A Graph Model for Words and their Meanings

Von der Philosophisch-Historischen Fakultät der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Philosophie (Dr. phil.) genehmigte Abhandlung

vorgelegt von

Beate Dorow

aus Albstadt

Erstgutachter: PD Dr. phil. habil. Ulrich Heid
Zweitgutachter: Prof. Hinrich Schütze, Ph.D.

Tag der mündlichen Prüfung: 03.03.2006

Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart
2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The lexicon develops with its users. New words appear and old ones adopt new meanings to accommodate new technology, recent scientific discoveries or current affairs. Some parts of the lexicon grow incredibly fast. In particular those pertaining to booming fields such as the biosciences or computer technology. The lively research activity in these areas produces a flow of publications teeming with new terminology. The sheer volume of data and information makes computer-based tools for text processing ever more essential.

Most of these systems rely heavily on knowledge about words and their meanings as contained in a lexicon. General lexicons are often not adequate for describing a given domain. They contain many rare words and word senses which are completely unrelated to the domain in question. At the same time, they fail to cover relevant words. Since it has become impracticable to maintain lexicons manually, automatic techniques for classifying words are needed.

In this thesis we take a graph-theoretic approach to the automatic acquisition of word meanings. We represent the nouns in a text in form of a semantic graph consisting of words (the nodes) and relationships between them (the links). Links in the graph are based on cooccurrence of nouns in lists. We find that valuable information about the meaning of words and their interactions can be extracted from the resulting semantic structure.

In the following we briefly describe the different semantic phenomena explored in this thesis. The structure of the thesis very much follows the structure of this introduction.

1.2 Semantic similarity and ambiguity

1.2.1 Ambiguity

Semantic ambiguity occurs when the same word form can be used to refer to multiple different concepts. For example, *rock* "music" and *rock* "stone" have the same written form, yet their meanings are completely unrelated. Language is full of such examples, which present enormous difficulties to natural language processing systems. A machine translation system, for example, must be able to choose among possible translations of an ambiguous word based on its context, such as between *Rockmusik* and *Fels* when translating the following English sentence into German:

> *Nearer the top, all the soil and loose stones had been completely eroded away, leaving a huge dome of smooth gray rock.*

The problem of labeling an ambiguous word in a context with its intended sense is referred to as word sense disambiguation (WSD), and is receiving a lot of attention from the natural language processing community. Word sense disambiguation systems draw the sense labels from an existing collection of word senses such as provided by the WordNet electronic lexicon.

The word senses listed in an existing lexicon often do not adequately reflect the sense distinctions present in the text to be disambiguated. Different fields have their own often specialized vocabularies. A general lexicon such as WordNet contains many word senses which are irrelevant to the given text. What is even more problematic is that many of the senses present in the text are not covered by the lexicon. This means that even the best WSD system cannot correctly label their occurrences in the text. Domain-specific lexicons are, however, rare and, due to the constant introduction of new words, extremely difficult to maintain.

A way to overcome the lexical bottleneck is to learn the inventory of sense labels directly from the text to be disambiguated. We approach this task from a graph-theoretic perspective. The network representation of the nouns in a text provides a very effective way of predicting the different meanings of a word automatically from the web of connections among its neighboring words.

We find that the neighborhoods of ambiguous words have a characteristic shape. They are composed of different otherwise unconnected clusters of words. As an illustration, Figure 1.1 shows the neighborhood of the ambiguous word *rock* in the word web. We distinguish two different word clusters which correspond to the “stone” and “music” sense of *rock*.

1.2.2 Semantic similarity

A common difficulty many natural language processing applications face is data sparseness. The problem of data sparseness can be overcome by making use of semantic category information. A semantic category is a collection of words which share similar semantic properties. Semantic categories include kitchen tools, academic subjects, body parts, musical instruments and sports.

The availability of semantic category information allows a text processing system to generalize observed linguistic properties of the frequent category members to all other category members, including the infrequent and unfamiliar ones. In particular, semantic category membership can be used to make predictions about the meanings of unknown words based on the other known words in a category.

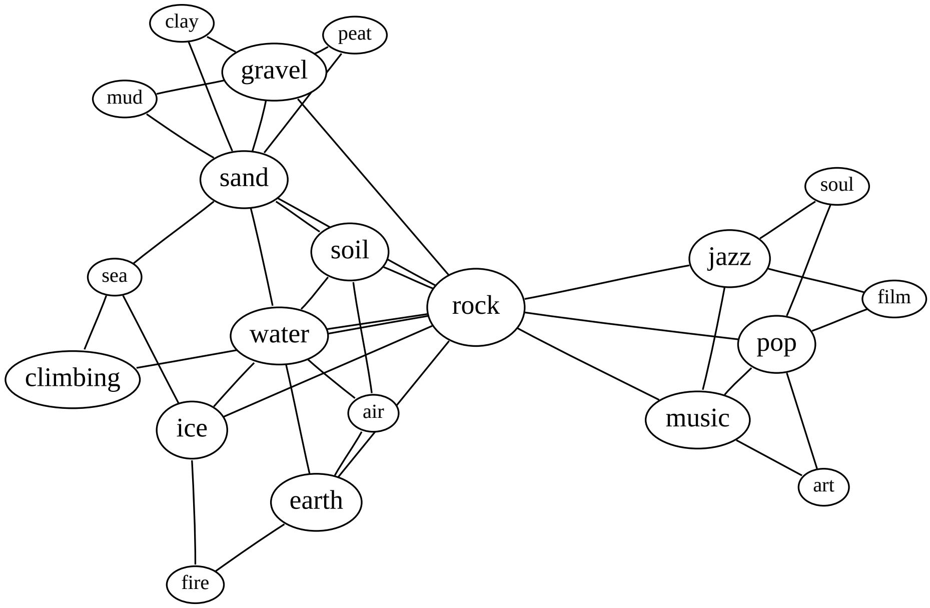


Figure 1.1: Graph snippet reflecting the ambiguity of the word *rock*. There are two different meanings of *rock* represented in the graph: *rock* "stone" and *rock* "music".

Semantic categories tend to aggregate in dense clusters in the word graph, such as the "music" and the "natural material" clusters in Figure 1.1. These node clusters are held together by ambiguous words such as *rock* which link otherwise unconnected word clusters. We attempt to divide the word graph into cohesive semantic categories by identifying and disabling these semantic "hubs".

In addition, we investigate an alternative approach which divides the *links* in the graph into clusters instead of the *nodes*. The links in the word graph contain more specific contextual information and are thus less ambiguous than the nodes which represent words in isolation. For example the link (*rock, gravel*) clearly addresses the "stone" sense of *rock*, and the link between *rock* and *jazz* unambiguously refers to *rock* "music". By dividing the links of the word graph into clusters, links pertaining to the same sense of a word can be grouped together (e.g. (*rock, gravel*) and (*rock, sand*)), and links which correspond to different senses of a word (e.g. (*rock, gravel*) and (*rock, jazz*)) can be assigned to different clusters, which means that an ambiguous word is naturally split up into its different senses.

1.3 Idioms

Idioms pose an even greater challenge to systems trying to understand human language than ambiguity. Idioms are used extensively both in spoken and written, colloquial and formal language. Any successful natural language processing system must be able to recognize and interpret idiomatic expressions.

The following sentence is an example of idiomatic word usage:

> *Your level of control needs to be high enough so that your carrot and stick power matters and is taken seriously by others.*

It is extremely difficult for non-native speakers of English, let alone for a computer, to grasp the meaning of the sentence. Neither the word *carrot* nor the word *stick* contributes to the meaning of the sentence with its literal meaning.

The word graph covers a certain type of idioms, namely those taking the form of lists. We find that words which are related on a purely idiomatic basis can be distinguished from the regularly related words in the word graph by the pattern of connections between their surrounding nodes.

We have seen that ambiguous words correspond to *nodes* in the word graph which link otherwise isolated areas of meaning. Similarly, idiomatically related words correspond to *links* which connect semantically incompatible concepts.

This effect is illustrated in Figure 1.2. The link between *carrot* and *stick* functions as a semantic bridge between the semantic categories of "vegetables" and "tools".

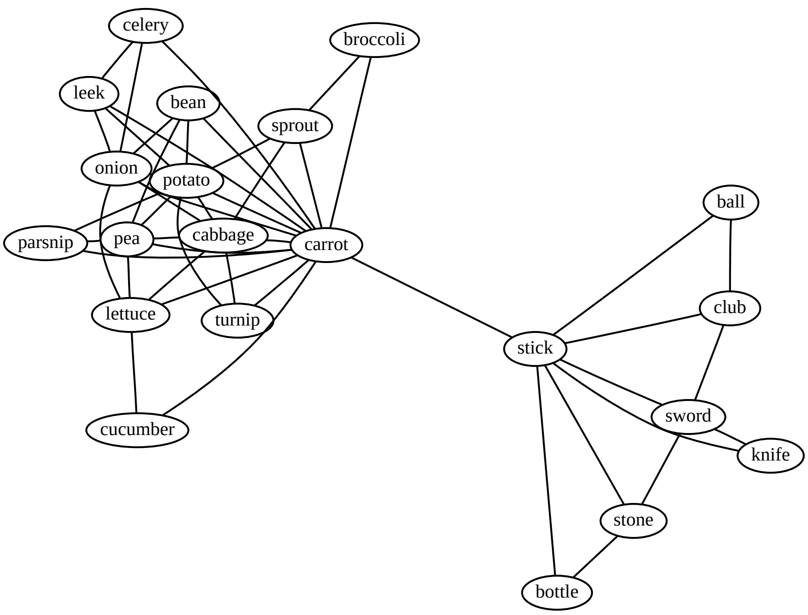


Figure 1.2: The neighborhood surrounding the idiomatically related words *carrot* and *stick*. The two nodes are linked to mutually exclusive clouds of meaning.

1.4 Synonyms

Ambiguity is the phenomenon that occurs when a single word form has several meanings. Conversely, synonymy is the phenomenon of several distinct word forms sharing the same single meaning.

Synonymy poses yet another challenge to automatic text analysis. For example, being asked

Who is the author of "Harry Potter"?,

a question answering system needs to be able to recognize that the sentence

JK Rowling is the writer of "Harry Potter".

constitutes an answer to the question raised, which means, it has to know that the words *writer* and *author* are synonyms.

Again, synonymy information is implicit in the pattern of interactions between the words in the word graph. We have seen that idiomatic word pairs correspond to pairs of nodes which are linked to one another, but have different neighbors. Synonyms show the opposite behavior: They are linked to the same neighbors, but not to one another (cf. Figure 1.3).

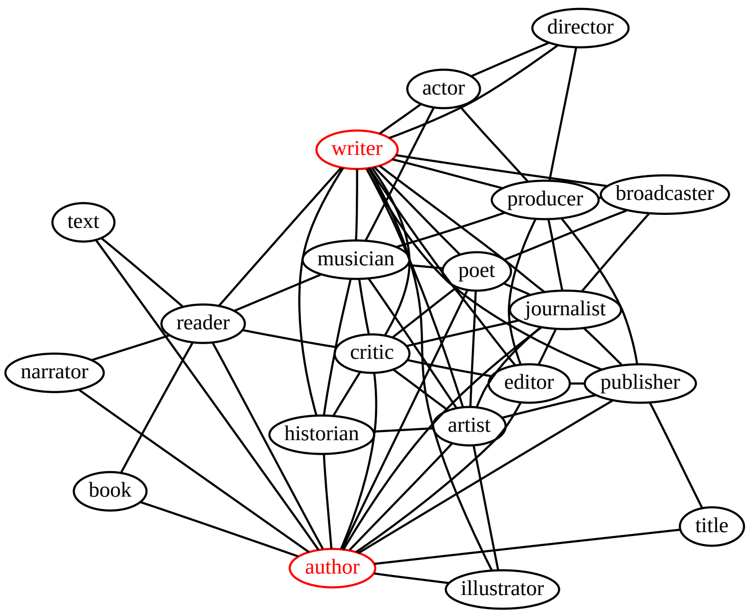


Figure 1.3: The neighbors of *writer* and *author* in the word graph. The *writer* and the *author* node share many neighbors, but are not connected to one another.

