	*drøfting* (discussion)	0.0927	*statistikk* (statistics)
0.3074	*analyse* (analysis)	0.0864	*anslag* (estimate)
0.2984	*drøftelse* (discussion)	0.0615	*resultat* (result)
0.2403	*utredning* (report, exposition)	0.0392	*prognose* (prognosis)
0.1652	*kartlegging* (mapping)	0.0314	*tabell* (table)
0.1283	*beregning* (estimate, calculation)	0.0286	*opplysning* (information)
0.1110	*omtale* (mention)	0.0283	*rapport* (report)
0.1042	*avveining* (priority, weighting)	0.0273	*funn* (find, discovery)

Table 4.10: Strongest cluster memberships of *rapport* (report)

Although many of the clusters seem quite sensible, many of the groups are admittedly rather noisy and incoherent. Some of the word groups are also overly similar, such as clusters c:24 and c:27, that are found to cover the target *reaksjon* (reaction) in table 4.7. Additionally, problems of a more fundamental kind are found if we also consider the values of the memberships, instead of looking at the words in isolation. One immediate danger sign is given by the slopes of the curves that are formed by the membership distributions within the classes. They seem to be either very flat or to decline very steeply. As an example of the latter type, consider the members within cluster c:57 in relation to *rapport* (report) in table 4.10. A few words reign supreme at the top of the list and dominate the cluster, while the memberships for the other words fall off very rapidly. For the words that rank at the top of such clusters, we see that only the most dominant sense can be discovered since their membership values are close to 1. This means that not much of their total membership is left to be distributed to other clusters. Recall that, due to the probabilistic constraint on memberships in equation (4.2.2a), the total membership values of each word must sum to unity. This leads to a situation were the membership of a word in a given class, is also dependent on its memberships in *other* classes.

In some of the clusters, we find instead that the curves formed by their membership distributions, seem to level out right from the start. One such example is cluster c:133 in table 4.7 for the classes assigned to the noun *reaksjon* (reaction). The values are uniformly low, and no members stand out more clearly than others. The general low memberships also means that a very large number of words are weakly associated with the cluster. As a related point, note that since the membership value for a word in a given cluster is also conditioned by the memberships it holds to every other cluster, the value is thereby also dependent on the number of classes itself. Since we operate with a fairly large number of clusters, this will make the average membership values quite low.

Both of the types of membership curves described above makes it very difficult to settle on a reasonable threshold for delimiting classes that are assigned to a word. It also muddles the interpretation of the the memberships. In section 3.4.2, we described the similarity based interpretation of fuzzy membership values. Recall that in Zadeh's (1965) formulation of fuzzy sets, the memberships were intended to denote degrees of belonging or typicality. This provided much of the basis for why we wanted to adopt the notion of fuzzy sets for modeling semantic categories. Nevertheless, the membership values of the clusters shown above, do not seem to accord well with the notion of typicality that we initially set out to capture, as described in section 3.4.

The Probabilistic Constraint As said, according to equation (4.2.2a), each word column of the partition matrix must sum to 1. Krishnapuram and Keller (1993) point out that the probabilistic constraint on $\mathbf{U}$ is too restrictive if the memberships are meant to represent "*degree of compatibility*". This constraint may give meaningful results in applications where we want to interpret the membership values as probabilities or as degrees of sharing. By contrast, in our case we want the memberships to indicate typicality or resemblance towards a prototype, rather than probabilities. Krishnapuram and Keller (1993) note that the memberships under such a typicality interpretation, should be *absolute*, and not *relative* such as the memberships generated under the constraints of the standard FCM model. As previously pointed out, the memberships of words in the FCM partitioning of $\mathbf{X}$, are "relative" in the sense that the grade of membership that is assigned to a word in a given class, is not only dependent on its distance from the respective prototype, but on its distance to prototypes of other clusters as well. However, if the membership values are to be construed indicators of similarity and typicality, the membership value of a point in a given class should not depend on its memberships in other classes.

Krishnapuram and Keller (1993) offer various illustrated examples that clearly depict the problems that arise due to the membership restrictions of the standard FCM model. Some of these situations are similarly sketched in figures 4.11 and 4.12 below. The cluster centers are marked as hollow circles, while the object points are drawn as smaller and solid dots.

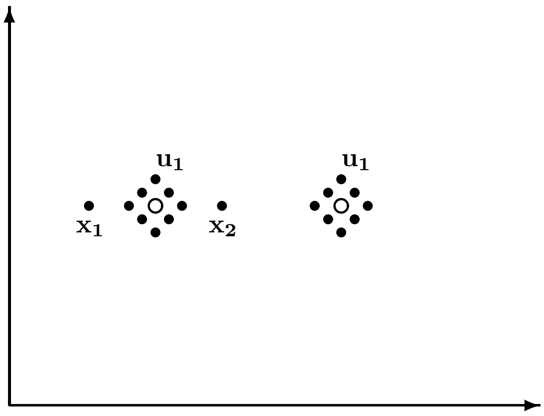

Table 4.11:

Consider the points $\mathbf{x}_1$ and $\mathbf{x}_2$, and the clusters u_1 and u_2 in figure 4.11. Which of $\mathbf{x}_1$ and $\mathbf{x}_2$ is the more typical member of class u_1? Intuitively, we want $\mathbf{x}_1$ and $\mathbf{x}_2$ to be judged as equally typical (or untypical) of cluster u_1, by virtue of being equally distant from the prototype. Still, due to the probabilistic constraint, $\mathbf{x}_1$ will be assigned a greater strength of membership than $\mathbf{x}_2$, since it does not have to "give away" as much of its membership to u_2. Since $\mathbf{x}_2$ is closer to the prototype of u_2, a greater part of its membership is shared between the two clusters.

Krishnapuram and Keller (1993) also illustrate how a related situation arise in the event of "noise points" or outliers. Consider the clusters formed by the data points in figure 4.12. Intuitively, neither $\mathbf{x}_1$ nor $\mathbf{x}_2$ would seem to be good candidates for any of the clusters, but even so, point $\mathbf{x}_1$ is clearly a much worse contestant than point $\mathbf{x}_2$. All the same, under the probabilistic membership restriction in models such as FCM, both points will be assigned memberships of 0.5. Krishnapuram and Keller (1993) remark that this not only shows that the constrained memberships are unrepresentative of the degree of belonging, but also that they are unable to distinguish between a "*moderately atypical*" and an "*extremely atypical*" member. They further comment that this situation may not be critical in the setting of crisp classification or in applications were the ultimate goal is a hard partition, – i.e. the fuzzy model is hardened to produce a crisp clustering. In "*fuzzy set applications*" on the other hand, such as our task of modeling word senses, this situation may not be appropriate. Krishnapuram and Keller thus reformulate the FCM model to generate memberships that can be given a clearer typicality interpretation. The result is the method of *possibilistic c-means* clustering (PCM). In the next section we briefly review the ideas that motivate and constitute the possibilistic c-means model, before we describe the application of PCM to the noun data in an attempt to produce a more satisfactory representation of semantic classes.

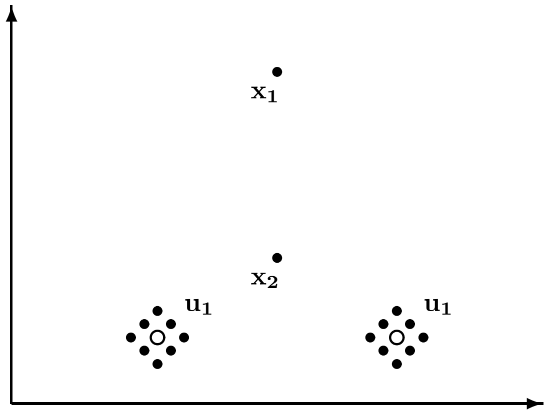

Table 4.12:

4.2.5 Possibilistic c-Means

Typicality Revisited Figures 4.11 and 4.12 in the previous section illustrated some potential drawbacks of the membership constraints in FCM: Two points that are equally distant from a cluster, can still have different memberships, and, two points that have equal memberships in a cluster can still lie at a different distance from the cluster center. In an attempt to address these problems, Krishnapuram and Keller (1993) define a possibilistic objective-function clustering in order to "*generate membership distributions that model vagueness*" and where the memberships can more naturally be given a typicality interpretation. A distinguishing characteristic of PCM, as opposed to FCM, is that the membership of a point in a cluster is not relative and depends only on its distance to the respective center (Krishnapuram and Keller, 1996). As part of their motivation for generating membership functions, they also refer to the model for membership functions of vague concepts or classes suggested by Zimmermann and Zysno (1985), which we touched upon in section 3.4.2.

In relation to the PCM method, Krishnapuram and Keller (1996) state that; "*Our approach differs from the existing clustering methods in that the resulting partition of the data can be interpreted as a possibilistic partition, and the memberships values may be interpreted as degrees of possibility of the points belonging to the classes, i.e. the compatibilities of the points with the class prototypes.*" The intended interpretation for the membership values sought through PCM, seems to be closer to the similarity based interpretation of fuzzy memberships that we described in section 3.4.2. Krishnapuram and Keller (1993) also note that the notion of memberships in the PCM model, is actually more in accord with the common fuzzy set theory concept of membership (i.e. as defined independent of the constraints in a fuzzy c-partition).

Possibilistic Partitioning The most important reformulation in PCM with respect to the FCM model, is that the probabilistic constraint on memberships is relaxed. Instead of requiring that $\sum_{i=1}^{c} u_{ij} = 1$ for all $1 \leq j \leq k$, as in equation (4.2.2a), the memberships within a possibilistic partition must simply obey

$$\max_i u_{ij} > 0 \text{ for all } j \text{, and} \tag{4.2.14}$$

$$0 < \sum_{j=1}^{k} u_{ij} \leq k \text{ for all } i. \tag{4.2.15}$$

As in FCM, the fuzzy memberships u_{ij} are, of course, required to be in the interval $[0, 1]$. In the same way as we previously saw the hard and fuzzy partition spaces defined in section 4.2.1, a possibilistic partition space might on the basis of equations (4.2.14) and (4.2.15) be defined as

(4.2.16)

$$M_{pc} = \left\{ \mathbf{U} \in \mathcal{V}_{ck} \;\middle|\; \forall\, i, j : u_{ij} \in [0, 1], \quad \forall\, j : \max_i u_{ij} > 0, \quad \forall\, i : 0 < \sum_{j=1}^{k} u_{ij} \leq k \right\}$$

with $M_f c$ being a subset of M_{pc}.

The PCM method is derived from a modification J_m^p of the objective function J_m that one seeks to minimize in FCM, which is defined by Krishnapuram and Keller (1993) as

$$J_m^p(\mathbf{U}, \mathbf{V}; w) = \sum_{i=1}^{c} \sum_{j=1}^{k} u_{ij}^m \, \mathrm{d}(\mathbf{x}_j, \mathbf{v}_i)^2 + \sum_{i=1}^{c} w_i \sum_{j=1}^{k} (1 - u_{ij})^m \tag{4.2.17}$$

where w_i are "suitable positive numbers" to be chosen by the user. Commenting on the function stated in (4.2.17), Krishnapuram and Keller (1993, p. 101) write: "*The first term demands that the distances from the feature vectors to the prototypes be as low as possible, whereas the second term forces the u_{ij} to be as large as possible, thus avoiding the trivial solution*".