1.5 Contributions of the thesis

We show that graphs can be learned directly from free text and used for ambiguity recognition and lexical acquisition. We introduce several combinatoric methods for analyzing the link structure of graphs, which can be used to discern the meanings of previously unknown words. This ability is particularly valuable in rapidly expanding fields such as medicine and technology. Furthermore, our methods allow us to reliably acquire the different meanings of ambiguous words in a text without invoking external knowledge. Word senses can be learned automatically for any particular domain if a text representative of the domain is available.

The proposed techniques also promise to be useful for the acquisition of complex terms, such as idiomatic expressions.

1.6 Outline of the thesis

Chapter 2 introduces the basic concepts of graph theory, and provides examples of naturally occurring graphs. We focus particularly on their topological properties which are concerned with the manner in which the nodes are interconnected, and which provide important information about the behavior of individual nodes as well as the graph as a whole. We introduce and motivate the concept of *graph curvature* which plays a fundamental role in this thesis, and which measures the cohesiveness of a node's neighborhood in the graph.

Chapter 3 explains how data was collected and how we built a graph of words from corpus text. We describe the preprocessing steps which were taken to eliminate spurious links and give a brief summary of the types of mistakes made during data acquisition.

Chapter 4 describes an algorithm which automatically discovers the different senses of a word in the text. Sense clusters are iteratively computed by clustering the local graph of similar words around an ambiguous word. Discrimination against previously extracted sense clusters enables us to discover new senses. We report the results of two small experiments which tested the effectiveness of the algorithm on words of different kinds and degrees of polysemy.

Chapter 5 explores the concept of graph curvature from a semantic perspective. We show that curvature is a good predictor for the degree of polysemy of a word. In addition, curvature can be used as a clustering tool: By removing the nodes of low curvature which correspond to ambiguous words, the word graph breaks up into disconnected coherent semantic clusters. We evaluate the results of the clustering against WordNet. We find that curvature clustering is particularly suited for uncovering the meanings of unknown words.

Chapter 6 investigates an alternative approach which treats *pairs* of linked words rather than *individual* words as semantic nodes, and agglomerates these more contextual units into usage clusters corresponding closely to word senses. This technique naturally allows ambiguous words to belong to several clusters simultaneously. We present an evaluation of the results using WordNet as a gold standard and give some examples of learned word senses.

Chapter 7 analyzes the asymmetric links in the word graph which correspond to pairs of words which favor a particular word order. We find two causes for asymmetry: Semantic and idiomatic constraints. We sketch several ideas for distinguishing pairs of regularly related words from idiomatically related words.

Chapter 8 describes a method for extracting synonyms from the word graph which is based on the assumption that synonyms share many neighboring words in the word graph but are not neighbors of one another. We find that, contrary to our intuition, synonyms are linked to one another quite often. We show by evaluation against WordNet that synonym retrieval can be improved significantly by relaxing our initial assumption such as to allow links between synonyms.

Chapter 9 gives an overview of relevant related work.

Chapter 10 provides a summary of this thesis, discusses its contributions to the field of automatic lexical acquisition, and provides ideas for future research.

Chapter 2

Graphs

Graphs are a very flexible mathematical model for numerous and varied problems. Any system that can be described as a set of objects and relationships or interactions between them can be naturally represented as a graph.

Graphs arising in real life are called *networks*. Networks pervade our daily life. Road, railway and airline networks, connecting different sites, cities or airports, allow us to travel from one place to another. Phone calls and emails are transmitted over a network of cables between telephones or computers. The World Wide Web is a network of Web pages connected by hyperlinks, and more and more becomes our primary source of information. We find ourselves being part of networks of people who are connected, for example, by friendship or professional relationships. There are also numerous networks within the human body itself. They include the network of biochemical reactions between the molecules in a cell, and the brain which is a complex network of neurons and their synaptic connections. Human language, too, can be described as a network of words or concepts which are connected if one makes us think of the other.

The broad applicability of graphs has the major advantage that even very different problems can be approached using the same tools from graph theory. Furthermore, problems which can be modeled as graphs have a natural visual representation. This is a very useful aspect of graphs as visualization is often the key to understanding a set of data.

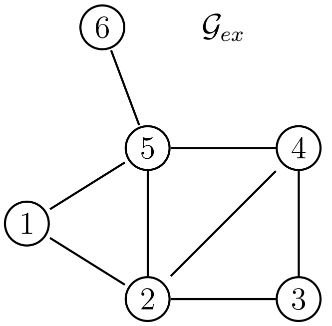
In Section 2.1, we provide the fundamental terminology and notation regarding graphs which is used throughout this book. Section 2.2 supplements the formal description of graphs with examples of natural and man-made systems taking the form of graphs, as well as some of the questions which scientists are trying to answer by studying their structure.

Section 2.3 introduces the concept of curvature (or clustering coefficient), a measure of the degree of transitivity in a graph, which is central to this thesis.

2.1 Terminology and notation

In its simplest form, a graph $\mathcal{G}$ consists of two components, a set of *vertices* V and a set of unordered pairs of vertices E which are called *edges*. In short, $\mathcal{G} = (V, E)$, where E is a subset of $V \times V$, the set of all possible combinations of two vertices. Less formally, vertices and edges may also be called *nodes* and *links*. The *order* of a graph is its number of vertices.

A graph can be visualized as a collection of points and lines joining these points. Figure 2.1 is an example of a graph of order 6. It has vertices $V = \{1, 2, 3, 4, 5, 6\}$ and edges $E = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (1, 5), (2, 5), (2, 4)\}$.



$A_{\mathcal{G}_{ex}}$

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	1	1	0
3	0	1	0	1	0	0
4	0	1	1	0	1	0
5	1	1	0	1	0	1
6	0	0	0	0	1	0

Figure 2.1: Pictorial and adjacency matrix representation of a simple graph $\mathcal{G}_{ex}$.

A graph $\mathcal{G}$ may also be represented as a matrix $A_{\mathcal{G}}$ with rows and columns labeled by the vertices of $\mathcal{G}$. The matrix element in row i and column j is 1 if there exists an edge between vertices i and j, that is, if $(i, j) \in E$, and 0 otherwise. $A_{\mathcal{G}}$ is called the *adjacency matrix* of graph $\mathcal{G}$. The adjacency matrix of the example graph $\mathcal{G}_{ex}$ is shown on the right of Figure 2.1. Note that $A_{\mathcal{G}_{ex}}$ is *symmetric*: For all vertex labels i and j, the corresponding matrix entries (i, j) and (j, i) of $A_{\mathcal{G}}$ are the same. Naturally, if vertices i and j are connected by an edge, then so are j and i.

A graph $\mathcal{G}' = (V', E')$ is called a *subgraph* of $\mathcal{G}$ if V' and E' are subsets of V and E. Figure 2.2 shows two subgraphs of our example graph.

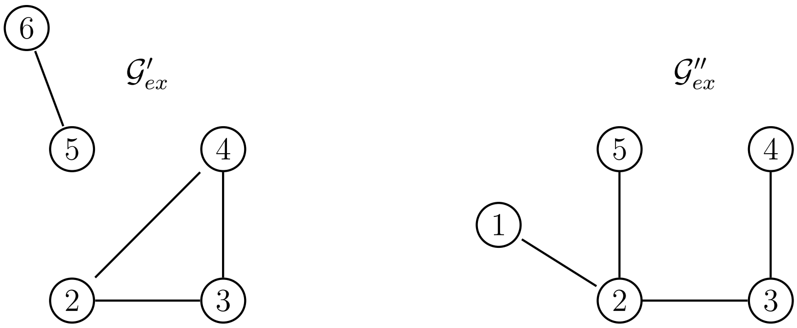
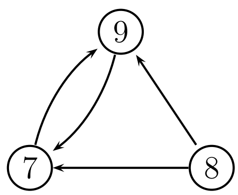


Figure 2.2: Two subgraphs $\mathcal{G}'_{ex}$ and $\mathcal{G}''_{ex}$ of the graph $\mathcal{G}_{ex}$ shown in Figure 2.1.

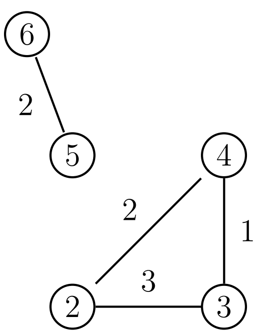
If the edges are given a direction, the graph is called *directed*. In mathematical terms, the set of edges of a directed graph consists of *ordered* pairs of vertices which can be referred to as *arcs*. The adjacency matrix of a directed graph no longer needs to be symmetric. There may be an arc from vertex j to vertex i, but not from i to j. An example of a directed graph and its adjacency matrix is given in Figure 2.3. An entry (i, j) of the matrix is positive if there is an arc from node j to node i.



	7	8	9
7	0	1	1
8	0	0	0
9	1	1	0

Figure 2.3: A directed graph and its adjacency matrix.

Both directed and undirected graphs may be *weighted*. A weighted graph has a number associated with each edge which can be thought of as reflecting the strength of the connection. The graph shown in Figure 2.4 is a weighted version of graph $\mathcal{G}'_{ex}$ shown in Figure 2.2. The entries of the adjacency matrix (shown on the right side) now represent edge weights.



	2	3	4	5	6
2	0	3	2	0	0
3	3	0	1	0	0
4	2	1	0	0	0
5	0	0	0	0	2
6	0	0	0	2	0

Figure 2.4: A weighted graph and its adjacency matrix.

A term which the reader will encounter many times throughout this dissertation is that of a node's *degree*. The degree of a vertex is the number of edges that touch it. For example, vertex 4 in Figure 2.4 participates in two edges, namely $(4, 2)$ and $(4, 3)$, and hence has a degree of 2.

Directed graphs distinguish between a node's *indegree* and *outdegree*. The indegree is the number of edges arriving at a vertex. Similarly, a node's outdegree is given by the number of edges departing from it. Just as for undirected graphs, the degree of a node is the number of edges attached to it, which is equal to the sum of in- and outdegree. Node 7 in Figure 2.3, for example, has indegree 2, outdegree 1 and degree 3.

A graph is called *complete*, or a *clique*, if there is an edge between any two of its vertices. The complete graphs of order one to five are given in Figure 2.5. In a complete graph of order k, each vertex is connected to all of the other $k-1$ vertices. In other words, each of the vertices has maximal degree $k-1$.

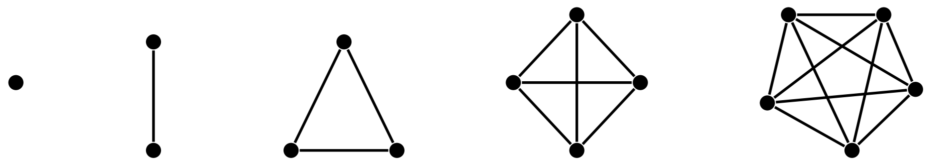


Figure 2.5: The complete graphs of order one to five.

Two vertices are *neighbors* in a graph if they are connected by an edge. The *neighborhood* of a vertex v is the set of all its neighbors, and is denoted by $\mathcal{N}(v)$. Note, that the number of a node's neighbors coincides with its degree. For example, node 2 of graph $\mathcal{G}_{ex}$ in Figure 2.1 has neighbors $\mathcal{N}(2) = \{1, 3, 4, 5\}$.