While the prototype update function remains the same as for FCM and is computed according to equation (4.2.11), the memberships in PCM are calculated according to equation (4.2.18) below.

$$u_{ij} = \left[1 + \left(\frac{\mathrm{d}(\mathbf{x}_j, \mathbf{v}_i)^2}{w_i} \right)^{1/m-1} \right]^{-1} \tag{4.2.18}$$

In each iteration, the update of u_{ij} depends only on the distance of $\mathbf{x}_j$ from $\mathbf{v}_i$, since it does not have to obey the probabilistic constraint of FCM. Krishnapuram and Keller (1993) thus point out that the possibilistic approach is *intrinsically fuzzy*, since the memberships would not be hard even if there was only one class defined on the data set. It is also claimed to be more immune to noise points, since they will be assigned a low degree of belonging in all clusters.

The value of the weight terms w_i determine the "*bandwidth*" of the membership function of a cluster or the "*zone of influence*" of a point with respect to a given cluster: If the distance $d(\mathbf{x}_j, \mathbf{v}_i)^2$ is large when compared with w_i, then $\mathbf{x}_j$ will have little influence over the center computation of u_i (see Krishnapuram and Keller, 1993, 1996). Krishnapuram and Keller (1993) suggest computing the weights w_i according to either equation (4.2.19a) or (4.2.19b) below.

(4.2.19a) $$w_i = L\left(\sum_{j=1}^{k}(u_{ij})^m\,\mathrm{d}(\mathbf{x}_j, \mathbf{v}_i)^2\right)\left(\sum_{j=1}^{k}(u_{ij})^m\right)^{-1}$$

(4.2.19b) $$w_i = \frac{\sum_{\mathbf{x}_j \in \mathbf{U}_{(i)\alpha}} \mathrm{d}(\mathbf{x}_j, \mathbf{v}_i)^2}{|\mathbf{U}_{(i)\alpha}|} \qquad \text{where } \mathbf{U}_{(i)\alpha} \text{ is an } \alpha\text{-cut of } \mathbf{U}_{(\mathbf{i})}$$

Equation (4.2.19a) makes the weight terms w_i proportional to the average fuzzy intra-group distance to the centroid within cluster u_i, and L is typically chosen to be simply 1 (see Krishnapuram and Keller, 1993). The alternative approach of equation (4.2.19b), means that the value is computed only on the basis of the strongest members which fall within the α-cut of u_i. In both cases we see that the quantity is related to the fuzzy cluster variance, and can perhaps more intuitively be understood as an expression of the radii of the clusters.

Krishnapuram and Keller (1996) point out that the weighting exponent m has a somewhat different role in PCM than in FCM. While m determines the degree of sharing of memberships in FCM, it determines the possibility of all points completely belonging to a given cluster in the PCM model. In both cases however, decreasing the fuzzifier m means a more rapid decay of the membership function.

The overall procedural set-up of PCM is similar to that outlined for FCM in table 4.5. The only difference is that the membership updates in each iteration are done according to equation (4.2.18) instead of (4.2.11), and that the penalty terms w_i must be estimated as part of the initialization.

PCM Applied to the Noun Data When applying the possibilistic c-means procedure to the noun data, the results are very far from satisfactory. The clusters receive very uniform membership distributions which renders the groups nearly identical. Varying the initialization scheme, using the output of both FCM and the bottom-up pass, and using the different equations for computing the weights w_i, does not make any difference to the homogeneity of the clusters in the resulting partition.

The problems with coinciding centers in relation to PCM is also reported by Barni et al. (1996). After testing PCM on a variety of data sets (based on satellite image data), they conclude that the method seems to have an undesirable tendency to produce coincident clusters and fail to recognize the structure underlying the data (Barni et al., 1996). This tendency of PCM was shown to be persistent also when starting from good initializations.

A re-interpretation of the results and problems encountered in (Barni et al., 1996) can be found in Krishnapuram and Keller (1996), which suggest that the relevant data may have been too "*contaminated*". This may of course also be the case with our noun data. In fact the objects in the noun data set are in many ways "inherently noisy", in the sense that, as we have seen numerous examples of, noise is not only introduced by errors in the sampling process as such, but also by the presence of ambiguity caused by polysemy and homonymy.

Krishnapuram and Keller (1996) furthermore call the case of coincident clusters "*a blessing in disguise*". The point is that coinciding clusters do not necessarily represent a bad result, but may simply suggest that some of the assumptions of the particular clustering model have been violated. Krishnapuram and Keller (1996, p. 390) write that "*By merging coincident clusters after overspecifying c [...] one can, in fact, determine the number of clusters, thus addressing the problem of cluster validity*". This is of course indicative of the pruning that we performed after the bottom-up clustering in section 4.1.3 and after the FCM clustering in section 4.2.4. In the face of the noun partition produced by PCM however, there would simply be too many clusters to merge.

One Step Back In the foregoing sections, we have described the objective-function based approaches of fuzzy and possibilistic c-means clustering. We have also seen that their application to the noun context data do not give satisfactory results in terms of modeling sense classes. Although many of the word classes of the FCM partition seem intuitively informative, and show clear tendencies towards what we are looking for, the corresponding membership values do not easily lend themselves to the type of typicality interpretation that we desire.

The PCM approach seems to lie closer to our aim, but its application to the noun data does not give good results. In the next section, we take a step back and reapproach the problem from another angle. The strategy taken by Masulli and Rovetta (2002), when formulating the soft clustering problem in general terms, is to shift focus away from the minimization of a cost function and instead direct attention to the cluster memberships directly. In order to reformulate our approach to the semantic categorization task we might benefit from the same shift of focus. In the following section, we change our viewpoint from the objective-function based perspective of c-means clustering, and focus instead on the membership functions. This facilitates a more intuitive approach towards the categorization problem.

4.3 Possibilistic Prototype Classifier

In the same way as in the preceding sections, the agglomerative clustering and pruning performed in section 4.1.3 will be our point of departure. That is, the hard clusters produced by the first step of bottom-up clustering will also here form the basis of a second step of fuzzy class assignments.

However, in contrast to the optimization-based formulations of the problem that we have worked with so far, we develop a more heuristically motivated approach in this section. In fact, instead of construing the semantic modeling task as a clustering problem, we rather approach it as a *classification* problem. More specifically, a set of prototypes $\mathbf{V}$ obtained from the first phase of bottom-up clustering can serve as the basis for a fuzzy classifier that assigns soft labels to the data in $\mathbf{X}$. A classification task typically consists of first *training* a classifier on a set of *labeled* data, and then applying the classifier to assign class labels to *unlabeled* data. A *label* is simply some sort of class marker indicating $\mathbf{x}_i \in \zeta_i$ for a class ζ_i and a data object $\mathbf{x}_j \in \mathbf{X}$.

In this section we define a *possibilistic prototype classifier* (PPC), which defines a fuzzy partition on $\mathbf{X}$ on the basis of $\mathbf{V}$. It is a *classification* task in the sense that we treat the constant vectors of $\mathbf{V}$ as if they were obtained from labeled training data, and do a one-pass assignment of class labels to all the word vectors in $\mathbf{X}$. It is *possibilistic* in the sense that the resulting partition on $\mathbf{X}$ is $\mathbf{U} \in M_{pc}$. It is a *prototype* classification in the sense that class assignments are done on the basis of prototype distance, rather than, for instance, voting by nearest neighbors. Although we reformulate this final phase of the semantic modeling as a classification problem, the overall method is, of course, still a case of unsupervised categorization. Note also that when we talk of “labels”, these will simply be the identities $1 < i \leq c$ of the initial clusters, and not actual symbolic tags.

Prototype Classifiers An example of a simple prototype based classification scheme, is the *nearest prototype classifier* (NPC) (Duda and Hart, 1973). Let $L = \{\lambda_1, \ldots, \lambda_c\}$ be a set of class labels, and let $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ be a set of prototypes, where $\mathbf{v}_i \in \Re^n$ and $p \geq c$. Each prototype is associated with a corresponding crisp class label in the set $L_V = \{\lambda_{V1}, \ldots, \lambda_{Vp}\}$, where $\lambda_{Vi} \in L$. NPC then operates by assigning each given vector $\mathbf{x} \in \Re^n$ to the same class as its nearest prototype.

As pointed out by Kuncheva and Bezdek (1997), the diffferent ways of obtaining the set of prototypes correspond to different classification rules. $\mathbf{V}$ is typically derived from a training set $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$, where $\mathbf{Y} \subset \Re^n$ and each y_i is associated with some label $\lambda_j \in L$. If $\mathbf{V} = \mathbf{Y}$, the NPC corresponds to a *nearest neighbor rule* (1-NN). If $|\mathbf{V}| = c$ and $\mathbf{v}_i$ is calculated as the mean vector for the members y_j of a group b_i (i.e. $\mathbf{v}_i$ is the mean of every y_j labeled λ_i in $\mathbf{Y}$), then NPC works as a *minimum distance* (1-NP) classifier (Duda and Hart, 1973).

As we saw in section 4.1.3, we obtain the prototypes $\mathbf{V}$ on the basis of the set of clusters $B = \{b_1, \ldots, b_c\}$ given by agglomerative clustering of $\mathbf{X}$, where $b_i \subset \mathbf{X}$ and $c = 167$. However, for the sake of formulating the classification task, we can treat the members of the classes in B as *labeled* exemplars in an *unknown* sample $\mathbf{Y} = \{\mathbf{y}_i, \ldots, \mathbf{y}_m\}$.

As an attempt to unify various classifiation techniques in a single model, Kuncheva and Bezdek (1997) propose the *Fuzzy Generalized Nearest Prototype Classifier* (FGNPC). Despite being a fuzzy method, FGNPC ultimately performs a *crisp* classification decision. For our semantic categorization however, we want a *soft* classifiaction decision, in the sense that the unlabeled word in $\mathbf{X}$ should be assigned to *multiple* classes with *graded* memberships. In other words, instead of doing a crisp classification based on the nearest prototype, PPC performs a soft classification based on all prototype proximities.

During the classification process, however, the FGNPC method employs a set of *soft labels*. As opposed to the crisp labels used by NPC above, FGNPC assigns a label *vector* to each object that indicates its degree of membership in each of the c classes. Such a fuzzy label vector for a given word can simply be given by its corresponding column in the partition matrix $\mathbf{U}$, which we are already familiar with. $u(\mathbf{x}_j) = [u_{ij}, \ldots, u_{cj}]^T$ denotes the membership degrees of $\mathbf{x}_j \in \mathbf{X}$ across the c classes (Kuncheva and Bezdek, 1997). A fuzzy partition matrix $\mathbf{U}$ can thus be seen to correspond to a *label matrix* for the objects in $\mathbf{X}$.

As pointed out, FGNPC employs only a soft labeling of the objects with the aim of finally making a more accurate crisp classification. Our PPC approach is rather a fuzzy extension of the simple minimum distance classifier. Instead of doing a simple crisp 1-NP classification, we let the soft classification of each $\mathbf{x}_j \in \mathbf{X}$ be a function of its distance to each class prototype $\mathbf{v}_j \in \mathbf{V}$. The value of the ith element in the soft label vector $u(\mathbf{x}_j)$ will be a function of the distance between the word vector $\mathbf{x}_j$ and the corresponding cluster center $\mathbf{v}_i$.