Two edges are called *adjacent* if they share a node, and a sequence of adjacent edges in the graph is called a *path*. The length of a path is the number of edges it is composed of. For example, a path of length 4 connecting nodes 1 and 6 in $\mathcal{G}_{ex}$ (Fig. 2.1) is given by $\{(1,2),(2,4),(4,5),(5,6)\}$. In the case of a weighted graph, we can speak of the *weight of a path* as the sum of the weights of the traversed edges, e.g., the path $\{(2,3),(3,4)\}$ in the graph shown in Figure 2.4 has length 2 and weight 4.

A path connecting two vertices v and w is called a shortest path if it has the shortest length among all paths connecting v and w. If edges are weighted, the term shortest path is sometimes used to refer to the path of lowest weight rather than shortest length. The shortest path between vertices may not be unique. For example, there are two shortest paths connecting vertices 3 and 5 in $\mathcal{G}_{ex}$: $\{(3,2),(2,5)\}$ and $\{(3,4),(4,5)\}$.

The notion of shortest path naturally gives rise to a measure of distance between the vertices of a graph. The distance between two vertices is simply defined as the length of their shortest connecting path, i.e. the smallest number of steps that it takes to get from one to the other.

A path is called a *cycle* if it ends at the same vertex it started from, e.g., edges $(1,2),(2,4),(4,5),(5,1)$ in $\mathcal{G}_{ex}$ form a cycle of length 4. Similarly, edges $(7,9),(9,7)$ in the directed graph shown in Figure 2.3 form a cycle of length 2. An *acyclic* graph (also called a *tree*) is a graph without cycles. The graph $\mathcal{G}''_{ex}$ is an example of a tree. A directed acyclic graph (see Fig. 2.6 for an example) is called a hierarchy.

A graph is *connected* if there exists a path from any vertex to any other vertex. Otherwise it is *disconnected*. Graphs $\mathcal{G}'_{ex}$ and $\mathcal{G}''_{ex}$ in Figure 2.2 are examples of a disconnected and a connected graph. The connected pieces of a disconnected graph are called its *connected*

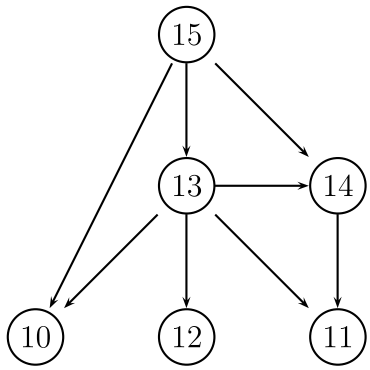


Figure 2.6: An example of a hierarchy, i.e. a directed acyclic graph.

components. Graph $\mathcal{G}'_{ex}$ consists of two connected components, one containing vertices 5 and 6 and the other one containing vertices 2, 3 and 4.

The *characteristic path length* of a connected graph is the average distance between any two of its nodes. The notion of characteristic path length can be generalized to unconnected graphs by considering only nodes in the largest connected component.

In this section, we have introduced the fundamental concepts of graph theory necessary to follow the remainder of the book. For a more thorough overview, we refer the reader to Chartrand (1985).

2.2 Real-world graphs

Graphs arise in a large number of fields including sociology, biology, computer science and engineering, to name but a few. Graphs also play an important role in linguistics. Graphs are very useful for representing semantic knowledge (Fellbaum, 1998). Moreover, modeling human language as a graph of interacting words (Dorogovtsev and Mendes, 2001; Ferrer-i-Cancho and Solé, 2001) has shed light on the very nature of language, in particular, on how words may be organized in the brain and on how language evolves.

2.2.1 Social networks

We are all involved in various social networks, either as individuals or as part of a group. A social network can be seen as a graph depicting the social relationships which hold the members of a community together. The nodes of the graph are people or groups of people, and edges are the social ties connecting, for example, friends, relatives, collaborators or business partners.

The main goal of social network analysis is to gain insight into how certain communities operate. Issues which are commonly addressed include the identification of influential members of a community, or the detection of "interest groups" within a community (Girvan and Newman, 2002; Eckmann et al., 2004).

However, social scientists are not only interested in the static properties of social networks but also in their dynamics. For example, they try to understand the mechanisms by which social networks evolve, and how the structure of the network affects the communication between the members of a community. The latter issue has important applications in epidemiology. The structure of the connections between the members of a community determines how fast, for example, an infectious disease spreads across the network, and the challenge is to find ways of altering the pattern of interactions between the members of a group at risk such as to prevent an epidemic from spreading (Newman, 2002).

Two properties which the vast majority of social networks share are the *small world* pattern (Watts and Strogatz, 1998) and the *scale-free* structure (Barabási and Albert, 1999). A "small world" is one in which two strangers are connected by only a few intermediate acquaintances. In a scale-free network, few individuals are involved in an exceedingly large number of social relationships, whereas the majority of people have only a small number of social contacts. It is these two network properties which cause ideas, rumors and diseases to spread so quickly and thoroughly.

The small world and scale-free properties are not unique to social networks. Many other networks, technological as well as natural, show surprising similarity to social networks in that they exhibit these same properties which have important implications for their functioning.

2.2.2 Technological networks

Man-made networks include traffic and communication networks, electrical power grids and electronic circuits. The two technological networks which are attracting a lot of interest these days are the Internet and the World Wide Web. The Internet is a network consisting of millions of interconnected computers, and the World Wide Web can be treated as a huge directed graph whose nodes correspond to Web pages and whose arcs correspond to the hyperlinks pointing from one page to another.

Both these networks grow extremely fast. New computers constantly enter the Internet, and new physical connections between computers are established. Similarly, every day numerous Web pages and new hyperlinks between Web pages make their way into the Web.

The Internet and the World Wide Web were the first networks which have been found to be scale-free. Barabási and Albert (1999) have discovered that the scale-free property of these networks is the result of a certain mechanism of growth which they call *preferential attachment* and which stipulates that new nodes are likely to link to already highly connected nodes. This is a surprising result since there are no rules establishing how computers have to be connected with each other, and Web pages can link to any other Web page they want.

Apart from investigating the evolution of the Web, researchers have been able to detect communities of Web pages centered around the same topic (Eckmann and Moses, 2002) by looking only at the pattern of hyperlinks. And sophisticated Web search tools have been developed which make use of the Web's link structure to assess the relevance of a Web page (Brin and Page, 1998).

Analysis of the structure of the Internet has raised awareness of its vulnerability against attacks by hackers and has provided practical information on how to effectively protect it (Albert et al., 2000).

2.2.3 Biological networks

The range of physiological processes in a cell which keep an organism alive and growing give rise to many networks in which vertices represent genes, proteins or other molecules, and edges indicate the presence of a biochemical or physical interaction between the corresponding chemicals. By analyzing the pattern of molecular interactions, researchers are trying to gain insight into the functional organization of the cell (Jeong et al., 2001). Graph-theoretic analysis of cellular networks allows us to identify the molecules most essential to a cell's survival and to isolate groups of functionally related molecules (Eckmann and Moses, 2002).

Barabási and Oltvai (2004) could trace the amazing robustness of cellular networks against accidental failures of individual components to their scale-free architecture. Furthermore, they found interesting links between the connectivity structure of cellular networks and their evolutionary history.

Another important class of biological networks are neural networks, such as the human brain. A neural network is a collection of nerve cells (neurons) which are interconnected through synapses of different strength. The stronger a synapse between two neurons, the more likely a signal will be transmitted from one neuron to the next. In mathematical terms, a neural network is a weighted directed graph. Neural networks are in constant development with links being reorganized and changing in strength. The main goal of neural network analysis is to understand how information is represented and processed within the brain and the nervous system.

2.2.4 Graphs arising in linguistics

Many different aspects of language have a natural representation as graphs. For example, linguists use parse trees resulting from the syntactic analysis of sentences to depict how the words in a sentence structurally relate to one another (cf. Figure 2.7). The links in a parse tree represent syntagmatic relationships.

Graphs can also be used to describe how words relate to one another semantically.

WordNet: An example of a semantic graph is WordNet[1]. WordNet is an electronic lexicon in which words are organized by meaning (Fellbaum, 1998). It consists of four separate databases, one for each of the four major word classes (nouns, verbs, adjectives and adverbs).

[1] `http://wordnet.princeton.edu/`

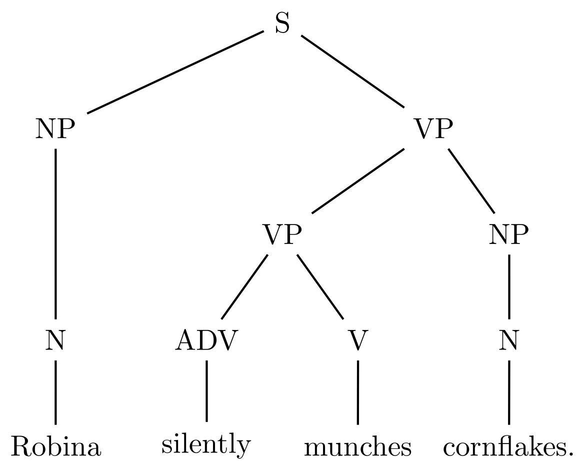


Figure 2.7: Example of a syntactic parse tree.

Within each word class, words are grouped into sets of (near-)synonyms, so-called synsets. The words in a synset collaboratively describe a single lexical concept. An example of a noun synset and its gloss is

> **{piano, pianoforte, forte-piano}**: *A stringed instrument that is played by depressing keys that cause hammers to strike tuned strings and produce sounds.*

Synsets are linked to one another by semantic relationships. These relationships (or any subset thereof), induce a graph with vertices corresponding to synsets, and edges indicating the presence of a semantic relationship between the linked synsets. Symmetric (bi-directional) relationships give rise to undirected, non-symmetric relationships give rise to directed edges.

The *hyponymy* relationship forms the backbone of the WordNet noun graph. A word u is a *hyponym* of another word w if u is a kind of w (in symbols, $u \sqsubset w$). Conversely, in this case, w is said to be a *hypernym* of u. For example, *piano* is a hyponym of *musical instrument* (*piano* $\sqsubset$ *musical instrument*), and *flower* is a hypernym of *poppy* (*poppy* $\sqsubset$ *flower*).

The hyponymy relationship is *antisymmetric*, that is if u is a hyponym of w, then W can't be a hyponym of u:

$$u \sqsubset w \quad \Rightarrow \quad w \not\sqsubset u. \tag{2.1}$$

Edges representing the hyponymy relationship are thus directed. Further, $\sqsubset$ is *transitive*. This means if u is a hyponym of v which in turn is a hyponym of w, then u is a hyponym of w:

$$u \sqsubset v \text{ and } v \sqsubset w \quad \Rightarrow \quad u \sqsubset w. \tag{2.2}$$

Properties 2.1 and 2.2 together imply that the hyponymy-induced subgraph of the WordNet noun graph is acyclic [2]. Being both directed and acyclic, the hyponymy graph is a hierarchy.

Other, less central semantic relations encoded in WordNet are *antonymy*, the symmetric relationship connecting synsets with opposite meanings (e.g., *war* and *peace* are mutual antonyms), and meronymy (holonymy), which is the antisymmetric relationship linking a part (whole) to its whole (part) (e.g., *pedal* is a meronym of *bicycle*, and *chair* is a holonym of *leg*).

Polysemy (a word having multiple meanings) is implicitly represented in WordNet by a polysemous word belonging to several different synsets. Figure 2.8 shows a fragment of the hyponymy-induced WordNet noun hierarchy. The directed edges connect a synset with its hypernymic synset. The word *chocolate* is a member of three different synsets which correspond to its different meanings and which are thus implicitly related by polysemy (red links).

As this thesis is concerned with nouns and their meanings, we will not describe the organization of the other word classes (verbs, adjectives, adverbs) here, but instead refer the reader to Fellbaum (1998).

2.3 Curvature as a measure of local graph topology

The pattern of connections between the vertices of a graph is what is known as its *topology*. It is independent of the nature of the nodes and links, and is concerned only with how the nodes are linked to one another. The topology of a network yields useful information about its behavior as well as about the behavior of individual nodes, and may even provide insight into the network's evolutionary history.

A quantity which locally characterizes the topology of a graph is the *clustering coefficient*, which was first introduced in (Watts and Strogatz, 1998). It is a property of the nodes in the graph which measures the interconnectedness of a node's neighbors. Eckmann and Moses (2002) show that the clustering coefficient has a geometric interpretation as the curvature of a graph at a given node, and thus suggest the alternative term *curvature*. In this thesis, we will use both terms exchangeably.

2.3.1 Clustering in social networks

The concept of curvature is best explained and understood by looking at social networks. Figure 2.9 shows the social network structure for the set of characters in Victor Hugo's "Les Misérables" [3]. A link between two nodes corresponds to a social tie between the associated characters. Locally, we observe very different link structures in the graph. At one extreme are fully-connected regions in the graph, such as the subgraph induced by the "Joly"-node and its neighbors in the network. Very often, ones friends are friends, too. Joly's circle of

[2] If there was a cycle $w_1 \sqsubset w_2 \sqsubset \cdots \sqsubset w_n \sqsubset w_1$ in the hyponymy graph, then Property 2.2 would imply that $w_1 \sqsubset w_n$. But at the same time $w_n \sqsubset w_1$, which contradicts Property 2.1.

[3] Borrowed from http://www-personal.umich.edu/∼mejn/networks/

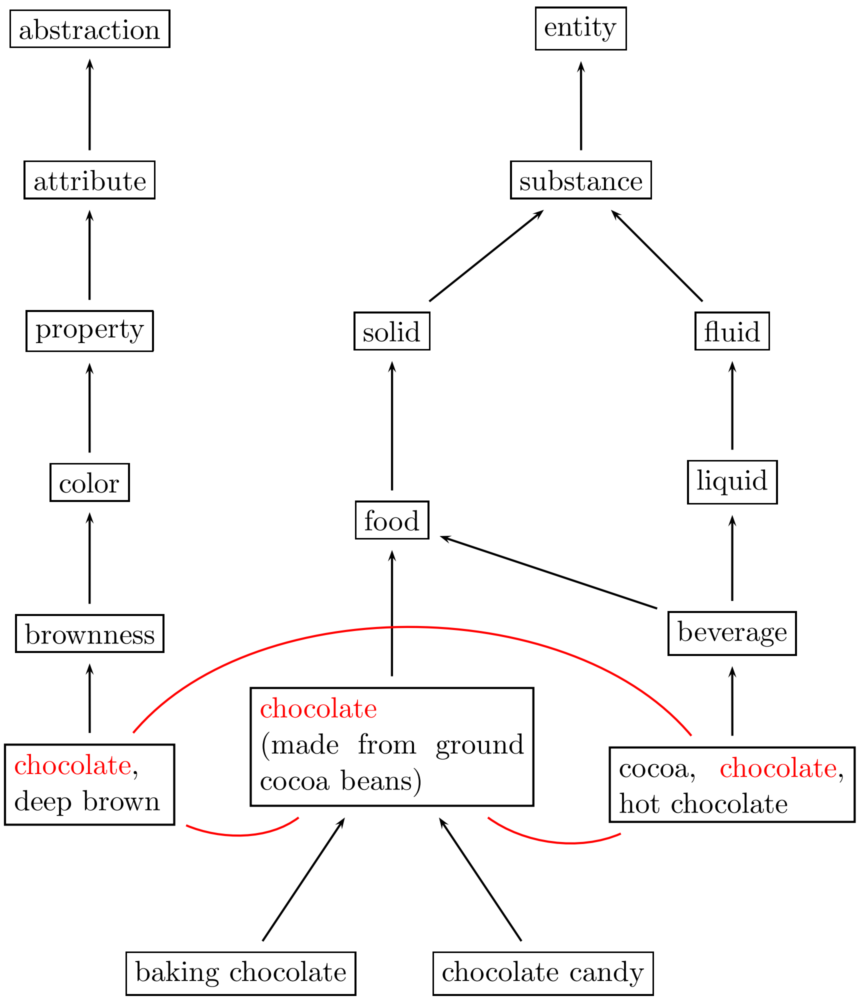


Figure 2.8: A fragment of the WordNet noun taxonomy.

friends carries this to an extreme. Any two of its members are linked. It forms a clique. In fact, Joly is a member of the "Société de la ABC", an association of revolutionary students which is tightly held together by strong companionship [4]:

> *... In addition to their revolutionary role, however, the Société also played an important role in its members' social lives. The members were best friends; they laughed and philosophized, ate and drank, and even shared lodgings with one another. They spoke about world politics and their mistresses in the same breath; even when they disagreed they wanted what was best for each other. This to Hugo was the ideal lifestyle of the student: carefree yet serious and surrounded with like minds...*

At the other extreme are subgraphs in which the only connections are those between a central node and its neighbors. The neighbors themselves are not interconnected. Topologically speaking, such a graph is a star. The set of the relationships in which the character "Myriel" is involved nearly exhibits such a *star topology*. Myriel's social surroundings are very loose: There are hardly any relationships among his acquaintances. It is he who holds his social neighborhood together. This is in accord with Myriel's nature. He is a very charitable bishop. His house is open to anyone who is in need, to people of any background, be it women, children, the poor or even aristocrats. Because of the diversity of the people Myriel cares for, most of them are mutual strangers.

In between, we observe neighborhoods which are neither fully-connected nor star-like, but feature a hybrid topology. Consider the example of the protagonist "Valjean". The neighborhood of the "Valjean"-node is a star of dense (almost fully-interlinked) subgraphs (consisting of equally colored nodes), an indication of Valjean being connected to different tight-knit communities. He is the "social hub" of the network of characters at which the different social groups meet.

Just as Figure 2.9 shows the overall social network structure in "Les Misérables", local neighborhoods provide insight into a single character's immediate social surroundings. The curvature measure attempts to quantify the nature of these local neighborhoods. Naturally, a single number can't fully describe the complex structure of a node's neighborhood. Curvature rather measures a node's degree of embeddedness in its immediate neighborhood.

2.3.2 Graph-theoretic definition

Let n be a node in a graph $\mathcal{G}$, and let $N(n)$ denote the set of n's neighbors. There are at most $\#N(n) * (\#N(n) - 1)/2$ links among the nodes in $N(n)$. The curvature (resp. clustering coefficient) of the node n is defined as

$$\begin{aligned}\mathrm{curv(n)} &= \frac{\#(\text{links among } n\text{'s neighbors})}{\#N(n) * (\#N(n) - 1)/2} \\ &=: \frac{k_{\mathrm{linked}}}{k_{\max}}.\end{aligned}$$

[4] Quoting from a plot summary at http://www.mtholyoke.edu/courses/rschwart/hist255/bohem/tmizbook.html

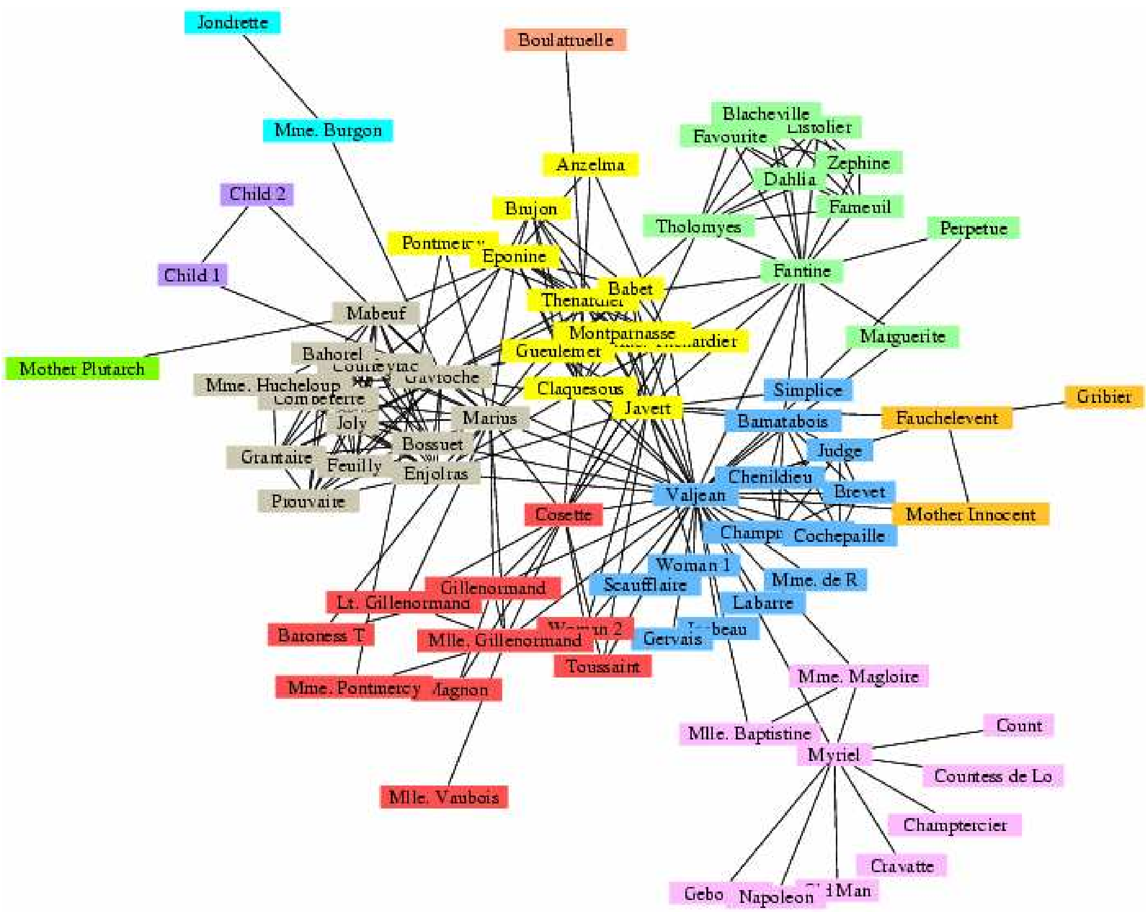
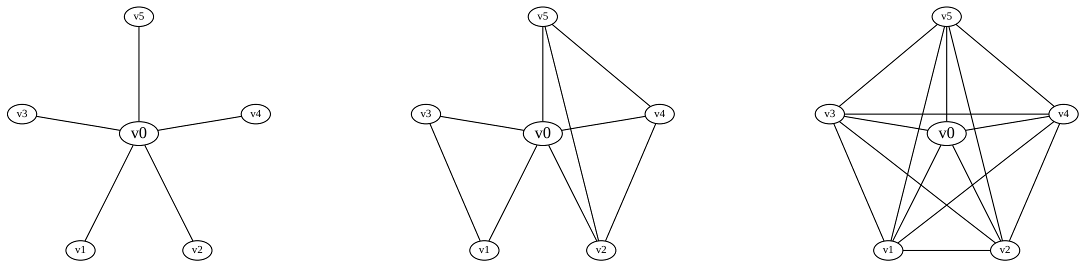


Figure 2.9: Social network in Victor Hugo's "Les Misérables".