The Membership Function In section 3.4.2, we mentioned the model for vague concepts or classes that is suggested by Zimmermann and Zysno (1985), in which the grade of belonging can be expressed as a function of the distance between an object and some ideal prototype. However, we left unspecified the further properties of this functional relationship. Other psychological studies of concept formation have advocated that similarity should be modeled as an exponentially decaying function of distance in psychological space (Nosofsky, 1986; Shepard, 1987). Given a distance measure d and a sensitivity parameter m, Nosofsky (1986) suggests a Gaussian similarity function s, formulated as

$$s_{ij} = e^{-w\,\mathrm{d}_{ij}^2} \tag{4.3.1}$$

Another model closely related to the Gaussian function of distance that is suggested by Nosofsky (1986) is the "*universal law of generalization*" formulated by Shepard (1987). Shepard (1987) argues that there is a psychological law that relates the similarity of two items to a negative exponential function of the distance between them, which is defined (somewhat simplified) as

$$s_{ij} = e^{-\,\mathrm{d}_{ij}} \tag{4.3.2}$$

The important idea here is, in the words Gärdenfors (2000, p. 21), that "*the similarity between two objects drops quickly when the distance between them is relatively small, while it drops much more slowly when the distance is relatively large*", where the rate of decay depends on the value for the w parameter. A similar approach to constructing a similarity function as in equation (4.3.1), is found in FGNPC, formulated as

$$s(\Delta;\Theta) = e^{\left(-\frac{\Delta^2}{\Theta}\right)} \tag{4.3.3}$$

where Δ is any norm metric on $\Re^n$ and Θ is a set of parameters (Kuncheva and Bezdek, 1997).

The functions above are designed to express the similarity for a given pair of items. In our case, one of the items is, more specifically, a semantic class, while the other is an unlabeled word. Moreover, by implementing the similarity based interpretation of fuzzy memberships in a more literal and direct sense (compared to what we have done in the previous sections), we can let a similarity relation, such as those defined above, directly specify the membership function. In the possibilistic noun classification, we thus define the membership function as

$$u_i(\mathbf{x}_j) = e^{-\frac{\mathrm{d}(\mathbf{x}_j,\mathbf{v}_i)^2}{w_i}} \tag{4.3.4}$$

Decreasing the weight parameter w leads to a more rapid decay of the membership function. This function directly determines the degree of membership that a word holds in a given class. Note that, although the weight parameters $w_i \in W$ may be specified individually for each class, for example by estimates based on within groups variance, we here simply set a common and constant value for w that applies to all classes. What is to count as a reasonable value of w is, of course, highly dependent on the distance measure f. For the noun class assignments we simply use the inverted form of the correlation coefficient as our distance measure, and require that the association vectors of both $\mathbf{X}$ and $\mathbf{V}$ are normalized. Distance can thereby be calculated as $d(\mathbf{x}_j, \mathbf{v}_i) = 1 - (\mathbf{x}_j \cdot \mathbf{v}_i)$ (see section 3.3 for further details). It might seem overly awkward to define a similarity measure in terms of a distance measure, which is, in turn, defined in terms of a similarity measure. However, since the dot-product for normalized vectors has the constant range of $[0, 1]$, this definition makes it easier to determine the value of w empirically. In the results of the noun classifications reported below we use $w = 0.4$.

When we discussed the motivation for the possibilistic c-means method of Krishnapuram and Keller (1993) in section 4.2.5, we established that their unconstrained membership model seems much better suited for the task of modeling semantic word classes and typicality relations. That is, we do not want the fuzzy memberships to be bound by a probabilistic constraint as in fuzzy c-means. We therefore require that the partition matrix $\mathbf{U}$, produced by the possibilistic prototype classifier, must be in M_{pc} as defined in equation (4.2.16). It is only in this sense that the PPC approach is "possibilistic", – the resulting partition $\mathbf{U}$, or rather the soft label matrix $\mathbf{U}$, obeys the same membership requirements as for possibilistic c-means, instead of the probabilistically constrained c-partition of FCM. The graded membership values can thus be given a typicality or similarity interpretation, either in terms of fuzzy sets or possibility distributions. In relation to their PCM model, Krishnapuram and Keller (1996, p. 385) remark: "*It is important to remember that the PCM is just a particular implementation of the general idea of the possibilistic approach. The possibilistic approach simply means that the membership value of a point in a cluster (or class) represents the typicality of the point in the class, or the possibility of the point belonging to the class*".

Association Weighted Prototypes In the single-pass class assignment of PPC, the centers are not iteratively recalculated as in the FCM or PCM clustering. This means that more care has to be taken when positioning the initial prototypes. We have previously mentioned that the hard clusters B, obtained through the initial bottom-up pass, can be seen to resemble the notion of *committees* used by Pantel and Lin (2002) (cf. section 3.4.1). Although we use an altogether different approach to *produce* the initial clusters, they can be seen to resemble the notion of committees by *intention*. Furthermore, the way of calculating prototypes on the basis of these tight initial clusters has a lot in common with how Pantel and Lin (2002) compute the *committee based centers*. Pantel and Lin (2002) first compute the averaged *mean vector* corresponding to the *frequency vectors* of the committee members. In the same way as one weights the feature vectors of words, they then apply a mutual information based weighting scheme to the resulting centroid, in order to produce a "*mutual information vector*" for the cluster.

In a similar vein, when defining the prototypes $\mathbf{V}$ on the basis of B for our fuzzy classification task, we apply the *log odds ratio* association measure A, which we specified for the semantic space $\langle \mathbf{F}, A, s \rangle$ in section 3.2. Recall that the association matrix $\mathbf{X}$ that describes the 3000 nouns in our data set was obtained by applying A to each component of the noun–context co-occurrence matrix $\mathbf{F}$. We now construct a corresponding *frequency matrix* $\mathbf{V}$ for the set of classes that were output from the bottom-up clustering and pruning which were defined in section 4.1.3. Each vector $\mathbf{v}_i \in \mathbf{V}$ is the *unnormalized sum* of the frequency vectors $\mathbf{f}_j \in \mathbf{F}$, corresponding to the elements $\mathbf{x}_j \in b_i$. We then construct a set of association vectors $\mathbf{v}'_i \in \mathbf{V}'$ by applying the weight function A to each element of the class co-occurrence vectors. The centers are finally normalized to have unit length. The set of sum vectors $\mathbf{V}$ can be seen as a class–context co-occurrence matrix, analogous to $\mathbf{F}$, while $\mathbf{V}'$ is an association matrix analogous to $\mathbf{X}$. Finally, after the association weighted prototypes are computed, the single-pass class assignment performed by PPC proceeds by calculating the 167×3000 soft label matrix $\mathbf{U}$ according to the membership function given in equation 4.3.4. The whole procedure is summarized in table 4.13 below.

Parameters:
Frequency matrix $\mathbf{F}$
Association measure A
Clusters $B = \{b_1, \ldots, b_c\}$
Distance function $d : \Re^n \times \Re^n \rightarrow \Re$
Sensitivity weights $W = \{w_1, \ldots, w_c\}$

---

for all $b_i \in B$ **do**
  $\mathbf{v}_i \leftarrow \sum_{\mathbf{x}_j \in b_i} \mathbf{f}_j$

$\mathbf{V} \leftarrow \{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$

for all $\mathbf{v}_i \in \mathbf{V}$ **do**
  $\mathbf{v}'_i \leftarrow \langle A(\mathbf{v}_{i1}), \ldots, A(\mathbf{v}_{in}) \rangle$

$\mathbf{V}' \leftarrow \{\mathbf{v}'_1, \ldots, \mathbf{v}'_c\}$

ensure $\forall\ \mathbf{v}'_i \in \mathbf{V}' :\ |\mathbf{v}'_i| = 1$

---

for all $u_{ik} \in \mathbf{U}$ **do**
  $u_{ik} \leftarrow \exp\left(-\frac{\mathrm{d}(\mathbf{x}_j, \mathbf{v}_i)^2}{w_i}\right)$

return $(\mathbf{U}, \mathbf{V}'')$

Table 4.13: Association Weighted Prototypes and PPC

4.3.1 PPC: Results and Discussion

This section presents some examples of semantic class memberships defined by PPC for various target nouns. The clusters are displayed in the same way as in section 4.2.4 after the FCM partitioning. Above each noun-cluster that is listed for a given target, we see the cluster index (*c:*∗) and the corresponding target membership value (m=∗). Analogously to the k-nearest-neighbor lists (kNNs) that were retrieved for words in section 3.3.2, we here retrieve lists of *k-nearest-prototypes* (kNPs). For each noun shown below we list the 4 most prominent senses (i.e. the 4-NPs), as defined by the soft label matrix $\mathbf{U}$, with the additional requirement that the strength of membership be higher than 0.2. Only the 10 most typical members are included when displaying a given cluster, with each noun member listed together with its respective typicality values. The clusters are displayed in decreasing order of target membership, from top to bottom and left to right.

Section 3.3.2 noted on some "oddities" in the word groups caused by the ambiguous lemmatization of certain word forms in the corpora. Although the source of such quirks may often be quite evident from the Norwegian forms, it may be less so in the corresponding translations. In fact, some of the relations might be quite obscure even to a native Norwegian, seeing that some of the involved lemmas are very marginal in normal language use. We therefore sometimes explicitly mark such cases by indicating the ambiguous forms that give rise to the similarities. As an example, consider the Norwegian words *bil* and *bile*, with the respective meanings of a 'car' and a type of 'broad axe'. These two words can be found listed in table 4.14 below, as parts of a cluster with various types of means of transportation. The two lemmas *bil* and *bile* are ambiguous in all their word forms except for the indefinite singular. We see that the translated entry for *bile* is given as "broad axe, Def Sg/Pl = *bil*", indicating that the definite singular form and plural forms of *bile* can also correspond to the base form *bil* (car). Issues related to extracting the words from tagged corpora is further described in section 2.2.

In relation to the kNN lists retrieved for various target nouns in section 3.3.2, we also mentioned the problem of "selective reading" when interpreting the neighboring words. Although the phenomenon is particularly apparent when translating the nouns from Norwegian to English, the problem of contextually biased interpretation pertains to any attempt to evaluate the semantic coherency of a given set of words. As opposed to kNN lists and hard clusters, the semantics of a word is here more properly characterized by a set of fuzzy classes, and not just a single set of words. Although the problem of selective interpretation is still present, the biased readings are perhaps more warranted against a background of multiple classes and graded memberships. As a related issue, we might reemphasize the proviso stated in section 3.3.2 that the English transcripts given, when displaying the clusters, in no way represent an effort to give complete and exhaustive translations, but are merely meant to provide a "rough guide" to the meanings expressed.

We will not go into the details of interpreting all the individual various sense suggestions, but rather let the classes speak for themselves. We will, however, comment on some general issues related to the creation and interpretation of such soft classes. The intent is to point out various unresolved issues, as well as possible directions of further development. We first discuss the distinction of typicality and memberships, and then briefly look at the relation between the notion of possibility and independence. Furthermore, the next section starts off with a review of some general issues related to the evaluation of unsupervised modeling of word similarity.

The end of this section shows examples of the most salient local contexts for various clusters, in a similar fashion as for individual words at the end of section 3.2, as determined through the association weighting.