It is the fraction of actual links among a node's neighbors out of the maximally possible number of such links. It assumes values between 0 and 1. A value of 0 occurs if there is no link between any of n's neighbors (i.e., the neighbors are maximally disconnected), and a node has a curvature of 1 if all its neighbors are linked (i.e., its neighborhood is maximally connected). Figure 2.10 shows nodes of low, medium and high curvature respectively. The



(a) $\mathrm{curv}(v_0) = 0$. (b) $\mathrm{curv}(v_0) = 0.4$. (c) $\mathrm{curv}(v_0) = 1$.

Figure 2.10: Nodes of low, medium and high curvature.

three graphs depicted in Figure 2.10 reflect the different topologies we observed in the social network of the previous section: “Myriel”, “Valjean” and “Joly” are nodes of, respectively, low, medium and high curvature. Nodes of high curvature are well-integrated in a tight node community in which almost any two nodes are interconnected. Such tightly-knit communities exhibit a high degree of transitivity: if two nodes a and b are linked to a third node c, most likely, a and b are linked, too, and the three nodes form a triangle. Therefore, another equivalent way of expressing the curvature value is

$$\mathrm{curv}(n) \quad = \quad \frac{\#(\text{triangles } n \text{ participates in})}{\#(\text{triangles } n \text{ could possibly participate in})}.$$

2.3.3 Geometric interpretation

Curvature, as Eckmann and Moses (2002) show, has a geometric interpretation which is related to the *Gaussian curvature* of a *surface*. If we consider the three-dimensional (Euclidean) space, then a surface is a 2-dimensional object which, locally (in a, no matter how small, neighborhood of every point), looks like a plane. Examples of surfaces are depicted in Figure 2.11.

The Gaussian curvature measures the ”curvedness” of a surface. For each of the surfaces in Figure 2.11 imagine a closed curve on the surface surrounding the red point. The curvature of the surface at the red point is the angle by which a narrow strip surrounding the curve opens up when cut open and spread out in the plane. Figure 2.12 shows such strips for each of the surfaces shown in Figure 2.11.

The two ends of the strip enclose an angle. We can picture each end of the strip as the position of the hand of a clock. Since the convention is to measure angles counter-clockwise, our clock runs backwards. For a parabolic surface, like the one on the left in Figure 2.11, to get from the clock time of one end of the strip to the clock time of the other by moving on the strip (that is, in the direction indicated by the arrow), the hand has to do less than a turn. In the case of a hyperbolic surface, like the one on the right, however, the hand of the clock will even pass its original position as we walk from one end of the strip to the other. This can also be interpreted as moving the hand of the clock less than a full turn *against* the direction of our funny clock. Another way to put this is, is to say that in the first case the two ends enclose a positive angle, i.e., the surface has positive curvature, whereas in the second case the enclosed angle is negative or the surface has negative curvature. The curvature of a plane is zero as the two ends are parallel.

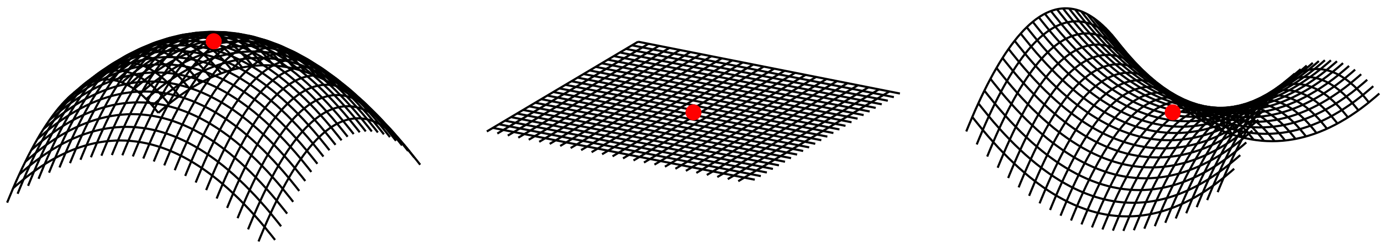


Figure 2.11: Three surfaces in three-dimensional space. The curvature at the red point is positive, zero and negative, respectively.

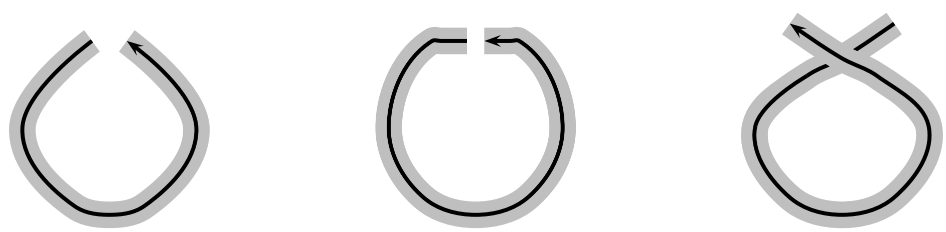


Figure 2.12: Surface strips cut along a closed curve surrounding a point of positive, zero and negative curvature, respectively.

The geometry of non-planar surfaces is very different from the Euclidean geometry in the plane. In fact, the more curved a surface, the more its geometric properties differ from the properties of the plane. For example, in Euclidean geometry, we have the law of cosines which tells us how to compute the length c of one side of the triangle, given the lengths a and b of the other two sides and the angle γ they enclose (cf. Fig. 2.3.3):

$$c^2 = a^2 + b^2 - a * b * \cos(\gamma) \tag{2.3}$$

In the case of an isosceles triangle, that is, $a = b$ and $\gamma = \pi/3$, the formula becomes

$$\begin{aligned} c^2 &= 2a^2 - 2 * a^2 cos(\pi/3) \\ &= 2a^2 * (1 - 1/2) \\ &= a^2 \\ \iff \qquad c &= a \end{aligned}$$

and the triangle is equilateral, which doesn't come as a surprise. However, this is not the case on curved surfaces.

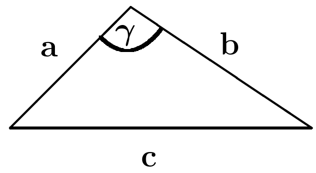


Figure 2.13: A planar triangle.

Figure 2.15 shows a triangle living on a hyperbolic surface (i.e. surrounding a point of negative curvature), using the so-called Poincaré Disk model. Figure 2.16 illustrates how points in the Poincaré Disk map to points on a hyperbolic surface. As a point gets closer to the boundary of the Poincaré Disk, its distance to the center of the Poincaré Disk approaches infinity. Therefore, two points which are a fix distance apart, are represented using a shorter line segment if they are close to the boundary than if they are close to the center. The triangle in Figure 2.15 was drawn using NonEuclid [5], an interactive tool for constructions in hyperbolic geometry. As you can see from the measurements, the sides (A, B) and (B, C) of the triangle have the same size, namely 2, and enclose an angle of 60 degrees. On the plane, as a consequence of the law of cosines (Equation 2.3), the triangle (A, B, C) would be equilateral. But, as Figure 2.15 makes evident, on a hyperbolic surface, the third side of the triangle is *longer* than the other two.

On a parabolic surface, we observe the opposite effect. Figure 2.14 shows an isosceles triangle on the sphere, drawn using Spherical Easel drawing [6], a tool for drawing in spherical geometry. Again, the two equally long sides enclose an angle of 60 degrees. This time, the third side of the triangle is *shorter* than the other two.

Intuitively, this makes sense. On a sphere, the shortest line connecting two points is along a great circle (a circle whose center is the center of the sphere), i.e., the sides of a triangle lie on such great circles. Any two great circles meet at two points on opposite sides of the sphere. Therefore, as we extend two sides of the triangle, their (non-common)

[5]http://www.cs.unm.edu/ joel/NonEuclid/NonEuclid.html
[6]http://merganser.math.gvsu.edu/easel/

endpoints converge closer and closer together, whereas in the plane, they diverge. On a hyperbolic surface, two nonparallel lines diverge even more than two nonparallel lines in a plane, since the surface stretches into *all* directions of the surrounding space[7].

If we look at an isosceles triangle two sides of which have fixed equal lengths $a = b$ and enclose a fixed angle γ, then the length c of the third side will tell us whether the surface is negatively or positively curved. The more extreme the difference in side length to what we would expect on the plane (as predicted by Equation 2.3), the more strongly the surface is curved. This can be understood by shrinking the size of the triangle. The smaller a triangle on a surface, the closer it is to being planar, and the more and more Euclidean geometry (the law of cosines amongst others) becomes adequate. Shrinking the size of the triangle has the same effect as making the surface less curved (flatter). Note, that we are able to compute a surface's curvature on the surface itself without looking at it from the 3-dimensional space in which it is contained. Mathematicians make this clear by calling Gaussian curvature an *intrinsic* property of a surface. For example, even without satellite photos showing that the Earth is a sphere, we can conclude that the Earth is positively curved by measuring big triangles on the Earth's surface. Similarly, physicists are trying to deduce the shape of the Universe from (large-scale) local measurements.

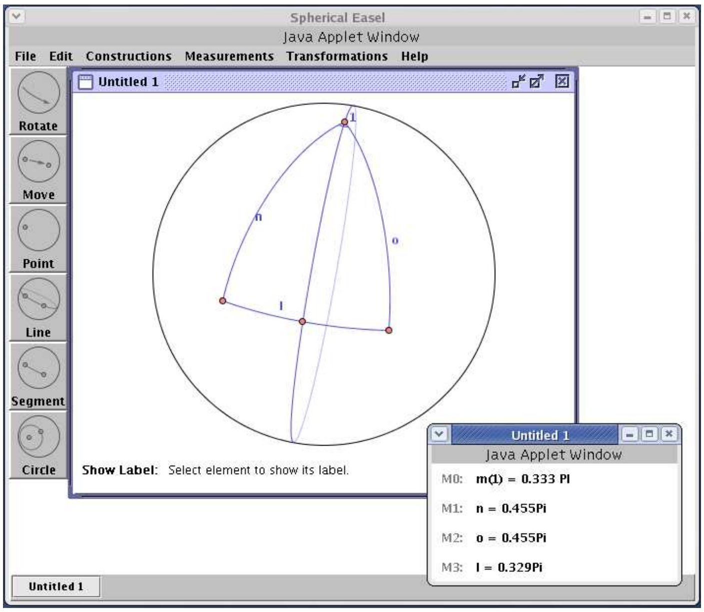


Figure 2.14: An isosceles triangle on a sphere.

To see the connection between the Gaussian curvature of a surface at a given point and the curvature of a node in the graph, let us have a look at triangles of vertices in a graph. Let us consider the subgraph induced by a node n and its set of neighbors $N(n)$. Any two neighbors a and b in $N(n)$ which are not directly linked are linked via n. Any two nodes

[7]In fact, on a hyperbolic surface, even parallel lines diverge.

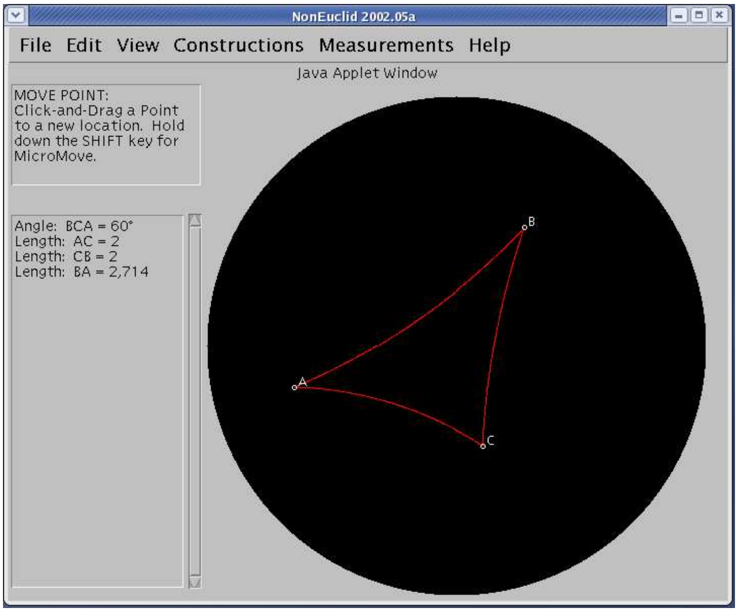


Figure 2.15: An isosceles triangle on a hyperbolic surface.