Typicality vs. Membership Many of the kNP relations that are associated with the various target words seem very encouraging, and many of the classes themselves appear to be highly coherent. Nevertheless, this particular mode of displaying the results conceals a major difficulty, namely the problem of determining global thresholds. This a general source of difficulty that may also appear at various other levels of the analysis. If our aim was a crisp classification, as in FGNPC, we would simply uniquely assign every word to the class to which it holds the strongest degree of membership, thus obliterating the need for any threshold. The same situation holds when hardening a fuzzy partition as produced through FCM clustering, in order to yield a crisp partition. When dealing with a fully fuzzy partition, on the other hand, we need (for most practical purposes at least) to determine to what degree a given word must be associated with a given class u_i, in order for u_i to be included among its senses. Trouble is soon to follow if we define a single such threshold to apply for all words and classes. To illustrate the problem, consider the two highest ranking clusters for the nouns *hest* (horse) and *gris* (pig) shown in table 4.14 and 4.15 below.

Target noun: ***hest*** (**horse**)			
c:154 m = 0.5746		c:62 m = 0.4791	
0.9711	*bil* (car)	0.8558	*fugl* (bird)
0.9611	*bile* (broad axe, Def Sg/Pl = *bil*)	0.8330	*hund* (dog)
0.7988	*buss* (bus)	0.7990	*katt* (cat)
0.7617	*busse* (buddy, Def Sg/Pl = *buss*)	0.7660	*katte* (cat)
0.7248	*båt* (boat)	0.6261	*slange* (snake)
0.6735	*tog* (train)	0.6039	*slang* (slang, Def Sg = *slange*)
0.6212	*drosje* (taxi)	0.5556	*mann* (man)
0.6152	*fly* (airplane)	0.5293	*dame* (woman)
0.5746	*hest* (horse)	0.4998	*dyr* (animal)
0.5635	*trikk* (tram)	0.4810	*gutt* (boy)

Table 4.14: Strongest cluster memberships of *hest* (horse)

Target noun: ***gris*** (**pig**)			
c:62 m = 0.2507		c:116 m = 0.2433	
0.8558	*fugl* (bird)	0.8008	*fisk* (fish)
0.8330	*hund* (dog)	0.7990	*brød* (bread)
0.7990	*katt* (cat)	0.7939	*kjøtt* (meat)
0.7660	*katte* (cat)	0.6599	*kak* (?)
0.6261	*slange* (snake)	0.6429	*kake* (cake)
0.6039	*slang* (slang, Def Sg/Pl = *slange*)	0.5663	*pølse* (sausage)
0.5556	*mann* (man)	0.5413	*bolle* (bun, bread roll, bowl)
0.5293	*dame* (woman)	0.5153	*melk* (milk)
0.4998	*dyr* (animal)	0.4821	*mat* (food)
0.4810	*gutt* (boy)	0.4648	*vin* (wine)

Table 4.15: Strongest cluster memberships of *gris* (pig, hog, swine)

The two highest ranked sense classes for the noun *hest* (horse), clusters *c:*154 and *c:*62 displayed in table 4.14, seem quite appropriate and can be seen to correspond to its *vehicle* and *animal* sense respectively. Classes with a lower rank than these two, that have associated memberships less than $u_{c:62}(horse) =$

0.4791, seem less appropriate. We might therefore set a threshold to 0.45, and block every sense class with a membership value that falls below this limit.

However, with this cut-off, none of the 2 nearest prototypes of the noun *gris* (pig), corresponding to classes *c:*62 and *c:*116 as displayed in table 4.15, would pass through, rendering the target "senseless", so to speak. If, in order to cover these senses for *gris* (pig), we were to lower the limit to, say, 0.2, then too many classes might be associated with other words, such as *hest* (horse).

Important work remains to be done in relation to delimiting the sense classes for various words on an individual basis. Instead of settling on some global criterion common to all objects, the final sense assignments should be based on individually estimated thresholds. As an interesting aside in this connection, Zipf argues that, as one of the many empirical laws that sorts under his "Principle of Least Effort", the *number of senses* of a given word, is proportional to its number of occurrences or to its rank according to a frequency based order (see e.g. Manning and Schütze, 1999, ch. 1.4).

One might argue that part of the "thresholding problem" described above actually reveals a deeper ignorance pertaining to the overall approach; – typicality is seen from the perspective of classes rather than objects. Consider, as a classic example from linguistic and psychological prototype theory, the case of *penguins*, *robins* and the category BIRD. Imagine a membership function u_B characterizing the category BIRD in a fuzzy partition $\mathbf{U}$, and $\mathbf{x}_P$ and $\mathbf{x}_R$ denoting the representation of penguins and robins respectively. Under a typicality based membership interpretation as maintained in this paper, we would expect the value of $u_B(\mathbf{x}_P)$ to be considerably lower than $u_B(\mathbf{x}_R)$. Penguins are typically regarded as less typical birds than robins. All the same, this does not mean that we want to abandon the fact that penguins really are birds. When conflating the notions of typicality and class membership, an important piece of the puzzle seems to be missing. A better approach might be to construct a separate membership matrix $\mathbf{M}$, in addition to the typicality matrix $\mathbf{U}$ and the prototypes $\mathbf{V}$.

A similar need seems to have been recognized by Pal et al. (1997), who suggest that the notion of membership and possibility might favorably be construed as distinct quantities during the c-means clustering process itself. Pal et al. (1997) propose a method called *fuzzy-possibilistic c-means* (FPCM), designed to overcome "the noise sensitivity defect of FCM", and additionally solve the problem with coincident clusters of PCM (Pal et al., 1997). FPCM produces both memberships and possibilities simultaneously. Such a line of approach might also be interesting to pursue in relation to the task of modeling semantic classes and word senses.

Graded Possibility The idea of a *graded possibilistic model* described by Masulli and Rovetta (2002) is another approach that might be interesting for the task of semantic modeling. The concept of graded possibility is intended as an alternative to the notions of both probabilistic and possibilistic partitionings. Masulli and Rovetta (2002) point out that, intended interpretation aside, each membership u_{ij} of the partition produced by FCM can be seen as formally and mathematically equivalent to the probability that an experimental outcome is one of c *mutually exclusive* events. Under the possibilistic model, however, there is no constraint on the set of membership values, and the memberships can be seen as probabilities over c mutually *independent* events (Masulli and Rovetta, 2002). By contrast, pairs of events are in real life often seen to be neither completely mutually independent nor mutually exclusive. Rather, as remarked Masulli and Rovetta (2002), events can often be found to provide *partial information* about other events. While the standard possibilistic approach implies that all c membership values for a given object are independent, the concept of *graded possibility* suggests that once a given membership value is determined, the remaining $c-1$ values are constrained into an estimated interval *contained in* $[0, 1]$ (see Masulli and Rovetta, 2002).

Salient Class Contexts In section 3.2 we showed examples of how local contexts for various target words can be ranked according to salience. The association measure A, based on the log odds ratio, is applied to the co-occurrence matrix $\mathbf{F}$ in order to produce an association matrix $\mathbf{X}$. We then simply sort the context features for a given target noun t_i according to the salience scores assigned in $\mathbf{x}_i$. Analogously, the most salient local contexts for various sense classes can be obtained from the association weighted prototypes $\mathbf{V}$, as defined in table 4.13. By applying the association measure A to the sum-vector of co-occurrences for words in a (crisp) cluster, we define a prototypical context profile for the entire class. The various contextual features are thus weighted for salience on the basis of a class–context co-occurrence matrix. In tables 4.25 – 4.30 we display examples of a few clusters, represented by the 20 most typical cluster members, succeeded by their highest ranked local contexts according to association strength. These "class sketches" also show that the different semantic classes are best characterized by different *types* of local contextual features, i.e. different types of syntactic and grammatical relations. Note that the salience scores that are reported in these tables pertain to the *unnormalized* prototypes.

Target noun: ***språk*** (**language**)			
c:54 m = 0.9157		c:132 m = 0.4432	
0.9332	*kultur* (culture)	0.9761	*norsk* (Norwegian)
0.9157	*språk* (language)	0.7895	*engelsk* (English)
0.6337	*tradisjon* (tradition)	0.6423	*tysk* (German)
0.5628	*litteratur* (literature)	0.6351	*fransk* (French)
0.5507	*religion* (religion)	0.4804	*samisk* (Lapp)
0.5101	*kunst* (art)	0.4432	*språk* (language)
0.4562	*identitet* (identity)	0.3445	*morsmål* (mother tongue)
0.4475	*samfunn* (community, society)	0.3347	*matematikk* (mathematics)
0.4153	*miljø* (environment)	0.3200	*ord* (word)
0.3910	*tenkning* (thought, thinking)	0.3085	*fag* (subject)
c:86 m = 0.2957		c:29 m = 0.2403	
0.9143	*ord* (word)	0.9127	*uttrykk* (expression)
0.8142	*ting* (thing)	0.7510	*begrep* (notion, conception)
0.7963	*navn* (name)	0.6710	*setning* (sentence)
0.5780	*sang* (song)	0.6690	*ytring* (statement, utterance)
0.5779	*musikk* (music)	0.4715	*utsagn* (statement, assertion)
0.4898	*lyd* (sound)	0.4498	*ord* (word)
0.4785	*vers* (verse)	0.4084	*tekst* (text)
0.4768	*melodi* (melody)	0.3868	*fortelling* (story)
0.4598	*tekst* (text)	0.3727	*tegn* (sign)
0.4138	*dikt* (poem)	0.3614	*formulering* (formulation)

Table 4.16: Strongest cluster memberships of *språk* (language)

Target noun: ***reaksjon*** (**reaction**)			
c:10 m = 0.3834		c:105 m = 0.3427	
0.8885	*tanke* (thought)	0.9933	*kritikk* (criticism, review)
0.8806	*tank* (tank, Def Sg/Pl = *tanke*)	0.6776	*anklage* (accusation)
0.8378	*følelse* (feeling)	0.6768	*beskyldning* (accusation, charge)
0.7318	*tanker* (? tanker, Pl = *tanke*)	0.3921	*angrep* (attack, charge)
0.6250	*kjærlighet* (love)	0.3904	*innvending* (objection)
0.6239	*opplevelse* (experience)	0.3858	*spark* (kick)
0.5888	*glede* (pleasure, happiness)	0.3665	*protest* (protest)
0.5748	*sorg* (sorrow, grief)	0.3654	*oppfordring* (invitation, appeal)
0.5710	*smerte* (pain, ache)	0.3600	*press* (pressure, stress)
0.5476	*lengsel* (yearning, longing)	0.3427	*reaksjon* (reaction)
c:49 m = 0.3181		c:152 m = 0.3145	
0.8494	*faktor* (factor)	0.8708	*virkning* (effect)
0.8112	*egenskap* (quality, property)	0.8607	*konsekvens* (consequence)
0.7797	*trekk* (feature, move)	0.8236	*betydning* (meaning, consequence)
0.7651	*element* (element)	0.7783	*effekt* (effect)
0.6440	*kjennetegn* (mark, characteristic)	0.4903	*utslag* (outcome, result)
0.5010	*aspekt* (aspect)	0.4700	*skadevirkning* (harmful effect)
0.4325	*forutsetning* ((pre)condition)	0.4596	*sammenheng* (connection)
0.4258	*komponent* (component)	0.4340	*årsak* (cause)
0.4214	*svakhet* (weakness)	0.3975	*problem* (problem)
0.4125	*holdning* (attitude)	0.3948	*forskjell* (difference)