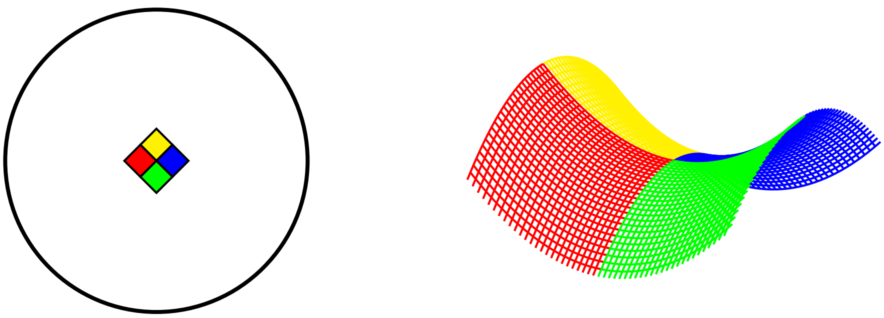


Figure 2.16: The Poincaré disk model of hyperbolic geometry.

in $N(n)$ are thus either one or two links away from one another. The average length of a path connecting two neighbors a and b is

$$\begin{aligned}
d_{avg} &= \frac{k_{\text{linked}} * 1 + (k_{\max} - k_{\text{linked}}) * 2}{k_{\max}} \\
&= 2 - \frac{k_{\text{linked}}}{k_{\max}} \\
&= 2 - \text{curv}(n) \\
\Leftrightarrow \qquad \text{curv}(n) &= 2 - d_{avg}.
\end{aligned}$$

A triangle with apex n has thus two sides of unit length (the two links from n to the two neighbors a and b) and a third side which has an average length of $d_{avg} = 2 - \text{curv}(n)$. The larger (smaller) the curvature of a node, the shorter (longer) the average distance is between any two of its neighbors, and hence the bigger the curvature is of a surface on which an equiangular triangle with side lengths 1, 1 and $2 - \text{curv}(n)$ can be drawn.

Chapter 3

Data acquisition

In this thesis, we study a network of nouns built from free text. The vertices are nouns, and edges connect nouns if they interact with each other. We define word interaction as cooccurrence in lists. Our main aim is to predict the meanings of a word from the pattern of interactions with the other words in the word web. In the following we explain how the word graph is assembled.

3.1 Acquisition assumption

Several people, inspired by Hearst's work (Hearst, 1992) on the acquisition of hyponymy relationships, have successfully used lexicosyntactic patterns to learn semantic relations from text (Kietz et al., 2000; Morin and Jacquemin, 2004; Cederberg and Widdows, 2003). The idea is that certain patterns provide evidence of a specific semantic relationship between the constituent words. For example, the pattern *y such as* x_1*(,* x_2*, ... and/or* x_n*)*, as in

> *Start with helmets: They are important for <u>sports such as football, hockey, baseball, softball, biking, skateboarding, in-line skating, skiing, and snowboarding</u> - to name just a few.*

indicates that all of the x_i are kinds of y. Section 9.1 provides a summary of previous work on the extraction of semantic relationships using lexicosyntactic patterns.

We take a similar pattern-based approach to find instances of semantic similarity between nouns. Our basic hypothesis is that conjunctions (*books and catalogs*) and, more generally, lists (*art historians, philosophers, aestheticians, anthropologists, historians or biographers*), consist of words which are related semantically. More specifically, the members of a list often belong together in a particular semantic category such as musical instruments (*trumpet, violin, flute, flugel horn and recorder*), academic subjects (*optics, physics, mechanics, astronomy, mathematics and philosophy*), diseases (*TB, measles, polio, whooping cough, tetanus or diphtheria*) or countries (*Britain, India, Israel, and the USA*).

3.2 Text material

As a text source, we use the British National Corpus (BNC)[1], a 100 million word collection of contemporary British English. The corpus covers a wide range of domains (such as fiction, science, journalism), and includes texts of very different genres, ranging from transcribed phone conversations to user manuals and recipes, novels, newspaper articles and scientific papers, giving a well-balanced account of the modern English language. The entire BNC has been marked up with part of speech (PoS) tags which makes it relatively easy to extract specific grammatical patterns with the help of regular expressions.

3.3 Building the graph

Lists of words have previously been used to find categories of similar objects (Roark and Charniak, 1998; Riloff and Shepherd, 1997). They occur frequently and are relatively easy to recognize. In the following, we will describe how associations between words as conveyed by lists can be used to build a graph representation of the nouns in a text.

The structures we are interested in are lists of noun phrases which we capture by the following regular expression:

$$\text{NP(, NP)*,?(CJC NP)+} \tag{3.1}$$

$$\textit{where}$$

$$\begin{array}{lcl} \text{CJC (conjunction)} & = & (\textit{and} \mid \textit{or} \mid \textit{nor}) \quad , \\ & \text{and} & \\ \text{NP (noun phrase)} & = & \text{AT?(CRD)*(ADJ)*(NOUN)+.} \end{array}$$

The PoS tags provided by the BNC (see examples below) are used to determine the grammatical class of a word.

Pattern matching on the word/PoS level is a very shallow approach to extracting semantic relationships. However, pattern-based approaches have little requirements for preprocessing of the text, which makes them robust and widely applicable. In particular, they do not rely on a priori lexical knowledge.

Our approach to extracting relationships between words is recall-oriented. To make up for the sacrifice of precision, we apply different filters to the extracted relationships to enhance the quality of the results (see Section 3.4).

The following are examples of words and word combinations matched by the constituents of Pattern 3.1:

AT (determiner):	*the, a, no, all, this, these, any, my, her*
CRD (cardinal/ordinal):	*two, 3, first, next, last, 23rd, million*
ADJ (adjective):	*public, better, nicest, obliged, increasing, other*

[1] `http://www.natcorp.ox.ac.uk/`

NOUN (noun):	*summer, daisies, housework, September, Mozart*
NP (noun phrase):	*late successful career, the four London areas, her own business, the Belgrave Beauty Clinic, Professor Jonathan Mann*

Pattern 3.1 as a whole reveals, for example, the following lists:

- *the bones, ligaments, tendons and muscles*
- *detainee, lawyer or human right activist*
- *pencil, crayon or burnt wood or cork*
- *seventeen crime stories, two novels, two plays and a book*
- *protestant loyalism and catholic nationalism*
- *shoes and gloves and belts*

In order to extract noun-noun relationships, we extract the head noun (the underlined words in the example coordinations above) of each of the coordinated noun phrases. For simplicity, we assume that the head of a noun phrase is the last noun in the construction and that the preceding words function merely as modifiers (e.g., a *human right activist* is a special kind of *activist*). Furthermore, we assume that

1. Any two of the coordinated nouns are equally strongly related.
2. Lists are symmetric in their constituents, i.e., the meaning of a list is independent of its order.

Having made these simplifying assumptions, we can identify a list of nouns $l = x_1, x_2, \dots x_k$ with the set $E_l = \bigcup_{x_i, x_j \in l}(x_i, x_j)$ of all possible pairs of co-members. Pairs (x_i, x_j) with $x_i = x_j$ are disregarded. If we think of the nouns as nodes in a graph, then the collection $E = \bigcup_l E_l$ defines a set of undirected edges between nodes. In particular, the members of a list of length k form a k-clique. For example, a coordination with three nouns A, B and C results in three edges (A, B), (A, C) and (B, C).

In summary, our method for constructing a conceptual network of nouns consists of the following steps:

1. Extract lists of noun phrases via pattern matching.
2. Identify the head noun of each noun phrase.
3. Introduce a node for each noun.
4. Draw an edge between any two nouns if they occur together in a list.

Figure 3.1 illustrates by means of a short piece of example text [2] how the word graph is assembled. As more and more lists are processed, new nodes and edges join the graph. Figure 3.2 shows a fragment of the word graph after the entire text has been analyzed.

[2] The sentences are taken from the BNC.

- *Legend has it that the mandarin was so grateful to Earl Grey for services rendered that he gave him his secret tea recipe, to keep mind, body and spirit together in perfect harmony.*
- *So far as ordinary citizens and non-governmental bodies are concerned, the background principle of English law is that a person or body may do anything which the law does not prohibit.*
- *Christopher was also bitten on the head, neck and body before his pet collie Waldo dashed to the rescue.*

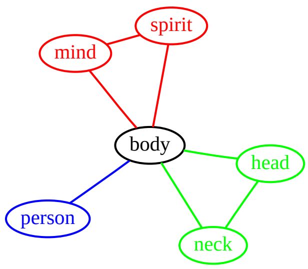


Figure 3.1: Illustration of the graph building process. Two nouns are connected by a link if they cooccur in a list.

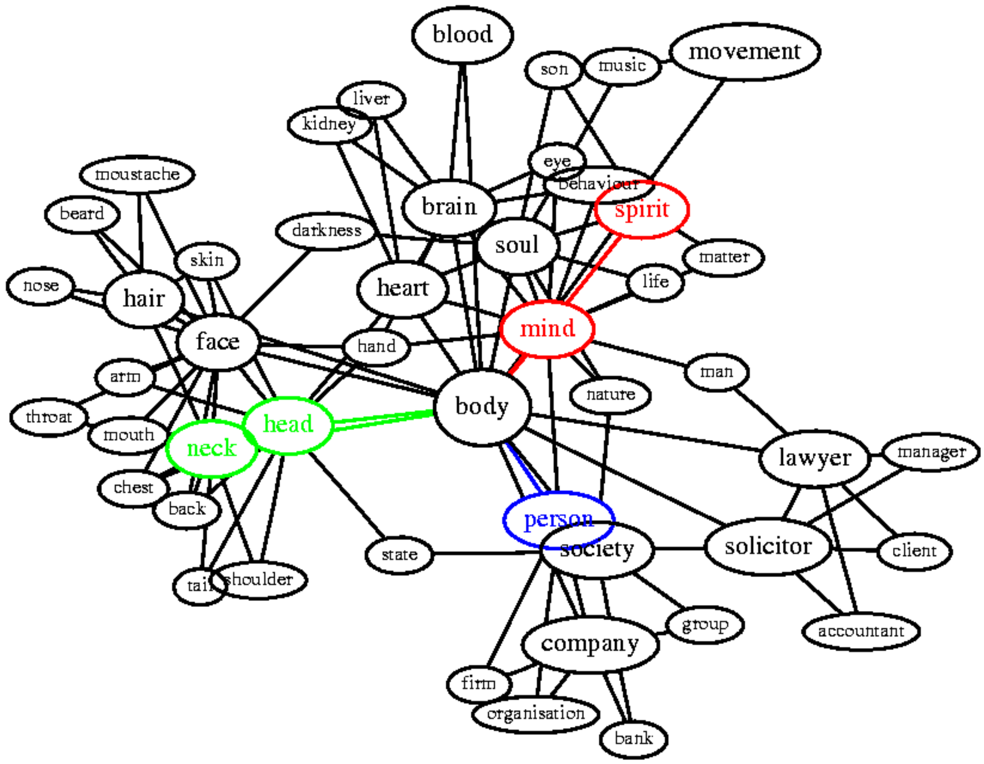


Figure 3.2: The neighborhood of body in the final word graph.