Table 4.17: Strongest cluster memberships of *reaksjon* (reaction)

Target noun: ***studio*** (**studio**)			
c:138 m = 0.3261		c:51 m = 0.2861	
0.9306	*hus* (house)	0.9782	*bergen* (Bergen)
0.8525	*leilighet* (apartment, flat)	0.9511	*oslo* (Oslo)
0.6948	*gård* (estate, farm)	0.8103	*stavanger* (Stavanger)
0.6432	*hotell* (hotel)	0.7998	*trondheim* (Trondheim)
0.6364	*butikk* (shop, store)	0.6792	*london* (London)
0.6046	*kafe* (cafe)	0.5968	*hordaland* (Hordaland)
0.6025	*gate* (street)	0.5943	*bye* (?)
0.5858	*restaurant* (restaurant)	0.5910	*paris* (Paris)
0.5803	*hjem* (home)	0.5896	*by* (city, town)
0.5423	*villa* (private house)	0.5871	*københavn* (Copenhagen)
c:148 m = 0.2592		c:71 m = 0.2419	
0.9634	*rom* (room)	0.8550	*sykehus* (hospital)
0.7640	*kontor* (office)	0.7898	*universitet* (university)
0.7303	*kjøkken* (kitchen)	0.7257	*høgskole* (technical college)
0.7068	*stue* (living room)	0.7225	*senter* (center)
0.6097	*værelse* (room)	0.6301	*museum* (museum)
0.5401	*leilighet* (apartment, flat)	0.5704	*høyskole* (technical college)
0.5165	*gate* (street)	0.5163	*institutt* (institute)
0.5021	*trapp* (stairs)	0.5075	*musé* (museum)
0.5012	*hus* (house)	0.4849	*avdeling* (department)
0.4830	*seng* (bed)	0.4354	*institusjon* (institution)

Table 4.18: Strongest cluster memberships of *studio* (studio)

Target noun: ***andel*** (**share**)

c:124 m = 0.6551		c:153 m = 0.4976	
0.9134	*del* (part)	0.8816	*pris* (price)
0.8288	*dele* (divide, Def Sg/Pl = *del*)	0.8578	*prise* (?, Def Sg/Pl = *pris*)
0.7986	*prosent* (percent)	0.8209	*lønn* (reward, pay, wage)
0.6551	*andel* (share)	0.8128	*skatt* (tax)
0.6438	*deler* (? ‘divider’)	0.7597	*avgift* (tax, duty, fee)
0.5836	*halvpart* (half)	0.7442	*rente* (interest rate)
0.5732	*rest* (rest)	0.5735	*kostnad* (cost)
0.5619	*tredjedel* (one-third)	0.5578	*inntekt* (income, earnings)
0.4229	*mesteparten* (the greater part)	0.5521	*lønning* (salary, pay, wage)
0.3486	*gjennomsnitt* (average)	0.5457	*sats* (rate)
c:96 m = 0.4898		**c:21 m = 0.4669**	
0.7728	*beløp* (deficit)	0.8961	*aksje* (stock, share)
0.7659	*overskudd* (profit, surplus)	0.7640	*eiendel* (asset)
0.7318	*underskudd* (deficit)	0.7177	*driftsmiddel* (funding)
0.7306	*egenkapital* (venture capital)	0.7131	*gjeld* (debt)
0.7295	*omsetning* (turnover)	0.6577	*verdipapir* (bonds, shares)
0.6783	*inntekt* (income, earnings)	0.6428	*fordring* (claim, demand, debt)
0.6483	*premie* (prize, award)	0.4819	*instrument* (instrument)
0.6321	*formue* (fortune)	0.4669	*andel* (share)
0.6314	*kostnad* (cost)	0.4631	*investering* (investment)
0.5747	*sum* (sum)	0.4535	*eiendom* (property, estate)

Table 4.19: Strongest cluster memberships of *andel* (share)

Target noun: ***skole*** (**school**)

c:81 m = 0.8127		c:71 m = 0.4082	
0.8697	*opplæring* (training, education)	0.8550	*sykehus* (hospital)
0.8127	*skole* (school)	0.7898	*universitet* (university)
0.7887	*utdanning* (education)	0.7257	*høgskole* (technical college)
0.6002	*grunnskole* (elementary school)	0.7225	*senter* (center)
0.5964	*undervisning* (teaching)	0.6301	*museum* (museum)
0.4658	*barnehage* (kindergarten)	0.5704	*høyskole* (technical college)
0.4437	*utdannelse* (education)	0.5163	*institutt* (institute)
0.4185	*kurs* (course)	0.5075	*musé* (museum)
0.3728	*voksenopplæring* (adult education)	0.4849	*avdeling* (department)
0.3341	*fag* (subject)	0.4354	*institusjon* (institution)
c:147 m = 0.3477		**c:9 m = 0.2913**	
0.9704	*elev* (pupil)	0.8833	*selskap* (company)
0.9075	*lærer* (teacher)	0.8740	*bedrift* (business)
0.7807	*student* (student)	0.7799	*firma* (firm)
0.5718	*rektor* (school principal)	0.7477	*bank* (bank)
0.5296	*personale* (staff)	0.7200	*banke* (bank)
0.5259	*forelder* (parent)	0.6585	*as* (Ltd., Inc.)
0.4506	*medarbeider* (coworker)	0.6442	*eier* (owner)
0.4276	*unge* (young)	0.6375	*foretak* (enterprise)
0.4175	*pasient* (patient)	0.6182	*investor* (investor)
0.4154	*lege* (doctor, physician)	0.5746	*produsent* (producer)

Table 4.20: Strongest cluster memberships of *skole* (school)

Target noun: ***erik*** (**Erik**)			
c:8 m = 0.5258		c:45 m = 0.4180	
0.8120	*georg* (Georg)	0.7711	*jagland* (Jagland)
0.8094	*far* (father)	0.7700	*olsen* (Olsen)
0.8050	*harry* (Harry)	0.7538	*larsen* (Larsen)
0.7833	*jørn* (Jørn)	0.7490	*hansen* (Hansen)
0.7766	*mor* (mother)	0.7478	*andersen* (Andersen)
0.7716	*karl* (Karl)	0.7251	*pedersen* (Pedersen)
0.7701	*ingrid* (Ingrid)	0.7084	*brundtland* (Brundtland)
0.7658	*knøtt* (Knøtt, small child)	0.6836	*clinton* (Clinton)
0.7528	*ole* (Ole)	0.6824	*johnsen* (Johnsen)
0.7491	*espen* (Espen)	0.6820	*nilsen* (Nilsen)
c:38 m = 0.3516		c:66 m = 0.3148	
0.8968	*mann* (man)	0.8649	*svensk* (Swedish)
0.8817	*kvinne* (woman)	0.8355	*svenske* (Swede)
0.8535	*gutt* (boy)	0.7734	*tysker* (German)
0.8532	*jente* (girl)	0.7376	*russer* (Russian)
0.7724	*menneske* (human)	0.7237	*amerikaner* (American)
0.7569	*folk* (people)	0.6824	*danske* (Dane)
0.6818	*dame* (woman, lady)	0.6549	*flo* (Flo, tide)
0.6807	*pike* (little girl)	0.6504	*dansk* (Danish)
0.6489	*nordmann* (Norwegian)	0.6479	*nordmann* (Norwegian)
0.5943	*mor* (mother)	0.6380	num-åring (`num`-year old)

Table 4.21: Strongest cluster memberships of *erik* (Erik)

Target noun: ***sjel*** (**soul**)			
c:93 m = 0.8428		c:55 m = 0.3558	
0.9136	*ånd* (spirit)	0.9350	*hånd* (hand)
0.8428	*sjel* (soul)	0.8933	*hand* (hand)
0.8226	*ånde* (breath, Def Sg/Pl = *ånd*)	0.7918	*ansikt* (face)
0.4058	*gud* (god)	0.7872	*arm* (arm)
0.3934	*dyr* (animal)	0.7571	*hode* (head)
0.3788	*følelse* (feeling)	0.7292	*finger* (finger)
0.3704	*vesen* (being)	0.6846	*skulder* (shoulder)
0.3697	*kropp* (body)	0.6817	*kropp* (body)
0.3686	*menneske* (human)	0.6689	*fot* (foot)
0.3489	*natur* (nature)	0.6628	*ben* (leg, bone)
c:10 m = 0.3392		c:62 m = 0.2907	
0.8885	*tanke* (thought)	0.8558	*fugl* (bird)
0.8806	*tank* (tank, Def Sg/Pl = *tanke*)	0.8330	*hund* (dog)
0.8378	*følelse* (feeling)	0.7990	*katt* (cat)
0.7318	*tanker* (? tanker, Pl = *tanke*)	0.7660	*katte* (cat)
0.6250	*kjærlighet* (love)	0.6261	*slange* (snake)
0.6239	*opplevelse* (experience)	0.6039	*slang* (slang, Def Sg = *slange*)
0.5888	*glede* (pleasure, happiness)	0.5556	*mann* (man)
0.5748	*sorg* (sorrow, grief)	0.5293	*dame* (woman)
0.5710	*smerte* (pain, ache)	0.4998	*dyr* (animal)
0.5476	*lengsel* (yearning, longing)	0.4810	*gutt* (boy)

Table 4.22: Strongest cluster memberships of *sjel* (soul)

Target noun: ***runde*** (**round**)			
c:16 m = 0.6289		c:145 m = 0.4312	
0.9737	*kamp* (fight)	0.9873	*minutt* (minute)
0.7267	*sesong* (season)	0.8821	*sekund* (second)
0.6769	*turnering* (tournament)	0.5888	*time* (hour)
0.6289	*runde* (round)	0.5715	*halvtime* (half-hour)
0.6115	*finale* (final)	0.5336	*kvarter* (quarter)
0.5725	*landskamp* (national final)	0.4312	*runde* (round)
0.4706	*omgang* (round)	0.4051	*døgn* (day, 24 hours)
0.4508	*mesterskap* (championship)	0.3889	*meter* (meter)
0.4267	*oppgjør* (settlement)	0.3812	*uke* (week)
0.4029	*rettssak* (trial)	0.3707	*måned* (month)
c:73 m = 0.3601		c:2 m = 0.2831	
0.9394	*vei* (road)	0.8168	*januar* (January)
0.7691	*tur* (trip, stroll)	0.8136	*mai* (May)
0.6495	*ture* (?)	0.8029	*november* (November)
0.6233	*kilometer* (kilometer)	0.7931	*desember* (December)
0.5327	*meter* (meter)	0.7880	*februar* (February)
0.5230	*mil* (mile)	0.7609	*oktober* (October)
0.4756	*veg* (road)	0.7605	*september* (September)
0.4089	*tog* (train, procession)	0.7548	*april* (April)
0.3952	*sted* (place)	0.7533	*juni* (June)
0.3940	*gate* (street)	0.7357	*mars* (March)

Table 4.23: Strongest cluster memberships of *runde* (round)