3.4 Preprocessing of the word graph

3.4.1 Conflating variants of the same word

Most words appear in various surface forms throughout the corpus, in particular in their singular and plural forms. Although they are only morphological variations of the same word, each of the surface forms has its own representation as a node in the word graph. To reduce redundant nodes and links in the graph, we fuse the nodes corresponding to a word's different morphological forms into a single node. The resulting node inherits all the edges of the nodes being fused. This process is illustrated using a hypothetical graph in Figure 3.3.

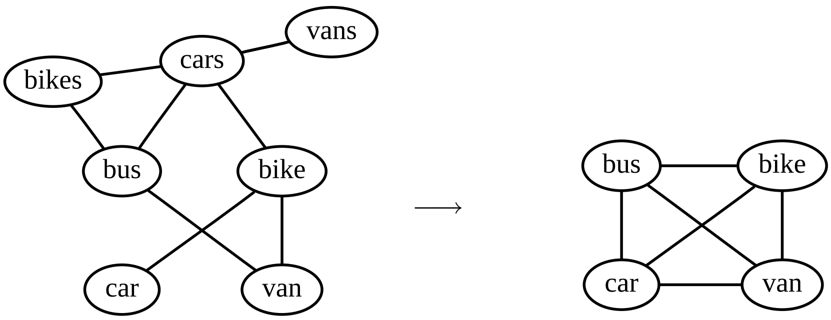


Figure 3.3: Nodes corresponding to nouns sharing the same base form are conflated to a single node.

We use WordNet to identify morphological variations of the same underlying base form. WordNet provides a word's different base forms. We replace a word with its base form if the base form is unique. For example, WordNet lists only a single base form for the noun *cars* which becomes *car*. Some words, however, have more than one base form because they are ambiguous across their morphological guises. The noun *aids*, for example, can be either the plural form of *aid* or the singular form of *Aids*. Consequently, it is left unchanged.

This means that words, such as *cars* and *car*, which share the same unique base form are considered to be surface representations of each other and are collapsed to a *single* node in the graph. Words, such as *aids* and *aid*, which, dependent on the context, may or may not be morphological variations of the same base form, are represented by *different* nodes in the graph.

Identifying the different morphological forms of a word with each other does not only reduce redundancy but also data sparseness. By generalizing across word forms, unobserved links, such as those between *van* and *car*, *car* and *bus*, and *bus* and *bike* in the above example graph can be inferred. Furthermore, the reduced graph has a higher level of clustering (the nodes have higher curvature, cf. Section 2.3), i.e. exhibits a higher degree of community structure than the original graph.

3.4.2 Eliminating weak links

One way of eliminating weak links without having to set up a threshold, is to discard all edges which do not occur in a triangle in the graph. Intuitively, an edge connecting two nodes in the word graph suggests that the corresponding words are semantically similar. This conjecture is substantiated further if, in addition, the two words share a semantic neighbor. In general, semantic similarity is not transitive: Two words which are similar to the same third word are not necessarily similar themselves. For example, the words *tennis* and *zucchini* are very different in meaning although they are both semantically similar to readings of *squash*. If we do observe transitivity among three nodes, this can be taken as strong evidence that the edges by which the nodes are connected correspond to true word associations. Extremely weak or mistaken associations due to errors in collecting or processing the data, are unlikely to occur in a triangle and are hence eliminated. Figure 3.4 illustrates the triangle filtering process.

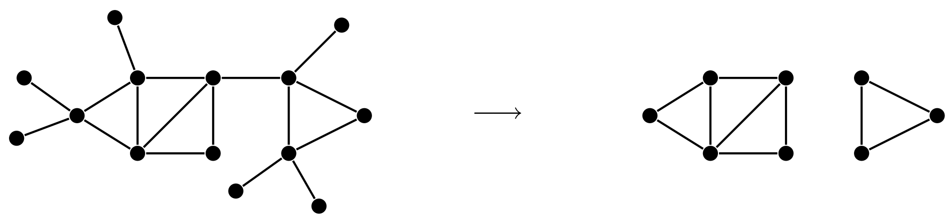


Figure 3.4: We discard edges which do not occur in a triangle.

3.5 Weighting links

The word graph can be extended by assigning weights to its edges to reflect the strength of the observed word associations. The simplest option for weighting edges is to use raw frequency counts. However, very frequent words naturally have a bigger chance of occurring together than rare words. Several statistical measures for assessing the degree of association between words have been proposed and studied (for a comprehensive overview see (Evert, 2004)).

We decided to use the *log-likelihood* score, which was first proposed by Dunning (1993) as a measure of association between words. The log-likelihood measure provides sensible estimates of association strength even for relatively rare words (Dunning, 1993).

Suppose we are interested in measuring the strength of association between two words v and w. All of the observed cooccurrences of words in lists can be partitioned into the following four categories which contain those word pairs which involve

- both v and w
- v but not w

- w but not v
- neither v nor w

We denote the number of word pairs falling into each of these categories with O_{ij} where the index i (j) indicates the presence (1) or absence (0) of v (w). If the words v and w cooccurred only by chance, we would expect to see the following category frequencies:

$$E_{ij} = \frac{O_{i\cdot}\, O_{\cdot j}}{N} \quad , \quad \text{where} \tag{3.2}$$

$$O_{i\cdot} = O_{i0} + O_{i1} \;\; , \quad O_{\cdot j} = O_{0j} + O_{1j} \;\; , \quad N = \sum_{i,j} O_{ij}.$$

The log-likelihood score measures the discrepancy between the observed category counts O_{ij} and the expected counts E_{ij}, and is given by the following term (Evert, 2004):

$$2 \sum_{(i,j)\in\{0,1\}^2} O_{ij} \;\log \frac{O_{ij}}{E_{ij}}. \tag{3.3}$$

The log-likelihood measure has also an information-theoretic interpretation. Let us introduce two random variables I_v and I_w on the sample space consisting of all cooccurrences of words, which are defined such that I_v (I_w) assumes a value of 1 or 0 depending on whether v (w) forms part of an observed pair of cooccurring words. We use $P(I_v = i)$, $P(I_w = j)$ and $P(I_v = i, I_w = j)$ to denote the empirical probabilities of the corresponding events. The log-likelihood measure can then be written in the following alternative form:

$$2\,N \sum_{(i,j)\in\{0,1\}^2} P(I_v = i, I_w = j) \;\log \frac{P(I_v = i, I_w = j)}{P(I_v = i)P(I_w = j)}. \tag{3.4}$$

This is, disregarding the multiplicative factor of $2N$, equal to the mutual information between the random variables I_v and I_w. Mutual information measures the “distance” between the joint probability distribution and the product of the individual probability distribution of two random variables. If I_v and I_w are statistically independent, then, by definition, the joint and the product probability distribution of I_v and I_w are identical and, as a result, the mutual information between I_v and I_w is 0. The mutual information between two random variables is therefore a measure of their dependency.

This means the log-likelihood score of v and w (which as we have seen is equal to the mutual information between I_v and I_w) measures the degree to which the presence or absence of word v influences the presence or absence of word w and vice versa.

In fact, most of the lexical acquisition methods described in the following chapters do not make use of edge weights, but rely solely on connectivity information.

3.6 Spurious edges

3.6.1 Errors in data acquisition

A number of wrongly detected lists are the result of tagging errors. Other types of problems occurring during both the collection and interpretation of lists are described in the following sections.

Wrongly detected lists: Our shallow approach to extracting lists leads to a number of spurious matches. For example, a problem arises when a list is preceded by a prepositional or adverbial phrase, as in the following example sentences:

- *For art historians, incomplete schemes or dismembered works such as altarpieces have the attraction of needing detective work.*
- *In compensation, picture or sculpture may be accompanied by comment from the artist.*

The noun phrase in the introductory phrase (e.g., *art historian* resp. *compensation*), and the comma which separates the introductory phrase from the main clause, are wrongly interpreted as forming part of the subsequent list. Pattern 3.1 cannot distinguish the above constructions from true lists such as the following ones:

- *For every city, town and village there are insider's reviews of accommodation, restaurants and nightlife.*
- *In France, Prussia and Spain, the aristocracy did not even have to pay taxes.*

Another source of mismatches are coordinations involving complex noun phrases. Consider the following example sentences.

- *His teaching was at variance with that of the Archbishop of Canterbury and other bishops.*
- *Like the flashing of fireflies and ticking of pendulum clocks, ...*

Both sentences contain a coordination of noun phrases, the first of which consists of a head noun followed by a prepositional phrase. Pattern 3.1 doesn't account for complex noun phrases and hence ignores the head (*archbishop* resp. *flashing*) and considers only the object of the preposition (*canterbury* resp. *fireflies*) to form part of the list.

Defining a more general pattern wouldn't improve this situation as it is impossible on the token/part-of-speech level to distinguish the above cases of coordinated complex noun phrases from cases such as the following ones, where a coordination forms the object of a prepositional phrase.

- *The Department of Physics and Astronomy of The University of Montana-Missoula.*
- *...for the identification and characterization of proteins and peptides from mass spectrometry data.*

Similar mismatches occur when Pattern 3.1 encounters a coordination of sentences, such as

- *It was a warm, sunny day and my foot was starting to ache.*

Our pattern cannot determine that the coordination is between sentences rather than nouns.

A large number of spurious associations arising from wrongly detected lists are eliminated by the triangle filter. A deeper syntactic analysis, e.g. using a chunker, may further improve the reliability of the extracted lists but will likely harm recall.

Another source of false positives are appositive constructions, such as

> *Mariam Firouz, a writer and translator in her mid-70s, was arrested for her political activities on April 6, 1983.*

Although appositions do not constitute lists in the strict sense, their constituents are semantically closely related and give rise to extremely useful links in the word graph. In fact, Riloff and Shepherd (1997) and Roark and Charniak (1998) have explicitly used appositives (amongst other constructions) for collecting groups of semantically similar words.

Wrongly interpreted lists: Some spurious links occur as a result of misinterpretation of the extracted lists, and are mostly due to our rather simplistic approach to bracketing coordinated compound nouns. In many cases, in particular for elliptic constructions (*music and theater critic*, *summer and autumn day*, *rose and herb bed*), the correct bracketing is of the form "A and B C" rather than the assumed "A and B C". A simple approach to avoiding such incorrect bracketings could be to look for occurrences of "A and B" and "A and C" in the corpus whose presence or absence confirms or disproves a potential bracketing.

Moreover, whereas stripping off prenominal modifiers often leads to relationships at a reasonable level of specificity, this approach is not adequate when dealing with compound nouns. For example, consider the following lists which give rise to links connecting the underlined nouns.

- *library and post office*
- *Paris and New York*
- *the Queen and the Prime Minister*
- *the black death and the plague*
- *more red tape and more regulation*

Naively splitting up a noun compound into a head and its modifiers often produces incorrect or at least extremely arguable relationships. Compound extraction techniques could alleviate this problem (Baldwin et al., 2003).

Many of the spurious links introduced while collecting and interpreting lists of words disappear with the triangle filter described earlier.

3.6.2 Violation of the symmetry assumption

Contrary to our intuition, the words in a list often cannot be arranged freely. Various, mostly semantic, rules make us prefer a particular ordering over the other possible ones. For example, lists are often ordered by the value, influence or size of their constituents (e.g., *food and clothing*; *landlord and tenant*; *elephant and mouse*), or according to temporal, logical or causal constraints (e.g., *spring, summer and autumn*; *shampoo and conditioner*; *initiation and enactment*). Assuming symmetry between the words in a list means ignoring word order preferences, which are adequately accounted for only if edges are given a direction. Asymmetric links are examined more closely in Chapter 7.