Target noun: ***hendelse*** (**event, incident**)			
c:35 m = 0.3887		c:95 m = 0.3776	
0.9793	*ulykke* (accident)	0.9371	*begivenhet* (event)
0.7094	*dødsfall* (death, decease)	0.8225	*jubileum* (jubilee, anniversary)
0.6534	*uhell* (mishap, accident)	0.7138	*jubile* (jubilee, anniversary)
0.6304	*tyveri* (theft)	0.3776	*hendelse* (event, incident)
0.6122	*innbrudd* (burglary)	0.3610	*tema* (theme, subject)
0.5284	*sykdom* (sickness)	0.3389	*arrangement* (arrangement)
0.4208	*drap* (homicide)	0.3101	*konferanse* (conference)
0.4194	*skade* (damage)	0.2877	*samling* (collection, gathering)
0.4098	*overgrep* (assault)	0.2873	*stevne* (meeting, gathering)
0.3943	*kollisjon* (collision)	0.2713	*høydepunkt* (highlight, peak)
c:49 m = 0.3082		c:127 m = 0.3000	
0.8494	*faktor* (factor)	0.9840	*problem* (problem)
0.8112	*egenskap* (quality, property)	0.7803	*konflikt* (conflict)
0.7797	*trekk* (feature, move)	0.6974	*vanskelighet* (difficulty)
0.7651	*element* (element)	0.6702	*krise* (crisis)
0.6440	*kjennetegn* (mark, characteristic)	0.6600	*vanske* (difficulty)
0.5010	*aspekt* (aspect)	0.6108	*utfordring* (challenge)
0.4325	*forutsetning* (assumption, (pre)condition)	0.5764	*kris* (dagger?, Def Sg/Pl = *krise*)
0.4258	*komponent* (component)	0.5744	*uenighet* (disagreement)
0.4214	*svakhet* (weakness)	0.4830	*dilemma* (dilemma)
0.4125	*holdning* (attitude)	0.4731	*problemstilling* (problem statement)

Table 4.24: Strongest cluster memberships of *hendelse* (event, incident)

Cluster c:4	
0.8204	*antall* (number)
0.8089	*omfang* (extent, size)
0.7833	*grad* (degree)
0.7025	*mengde* (quantity)
0.6074	*mengd* (quantity)
0.5662	*størrelse* (size)
0.5558	*utstrekning* (extent, extension)
0.4623	*vekt* (weight)
0.4606	*risiko* (risk)
0.4602	*forbruk* (consume)
0.4553	*etterspørsel* (demand)
0.4449	*utslipp* (emission)
0.4369	*pris* (price)
0.4214	*kostnad* (expense, cost)
0.4191	*volum* (volume)
0.4100	*prise* (?, Def Sg/Pl = *pris*)
0.4079	*produksjon* (production)
0.3938	*andel* (share)
0.3872	*tilgang* (supply, access)
0.3704	*forekomst* (occurrence)

Table 4.25: The 20 most typical members of cluster c:4

	Context Feature		
Rank	Feature Type	Feature Word	Association
1	`adj_mod_by`	`begrenset` (restricted)	4.45
2	`adj_mod_by`	`viss` (certain)	3.65
3	`adj_mod_by`	`total` (total)	3.48
4	`adj_mod_by`	`betydelig` (considerable, significant)	3.44
5	`adj_mod_by`	`øke` (increase)	3.38
6	`adj_mod_by`	`rimelig` (reasonable, moderate)	3.13
7	`adj_mod_by`	`særlig` (special, particular)	3.12
8	`adj_mod_by`	`mulig` (possible)	3.07
9	`pp_mod_of`	`økning` (increase, growth)	2.69
10	`adj_mod_by`	`gjennomsnittlig` (average, mean)	2.68
11	`adj_mod_by`	`enorm` (enormous)	2.66
12	`subj_of`	`variere` (vary)	2.61
13	`subj_of`	`ansette` (employ, hire)	2.50
14	`adj_mod_by`	`vesentlig` (essential, considerable)	2.50
15	`pp_mod_by`	`informasjon` (information)	2.31
16	`obj_of`	`begrense` (reduce, restrict)	2.24
17	`pp_mod_of`	`reduksjon` (reduction)	2.24
18	`obj_of`	`redusere` (reduce)	2.21
19	`adj_mod_by`	`sterk` (strong)	2.11
20	`prep_obj_of`	`bruke_i` (use_in)	1.97

Table 4.26: The 20 most salient local contexts of **cluster c:4**

.

Cluster c:47	
0.9516	*utvalg* (selection, committee)
0.9035	*kommisjon* (commission)
0.8315	*arbeidsgruppe* (work group)
0.7607	*departement* (department, ministry)
0.7599	*komite* (committee)
0.7556	*flertall* (majority)
0.7343	*regjering* (government)
0.7228	*utvalget* (The Committee)
0.7160	*kommisjonen* (The Commission)
0.6887	*mindretall* (minority)
0.6442	*råd* (advice, council, board)
0.6283	*regjeringen* (The Government)
0.6227	*styr* (mess)
0.5727	*styre* (management, committee)
0.5514	*stortinget* (The Norwegian Parliament)
0.5417	*storting* (parliament)
0.5237	*høyre* (right, conservative party)
0.5090	*byråd* (city council)
0.5027	*høyesterett* (Supreme Court)
0.4915	*justisdepartementet* (Department / Ministry of Justice)

Table 4.27: The 20 most typical members of cluster c:47

	Context Feature		
Rank	Feature Type	Feature Word	Association
1	`poss_of`	`forslag` (proposal, proposition)	5.39
2	`subj_of`	`foreslå` (propose, suggest)	4.71
3	`subj_of`	`peke` (point)	3.70
4	`subj_of`	`understreke` (underscore, emphasize)	3.69
5	`subj_of`	`anta` (assume, suppose)	3.45
6	`subj_of`	`vurdere` (evaluate, assess, consider)	3.20
7	`subj_of`	`påpeke` (point out)	3.09
8	`subj_of`	`drøfte` (discuss, debate)	3.01
9	`obj_of`	`opprette` (found, establish)	2.83
10	`pp_mod_by`	`stortinget` (The Norwegian Parliament)	2.65
11	`subj_of`	`uttale` (pronounce, express, state)	2.58
12	`subj_of`	`anse` (consider, regard)	2.56
13	`subj_of`	`utarbeide` (work out, prepare, compose)	2.52
14	`pp_mod_of`	`medlem` (member)	2.50
15	`subj_of`	`fremme` (promote, encourage)	2.48
16	`subj_of`	`foreta` (undertake)	2.43
17	`poss_by`	`utvalg` (selection, committee)	2.38
18	`subj_of`	`be` (ask)	2.37
19	`pp_mod_by`	`medlem` (member)	2.33
20	`subj_of`	`vekke` (suggest, call, wake, excite)	2.32

Table 4.28: The 20 most salient local contexts of **cluster c:47**

.

Cluster c:13	
0.9329	*erstatning* (compensation)
0.8852	*pensjon* (pension)
0.8206	*ytelse* (contribution, payment)
0.7557	*alderspensjon* (old-age pension)
0.6972	*uførepensjon* (welfare)
0.6542	*godtgjørelse* (allowance, compensation)
0.6008	*vederlag* (remuneration, charge)
0.5834	*dagpenger* (daily allowance)
0.5681	*refusjon* (repayment, reimbursement)
0.5487	*lønn* (reward, pay, wage)
0.5220	*tilskudd* (contribution, subsidy, grant)
0.5049	*tilleggspensjon* (supplementary pension)
0.5033	*rente* (interest (rate))
0.4852	*avskrivning* (write-off)
0.4825	*stønad* (subsidy, dole)
0.4823	*utbetaling* (payment)
0.4751	*avgift* (tax, duty, fee)
0.4677	*skatt* (tax)
0.4595	*kompensasjon* (compensation)
0.4411	*inntekt* (income)

Table 4.29: The 20 most typical members of cluster c:13

	Context Feature		
Rank	Feature Type	Feature Word	Association
1	`adj_mod_by`	`årlig` (yearly, annual)	3.79
2	`adj_mod_by`	`full` (full)	3.46
3	`obj_of`	`motta` (receive, accept)	3.46
4	`pp_mod_of`	`ordning` (arrangement)	3.41
5	`pp_mod_of`	`rett` (right)	3.39
6	`obj_of`	`yte` (yield, contribute)	3.32
7	`pp_mod_of`	`krav` (demand, request, claim)	3.25
8	`pp_mod_of`	`beregning` (estimate, calculation)	3.24
9	`pp_mod_of`	`utgift` (expense)	2.93
10	`subj_of`	`beregne` (estimate, calculate)	2.83
11	`obj_of`	`kreve` (demand, request, claim)	2.83
12	`pp_mod_of`	`rette` (?)	2.72
13	`pp_mod_by`	`sektor` (sector)	2.56
14	`pp_mod_of`	`tillegg` (addition, supplement)	2.53
15	`obj_of`	`betale` (pay)	2.48
16	`pp_mod_of`	`regle` (jingle, Def Sg/Pl = *regel*)	2.44
17	`pp_mod_of`	`prinsipp` (principle)	2.39
18	`pp_mod_of`	`regel` (rule)	2.35
19	`subj_of`	`yte` (yield, contribute)	2.34
20	`pp_mod_of`	`krone` (krone)	2.25

Table 4.30: The 20 most salient local contexts of **cluster c:13**

.

Chapter 5

Final Remarks

5.1 Evaluation Issues

Gold Standard Evaluation Unfortunately, due to lack of both time and resources, we are not able to include any systematic and objective evaluation of the word clusterings in this thesis. In fact, quite some work remains do be done before any form of evaluation could be accomplished at all. We have already mentioned the unresolved issue of delineating the number of senses for each word on an individual basis. Of course, such matters would have to be settled before one can carry out any evaluation of the overall method.

Furthermore, in order to assess the quality of automatically derived word classes, one needs to compare the results against some sort of gold standard. One way of evaluating a model of semantic word classes, is to rely on manually crafted resources such as WordNet and Roget's Thesaurus. But as yet there exists no such broad-coverage repository of semantic information for Norwegian. However, the Section for Norwegian Lexicography and Dialectology is, together with the Text Laboratory at the University of Oslo, currently working on developing a Norwegian version of the SIMPLE lexicon (Lenci et al., 2000), that richly encodes various aspects of semantic information about words. Such a resource can give a way of seeing how the automatically derived classes or relations measure up to a manually compiled counterpart, and can also provide a basis for defining such quantities as precision and recall. In the clustering approach to discovering word senses described by Pantel and Lin (2002) the measures of precision and recall are defined on the basis of WordNet *synsets* (i.e. sets of synonymous words corresponding to nodes in the graph encoded by WordNet) and the SemCor[1] corpus.

Moreover, in order to actually measure the quality of a particular algorithm as such, one might argue that a gold standard set of correct senses for a given word, should be defined relative to the particular *corpus* at hand. This is far from a trivial task, of course. This issue also highlights another important issue when assessing the quality and coverage of semantic knowledge derived through distributional methods; the acquired senses are of course highly dependent on the underlying corpus data and the sort of texts that they comprise. This point

[1] Semantic Concordance Corpus, semantically tagged with WordNet senses. (see e.g. Landes and Leacock, 1998).

thus emphasizes the need for having balanced and representative corpora in cases where the goal is to infer "general" word semantics.

Task Driven Evaluation Another important form of gold standard evaluation takes a more goal directed approach to the problem. Bezdek and Sankar (1992a) note that when the output of an algorithm has a well-defined purpose, this can be used as the benchmark for cluster validity. The gold standard would then consist of hand-labeled data which in some way identifies "correct decisions". Pereira et al. (1993) use their class-based model to predict pairs of verb–noun co-occurrences in test data, as an example of such task driven evaluation. Meanwhile, Li and Abe (1998) perform disambiguation of compound nouns and PP-attachments in hand-labeled test data. By way of such goal oriented evaluation, the pertinence of a particular categorization scheme is, more appropriately, judged relative to its intended purpose.

Similarity Relations It would be interesting to separately evaluate the similarity relations that can be directly drawn from the underlying vector space model, such as the nearest neighbor relations in section 3.3.2. When testing the relevance of the word similarities that are revealed by their model, Landauer and Dumais (1997) and Sahlgren (2001) apply the standardized synonym test that forms part of TOEFL (Test of English as a Foreign Language). When presented with a set of multiple choices, the task is to correctly identify the synonym for a given target. It might also be interesting to compare such automatically derived similarity relations to psychological data from human raters that are presented with the same material.

A discussion of various issues related to "gold standard evaluation techniques" is given by Grefenstette (1993), who proposes a method for comparing two knowledge-poor approaches to extracting semantic similarity relations, without being targeted to a specific application. Grefenstette (1993) provides examples on how to measure correspondence of results from automatic methods against hand-created semantic resources such as dictionaries and thesauri.

Tuning the Parameters The form of task driven evaluation described above can serve to tune the parameters of a particular algorithm. By evaluating different results obtained by using the same method one can empirically determine the best parameter values. We have established that much remains to be done with respect to an evaluation of this project. Moreover, this work should be targeted to the various components involved, in order to tune and specify some of the method parameters in a more principled way. By sheer necessity, a lot of simplifying assumptions have been made in this project, in the sense that many parameter values have been more or less arbitrarily specified. We mentioned the problem of determining the number of senses for a given word, but there are of course a range of other variables involved, the values of which should be separately assessed and tuned. Most importantly, we should further experiment with ways of determining the number of clusters c and the number of dimensions or features n. Both of these are central issues that have largely been ignored in this thesis. We should also more closely investigate the impact of the particular choice of association measure when weighting the frequency counts.

Context Features As the rankings of local contexts according to salience scores in sections 3.2 and 4.3.1 indicated, different lexical-syntactic relations seem to be fit for describing and distinguishing different words and semantic classes. Still, it might be interesting to compare the results of clustering on the basis of, say, only verb–argument relations and only adjectival modifications, etc. Moreover, combining all the various feature types in a single vectorial representation might not be the best solution.

We should also experiment with alternative ways of selecting the final feature set. As touched upon in section 3.1.2, there are others and more sophisticated approaches to selecting the features than the simple "frequency approach" taken in this project, for instance by means of global tests of reliability. Moreover, and perhaps most importantly, the initial shallow processing step also needs to be separately evaluated, and a lot can be done in order to improve the feature extraction process as to get more reliable data. Of course, the shallow processing tool implemented ad hoc for the purposes of this project is far from a robust syntactic parser.

5.2 Summary and Conclusions

This thesis has documented a project concerning the unsupervised acquisition of soft semantic classes with the purpose of modeling senses for a set of Norwegian nouns. We have described a distributional approach to meaning, where words was represented on the basis of their lexical-syntactic environment in text. An ad hoc shallow processing tool was implemented to extract the local contexts for word lemmas in morphosyntactically annotated corpora. The local context features for nouns were defined on the basis of various constructions such as verb–subject, verb–object, prepositional phrases, adjectival modification, noun–noun modification, noun–noun conjunction and possessive relations.

On the basis of the frequency counts for all the observed noun–context co-occurrences, the nouns have been given a vectorial representation in a semantic space model. The 1000 most frequent local contexts correspond to the dimensions of the space, in which 3000 nouns were furthermore positioned as points or vectors. The semantic space was defined as a triplet $\langle \mathbf{F}, A, s \rangle$, where $\mathbf{F}$ corresponds to the noun–context co-occurrence matrix, A is an association measure based on the log odds ratio, and s is a similarity function. A corresponding set of association vectors $\mathbf{X}$ were computed by weighting each component of $\mathbf{F}$ with the association measure A.

In order to partition the set of words as represented by their association vectors we developed a hybrid clustering scheme; an initial pass of bottom-up merging was followed by various fuzzy partitionings. In the first step of the analysis the words were clustered with the agglomerative within-groups average method followed by a separate pruning procedure. On the basis of the centers computed for the resulting hard clusters we then applied three different fuzzy methods – fuzzy c-means, possibilistic c-means and possibilistic prototype classification. The latter approach yielded the best results for our data set, and examples of the corresponding soft word classes were shown in section 4.3.1. As these word clusters seem to demonstrate, the notion of fuzzy sets appear to be quite suited for the task of representing semantic categories. Through the fuzzy partitioning each word was assigned varying degrees of membership across a set of fuzzy clusters. We have maintained a similarity based interpretation of the fuzzy memberships, where each graded membership is taken to indicate the degree of typicality or similarity between a given word and conceptual class. Furthermore, the spatial metaphor underlying the vector space model facilitates an intuitive approach to construing the fuzzy memberships as a function of the distance between a word and class prototype.

Clearly, other methods within the framework of fuzzy clustering might also profitably be applied. The fuzzy partitional procedures employed in this project are all quite simple, and the literature on fuzzy computing contains a well of other methods that might prove to be more fit for the task of inferring word senses directly from data. Section 4.3.1 pointed to some directions that might be interesting to pursue. In this project we have taken a rather explorative approach with respect to the appropriateness of both method and data. On one hand we wanted to look into the use of local contexts to characterize word semantics, and on the other hand we wanted to test the applicability of fuzzy methods on distributional language data.

The Middle Ground In addition to the fuzzy classes shown in section 4.3.1, section 3.3.2 gave examples of nearest neighbors for various target words, as defined by the context space $\langle \mathbf{F}, A, s \rangle$. These similarity relations seem to attest the potential of semantic knowledge inherent in the semantic space constructed on the basis of local contextual features. Moreover, these results might perhaps also be interesting with respect to arguments pro et contra syntactic bootstrapping of semantics in language learning. In this connection Li, Burgess, and Lund (2000) claim that local features would be ill-suited for indicating the semantic content of words, and argue that one should rely on aggregated representations of a broader window context instead. With respect to the construction of their HAL (Hyperspace Analogue to Language) model Lund et al. (1995) writes "*[...] we chose a window width of ten words. Our hope is that this preserves locality of reference, while obscuring the effects of different syntactic constructions. As a further move away from dependence on syntax (or any structuring of the language under consideration other than that given by the division of words) sentence boundaries are ignored.*" However, the local contexts that form the basis of the semantic space model in this thesis are given as various lexical-syntactic relations extracted for lemmatized words. This is in contrast to the broader window contexts that are traditionally employed in such vector space models. As a comment to this situation, Sahlgren (2002) remarks; "*Most of all, we need to think hard about how to incorporate more linguistic information into the vector representations. The perhaps most disqualifying feature of today's vector-space models is the blunt disregard for linguistic properties of the text data. The, at times explicit, opinion in the vector-space community is that the models should neglect linguistic structures [(Lund et al., 1995)]. We believe that this opinion is mistaken.*"

In the light of what has been said in the preceding sections, this thesis might be positioned within what Grefenstette (1992) calls *a middle ground*, where quantitative models and methods are combined with syntactic and grammatical information. The features that formed the basis of the distributional representations were extracted by means of syntactic and grammatical rules applied to annotated corpora with lemmatized word forms. This makes the overall approach linguistically informed in a way that contrasts many other unsupervised distributional methods, where the notion of co-occurrence is instead founded on context windows, or even documents, with “raw” or unprocessed text.

Nonetheless, for the task of modeling word meaning the most important direction for future work is probably towards ways of *combining* various sources for contextual information in one and the same model. This point is also stressed by Miller and Leacock (2000) and Resnik (1997) in relation to constructing distributional representations for performing word sense disambiguation. The data set used in this project consists of relatively high-frequent words. As we mentioned in section 2.1.2, however, different definitions of context may be suited for characterizing words from different frequency strata. In the setting of WSD, Leacock et al. (1993) note that representations based on local context provide for excellent precision but low recall.

In relation to the task of identifying similarity relations, Grefenstette (1993) analogously finds that while syntactic context can provide very precise sense indicators for common words with many observations, windowing techniques seem more appropriate for rare and low-frequent words for which there is less empirical evidence. In order to attain broad-coverage semantic models it might therefore be necessary to develop compound representations and models that can incorporate various types of contextual information.

Appendix A

Source File Index

	Description	Thesis Section	Source File
Shallow Processing	Spartan, extracting local context for nouns.	2	spartan.pl
	Regular expressions for tag matching etc.	2.3.2	S-Patterns.pm
	IO, corpus navigation, and auxiliaries.	2.3.2	S-Utils.pm
	Local context rules.	2.3.2	S-Rules.pm
	Mapping relations to word features.	2.3.2	rel2feat.pl
	Context windows.	2.3.3	conwin.pl
Data	List of nouns.	3.1.4	noun-rank-list
	List of context feaures.	3.1.4	context-rank-list
	Co-occurrence counts.	3.1.4	feature-counts
Semantic Spaces	Semantic space data type.	3	semantic-spaces.lisp
	Association Weighting.	3.2.	association.lisp
	Mapping words and features to integer indices.		string-index.lisp
Clustering	Agglomerative methods.	4.1.1.	bu-cluster.lisp
	Paritional methods.	4.2 and 4.3.	fuzzy.lisp
Data Types and General Operations	Matrix operations.		matrices.lisp
	Sparse matrix data types.		sparse-matrices.lisp
	Vector operations. Distance functions.	3.3	vectors.lisp
	Sparse vector data types.		sparse-vectors.lisp
	Symmetric matrix data type. Memoization.	4.1.2.	utilities.lisp

Table A.1: Source code index

Bibliography

Ball, G. and D. Hall (1967). A clustering technique for summarizing multivariate data. *Behav. Sci. 12*, 153–155.

Barni, M., V. Cappellini, and A. Mecocci (1996, August). Comments on "a possibilistic approach to clustering". *IEEE Transactions On Fuzzy Systems 4*(3), 393–396.

Bezdek, J. C. (1973). *Fuzzy Mathematics in Pattern Classification.* Ph. D. thesis, Cornell University.

Bezdek, J. C. (1980). A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*(1), 1–8.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms.* Advanced Applications in Pattern Recognition. Plenum Press.

Bezdek, J. C. (1998). *Handbook of Fuzzy Computation*, Chapter F6; Pattern Analysis. Institute of Physics Publishing.

Bezdek, J. C., R. J. Hathaway, M. J. Sabin, and W. T. Tucker (1987). Convergence theory for fuzzy c-means: Counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics SMC-17*(5), 873–877.

Bezdek, J. C. and K. P. Sankar (1992a). *Fuzzy Models for Pattern Recognition; Methods that Search for Structure in Data*, Chapter Cluster Analysis, pp. 29–34. IEEE Press.

Bezdek, J. C. and K. P. Sankar (1992b). *Fuzzy Models for Pattern Recognition; Methods that Search for Structure in Data*, Chapter Background, Significance and Key Points, pp. 1–27. IEEE Press.

Bilgiç and I. B. Türkşen (1999). Measurement of membership functions: Theoretical and experimental work. In D. Dubois and H. Prade (Eds.), *Handbook of Fuzzy Sets and Systems, Vol. 1 Fundamentals of Fuzzy Sets*, Chapter 3, pp. 195–202. Kluwer Academic Publishers.

Brown, P. F., V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer (1992). Class-based n-gram models of natural language. *Computational Linguistics 18*(4), 467–479.

Carlberger, J., H. Dalianis, M. Hassel, and O. Knutsson (2001). Improving precision in information retrieval for swedish using stemming. In *Proceedings of NODALIDA 01́ – 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden.

Carreira-Perpiñán, M. A. (1997, January). A review of dimension reduction techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield.

Church, K. W. and P. Hanks (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, Vancouver, Canada, pp. 76 – 82.

Church, K. W. and P. Hanks (1990, March). Word association norms, mutual information, and lexicography. *Computational Linguistics 16*(1), 22–29.

Cutting, D. R., D. R. Karger, J. O. Pedersen, and J. W. Tukey (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 318–329.

Dagan, I., L. Lee, and F. Pereira (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning 34*(1-3), 43–69.

Dagan, I., S. Marcus, and S. Markovitch (1995, February). Contextual word similarity and estimation from sparse data.

Deerwester, S. C., S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science 41*(6), 391–407.

Dubois, D. and H. Prade (1993). Fuzzy sets and probability: misunderstandings, bridges and gaps. In *Proceedings of the Second IEEE Conference on Fuzzy Systems*, pp. 1059–1068.

Dubois, D. and H. Prade (1998). *Handbook of Fuzzy Computation*, Chapter B4; Possibility Theory. Institute of Physics Publishing.

Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. New York, USA: John Wiley & Sons.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics 3*(3), 32–57.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 19*(1), 61–74.

Everitt, B. S., L. Sabine, and L. Morven (2001). *Cluster Analysis* (4. ed.). London: Arnold.

Evert, S. (2001, June). On lexical association measures.

Evert, S. and B. Krenn (2001). Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 188–195.

Fano, R. M. (1961, March). *Transmission Of Information: A Statistical Theory of Communication.* New York: MIT Press.

Firth, J. R. (1968). A synopsis of linguistic theory. In F. R. Palmer (Ed.), *Selected Papers of J. R. Firth: 1952–1959.* Longman.

Gärdenfors, P. (2000). *Conceptual Spaces, The Goemetry of Thought.* MIT Press.

Grefenstette, G. (1992). Sextant: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *30th Annual Meeting of the Association for Computational Linguistics*, pp. 324–326.

Grefenstette, G. (1993). Evaluation techniques for automatic semantic extraction: Comparing syntactic and window based approaches. In *ACL SIGLEX Workshop on Lexical Acquisition*, Columbus,Ohio. SIGLEX/ACL.

Hagen, K., J. B. Johannessen, and A. Nøklestad (2000). A constraint-based tagger for norwegian. In *17th Scandinavian Conference of Linguistics.*

Hall, M. and J. Mayfield (1993, September). Improving the performance of ai software: Payoffs and pitfalls in using automatic memoization. In *Proceedings of the Sixth International Symposium on Artificial Intelligence*, Monterrey, Mexico.

Harris, Z. S. (1968). *Mathematical Structures of Language.* New York: Wiley.

Hearst, M. A. (1991, October). Noun homograph disambiguation using local context in large text corpora. In *the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, Oxford.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 268–275.

Jantzen, J. (1998, October). Neurofuzzy modelling. Technical Report 98-H-874 (nfmod), Technical University of Denmark, Department of Automation.

Johannessen, J. B. (1998). En grammatisk tagger for norsk (bokmål). Technical report, Text Laboratory, University of Oslo.

Jonsdottir, A. B., E. Velldal, and R. Holberg (2002, May). Clustering of norwegian nouns. Working papers for the course Statistical Methods at the Swedish National Graduate School of Language Technology.

Karlsson, F., A. Voutilainen, J. Heikkila, and A. Anttila (Eds.) (1995). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text.* Number 4 in Natural Language Processing. Walter de Gruyter.

Klir, G. and B. Yuan (1998). *Handbook of Fuzzy Computation*, Chapter A1.1; Basic Concepts and History of Fuzzy Set Theory and Fuzzy Logic. Institute of Physics Publishing.

Krishnapuram, R. and J. M. Keller (1993, May). A possibilistic approach to clustering. *IEEE Transactions On Fuzzy Systems 1*(2), 98–110.

Krishnapuram, R. and J. M. Keller (1996, August). The possibilistic c-means algorithm: Insights and recommendations. *IEEE Transactions On Fuzzy Systems 4*(3), 385–393.

Kuncheva, L. I. and J. C. Bezdek (1997). A fuzzy generalized nearest prototype classifier. In *Proceedings of the 7th International Fuzzy Systems Association (IFSA) World Congress*, Volume 3, Prague, Czech, pp. 217–222.

Landauer, T. K. and S. T. Dumais (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* (104), 211–240.

Landes, S. and C. Leacock (1998). Building semantic concordances. In C. Fellbaum (Ed.), *WordNet, An Electronic Lexical Database*, pp. 199–216. MIT Press.

Leacock, C., G. Towell, and E. Voorhes (1993). Towards building contextual representations of word senses using statistical models. In *SIGLEX workshop: Acquisition of Lexical Knowledge from Text.* ACL.

Lee, L. and F. Pereira (1999). Distributional similarity models: Clustering vs. nearest neighbors. In *37th Annual Meeting of the Association for Computational Linguistics*, pp. 33–40.

Lenci, A. et al. (2000, Mar). Simple wp2 – linguistic specifications. Technical report, The Specification Group.

Levy, J. P. and J. A. Bullinaria (2001). Learning lexical properties from word usage patterns: Which context words should be used? In R. French and J. Sougne (Eds.), *Connectionist Models of Learning, Development and Evolution: Proceedings of the Sixth Neural Computation and Psychology Workshop*, pp. 273–282. Springer.

Li, H. and N. Abe (1998). Word clustering and disambiguation based on cooccurence data. In *COLING-ACL*, pp. 749–755.

Li, P., C. Burgess, and K. Lund (2000). The acquisition of word meaning through global lexical co-occurrences. In *Proceedings of the Thirty-first annual Child Language Research Forum*, Stanford, CA, pp. 167–178. CSLI.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *COLING-ACL*, pp. 768–774.

Lowe, W. (2001). Towards a theory of semantic space. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pp. 576–581. Lawrence Erlbaum Associates.

Lowe, W. and S. McDonald (2000, April). The direct route: Mediated priming in semantic space. Informatics Research Report EDI-INF-RR-0017, Division of Informatics, University of Edinburgh.

Lund, K. and C. Burgess (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers 28*(2), 203–208.

Lund, K., C. Burgess, and R. Atchley (1995). Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the Cognitive Science Society*, Hillsdale, New Jersey, pp. 660–665. Erlbaum Publishers.

Manning, C. D. and H. Schütze (1999). *Foundation of statistical natural language processing*. The MIT Press.

Masulli, F. and S. Rovetta (2002, April). Soft transition from probabilistic to possibilistic fuzzy clustering. Technical Report DISI-TR-03-02, University of Genova.

McDonald, S. (1997). Exploring the validity of corpus-derived measures of semantic similarity.

McDonald, S. and M. Ramscar (2001). Testing the distributional hypothesis: The influnce of context on judgements of semantic similarity. In *23rd Annual Conference of the Cognitive Science Society*, pp. 611–616.

Michie, D. (1968, April). "memo" functions and machine learning. *Nature* (218), 19–22.

Miller, G. and C. Leacock (2000). Lexical representations for sentence processing. In Y. Ravin and C. Leacock (Eds.), *Polysemy: Theoretical and Computational Approaches*, Chapter 8. Oxford University Press.

Mitchell, T. M. (1997). *Machine Learning*. Computer Science Series. McGraw-Hill.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General 115*(1), 39–57.

Pal, N. R., K. Pal, and J. C. Bezdek (1997). A mixed c-means clustering model. In *The Sixth IEEE International Conference on Fuzzy Systems*, pp. 11–21.

Pantel, P. and D. Lin (2002). Discovering word senses from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 613–619.

Patel, M., J. Bullinaria, and J. Levy (1998). Extracting semantic representations from large text corpora. In D. Glasspool and G. Houghton (Eds.), *4th Neural Computation and Psychology Workshop, London, 9-11 April 1997: Connectionist Representations*, London, pp. 199–212. Springer-Verlag.

Pedersen, T. (1996, October). Fishing for exactness. In *Proceedings of the South Central SAS User's Group (SCSUG-96) Conference*, Austin, Texas, pp. 188–200.

Pereira, F., N. Tishby, and L. Lee (1993). Distributional clustering of english words. *ACL 31*, 183–190.

Resnik, P. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph. D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Resnik, P. (1997, April). Selectional preference and sense disambiguation. Presented at the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How? Washington D.C., USA.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research 11*, 95–130.

Rosch, E. and C. B. Mervis (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology 7*, 573–605.

Ruspini, E. H. (1969). A new approach to clustering. *Inform. Control 15*, 22–32.

Ruspini, E. H. and E. Francesc (1998). *Handbook of Fuzzy Computation*, Chapter B2.3; Interpretations of Fuzzy Sets. Institute of Physics Publishing.

Sahlgren, M. (2001). Vector-based semantic analysis: Representing word meanings based on random labels. In *Proccedings of the Workshop on the Acquisition and Representation of Word Meaning at ESSLLI '01*, Helsinki, Finland.

Sahlgren, M. (2002, March 25–27). Towards a flexible model of word meaning. Paper presented at the AAAI Spring Symposium 2002, Stanford University.

Schütze, H. and C. Silverstein (1997). Projections for efficient document clustering. In *Proceedings of SIGIR.*

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science* (237), 1317–1323.

Sneath, P. H. A. and R. R. Sokal (1973). *Numerical Taxonomy.* San Francisco, CA: Freeman.

Steinbach, M., L. Ertöz, and V. Kumar (2003). The challenges of clustering high-dimensional data. In *New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition.* Springer-Verlag.

Tugwell, D. and A. Kilgarriff (2001). Word sketch: Extraction and display of significant collocatioins for lexicography. In *ACL workshop COLLOCATION: Computational Extraction, Analysis and Exploitation.*

Windham, M. P. (1982). Geometrical fuzzy clustering algorithms. In J. C. Bezdek and K. P. Sankar (Eds.), *Fuzzy Models for Pattern Recognition; Methods that Search for Structure in Data.* IEEE Press. 1982 is the year of the article, 1992 is the year of the book..

Wittgenstein, L. (1953). *Philosophical Investigations.* Oxford: Blackwell.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings, COLING-92*, Nantes, pp. 454–460.

Zadeh, L. A. (1965). Fuzzy sets. *Inform. Control 8*, 338–353.

Zadeh, L. A. (1978a). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* (1), 3–28.

Zadeh, L. A. (1978b). Pruf : A meaning representation language for natural languages. *International Journal of Man-Machine Studies* (10), 395–460.

Zimmermann, H. J. and P. Zysno (1985). Quantifying vagueness in decision models. *European Journal of Operational Research* (22), 148–158.

Zipf, G. K. (1935). *The Psycho-Biology of Language*. Boston: Houghton Mifflin.