3.6.3 Semantically heterogeneous lists

Other lists are even more rigid in terms of word order. These include proper names, such as book or movie titles, which often take the form of lists (e.g., *Romeo and Juliet*; *The lion, the witch and the wardrobe*), but also lists which through consistent usage have made their way into the lexicon (e.g., *fish and chips*; *bed and breakfast*). In addition to violating the symmetry assumption, many of these lists do not even meet our basic assumption of being composed of semantically similar words. This is particularly the case for idiomatic expressions, such as *carrot and stick* or *rank and file*, which are often semantically extremely heterogeneous.

Rather than considering these lists to contribute only noise, we take on the challenge and use the structural properties of the word graph to work out which links arise from idiomatic usage rather than semantic similarity. This work is described in Chapter 7.

3.6.4 Ambiguity due to conflation of variants

Very often, number is a good disambiguating clue, and conflating singular and plural may cause (previously absent) ambiguity where it could have been avoided. For example, the noun *squash* in its plural form *squashes* unambiguously addresses the "vegetable" and not the "racquet sport" sense of *squash*. Similarly, the noun *apples* most likely refers to a collection of fruits rather than a set of computers. However, as we will see in Chapter 5, conflating different words or different senses of a word leaves its mark on the connectivity pattern of the graph so that we are able to restore the different meanings.

3.6.5 Statistical properties of the word graph

Table 3.1 summarizes the basic statistical properties (the number of nodes, links, triangles and connected components, as well as the size of the largest component) of the word graph at different stages of preprocessing. Conflating the different variants of a word, reduces the number of nodes and links, but, at the same time, causes a considerable increase in triangles, that is in clustering. Before application of the triangle filter, the word graph contains a large number of isolated links. The number of connected components thus drops considerably if only links in triangles are considered.

	#nodes	**#links**	**#triangles**	**#comps**	**largest comp**
raw coocs	112,809	933,551	7,813,865	3,407	105,220
+ **variant confl**	99,755	861,985	10,968,040	3,177	92,692
+ **tri filter**	48,346	746,229	10,968,040	600	46,320
+ **WN filter**	21,082	617,947	10,667,181	21	21,019

Table 3.1: Some simple statistical properties of the word graph at different stages of preprocessing.

In the subsequent chapters, we will distinguish between two graphs, the fully-preprocessed word graph $\mathcal{W}$ (third row in Table 3.1), and the subgraph $\mathcal{W}_{\mathrm{wn}}$ which is induced by the subset of nouns which are known to WordNet (last row in Table 3.1). We remove dangling links in $\mathcal{W}_{\mathrm{wn}}$ by another application of the triangle filter.

Chapter 4

Discovering word senses

4.1 Intuition

Polysemous words link otherwise unrelated areas of meaning. For example, the words *rat* and *monitor* mean very different things, yet they both have an interpretation which is closely related to one of the meanings of the polysemous word *mouse*.

Consider Figure 4.1 which shows part of the word graph surrounding the word form *mouse*. While there are many links within the same area of meaning, nodes representing meanings from different areas are linked to one another only via the *mouse*-node. Upon removing the *mouse*-node and all its incident edges, the graph shown in Figure 4.1 decomposes into two parts, one representing the "electronic device" meaning of *mouse* and another one which represents its animal sense.

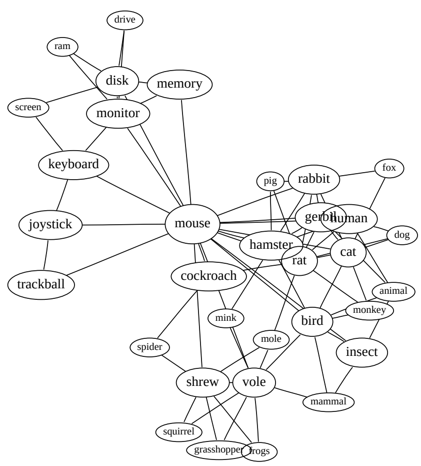


Figure 4.1: The word graph in the vicinity of the word form *mouse*.

The kind of variation in word meaning present in the *mouse*-example is just one of several types of ambiguity (Kilgarriff, 1992). As can be seen in Figure 4.2, *wing* "part of a

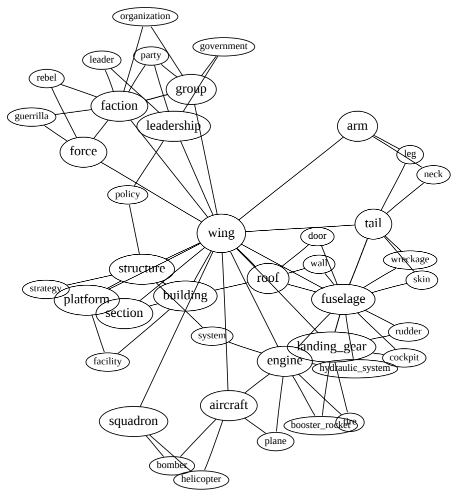


Figure 4.2: Neighborhood graph of the word form *wing*.

bird" is closely related to *tail*, as is *wing* "part of a plane". Therefore, even after removal of the *wing*-node, the two areas of meaning are still linked via *tail*. The same happens with *wing* "part of a building" and *wing* "political group" which are linked via *policy*. However, while there are many edges *within* an area of meaning, there is only a small number of (weak) links *between* different areas of meaning. To detect the different areas of meaning in our local graphs, we use a cluster algorithm for graphs (Markov clustering, MCL) developed by van Dongen (2000).

4.2 Markov Clustering

The idea underlying the MCL algorithm is that random walks within the graph will tend to stay in the same cluster rather than jump between clusters.

The following notation and description of the MCL algorithm borrows heavily from van Dongen (2000). Consider a weighted graph $\mathcal{G}$ with adjacency matrix $A_{\mathcal{G}}$ (cf. Section 2.1) whose entries are the link weights. Normalizing the columns of $A_{\mathcal{G}}$ results in the Markov Matrix $T_{\mathcal{G}}$ whose entries $(T_{\mathcal{G}})_{uv}$ can be interpreted as transition probabilities from node u to node v. It can easily be shown that the k-th power of $T_{\mathcal{G}}$ lists the probabilities $(T_{\mathcal{G}}^k)_{uv}$ of reaching node v from node u in k steps.

The MCL algorithm simulates flow in $\mathcal{G}$ by iteratively recomputing the set of transition probabilities via two steps, expansion and inflation. The expansion step corresponds to taking the k-th power of $T_{\mathcal{G}}$ as outlined above and allows nodes to see new neighbors. The inflation step takes each individual matrix entry to the r-th power and then rescales each column so that the entries sum to 1.Via inflation, popular neighbors are further supported at the expense of less popular ones.

Flow within dense regions in the graph is concentrated by both expansion and inflation.

Eventually, flow between dense regions will disappear, the matrix of transition probabilities $T_{\mathcal{G}}$ will converge and the limiting matrix can be interpreted as a clustering of the graph.

4.3 Word Sense Clustering Algorithm

The output of the MCL algorithm strongly depends on the inflation and expansion parameters r and k as well as the size of the local graph which serves as input to MCL.

An appropriate choice of the inflation parameter r depends on the ambiguous word w to be clustered. In case of homonymy, a small inflation parameter r is appropriate. However, there are ambiguous words with more closely related senses which are metaphorical or metonymic variations of one another. In that case, different regions of meaning are more strongly interlinked and a small power coefficient r would lump different meanings together.

Usually, one sense of an ambiguous word w is much more frequent than its other senses present in the corpus. If the local graph handed over to the MCL process is small, we might miss some of w's meanings in the corpus. On the other hand, if the local graph is too big, we will get a lot of noise.

Below, we outline an algorithm which circumvents the problem of choosing the right parameters. In contrast to pure Markov clustering, we don't try to find a complete clustering of the local graph $\mathcal{G}_w$, which surrounds the ambiguous word w, into senses at once. Instead, in each step of the iterative process, we try to find the most distinctive cluster c of $\mathcal{G}_w$ (i.e. the most distinctive meaning of w) only. We then recompute the local graph $\mathcal{G}_w$ by discriminating against the cluster c. This means, in the updated graph $\mathcal{G}_w$, we consider only words which are neither contained in, nor linked to any of the words (nodes) in c. The process is stopped if the similarity between w and its best neighbor in the updated graph is below a fixed threshold.

This is in a way similar to Pantel and Lin (2002)'s sense clustering method, who, once a word has been assigned to a cluster, remove from it those features which overlap with the features of the cluster. This approach allows to reveal even the less prominent senses of a word.

Consider an ambiguous word w. Let L be the output of the algorithm, i.e. a list of sense clusters, initially empty. The algorithm consists of the following steps:

1. Compute a small local graph $\mathcal{G}_w$ around w considering only words which are neither contained in, nor linked to any of the clusters c in L. If the similarity between w and its closest neighbor is below a fixed threshold go to 5.
2. Recursively remove all nodes of degree one. Then remove the node corresponding to w from $\mathcal{G}_w$.
3. Apply MCL to $\mathcal{G}_w$ with a fairly big inflation parameter r which is fixed.
4. Take the the most significant cluster (the one that is most strongly connected to w in $\mathcal{G}_w$ before removal of w), add it to the final list of clusters L and go back to 1.
5. Go through the final list of clusters L and assign a name to each cluster using a broad-coverage taxonomy (see below).

As the initial local graph $\mathcal{G}_w$ in Step 1, we choose the graph which consists of w, the n_1 top neighbors of w and the n_2 top neighbors of the neighbors of w. In the experiment described below, neighbors have been ranked by frequency of occurrence, but may alternatively be ranked by a different measure of association as, for example, the log-likelihood measure introduced in Section 3.5.

In each iteration, we attempt to find only one significant cluster. Therefore, it suffices to build a relatively small graph in Step 1. Step 2 removes noisy strings of nodes pointing away from $\mathcal{G}_w$. The removal of w from $\mathcal{G}_w$ might already separate the different areas of meaning, but will at least significantly loosen the ties between them.

In our simple model based on noun cooccurrences in lists, Step 1 corresponds to rebuilding the graph under the restriction that the nodes in the new graph must not cooccur (or at least not very often) with any of the cluster members already extracted.

We used the class labeling algorithm of Widdows (2003) in Step 5 to assign a name to the extracted clusters. This algorithm exploits the taxonomic structure of WordNet. More specifically, the hypernym which subsumes as many cluster members as possible and does so as closely as possible in the taxonomic tree is chosen as class label.

4.4 Experimental Results

As an initial evaluation experiment, we gathered a list of nouns with varying degree of ambiguity, from homonymy (e.g. *jersey*) to systematic polysemy (e.g. *cherry*), and applied our algorithm to each word in the list (with parameters $n_1 = 20$, $n_2 = 10$, $r = 2.0$, $k = 2.0$) and extracted the top two sense clusters. We then used the class labeling algorithm to determine the WordNet synsets which most adequately characterize the sense clusters. An extract of the results is listed in Table 4.1. Automatically generated sense clusters for several other words are similarly encouraging.

We expect that the word sense discovery algorithm can be considerably improved if words are linked based on shared grammatical relations (e.g. verb-object, verb-subject and adjective-noun relationships) instead of mere cooccurrence in lists. In particular, words with systematically related senses could benefit greatly from the use of more distinctive grammatical features. For example, the word graph has great difficulty distinguishing between *BMW* the car and *BMW* the company, because all the neighbors of *BMW* share the same sense ambiguity. Syntactic information can help disambiguate these two senses: *BMW* in its company sense occurs, for example, as a subject of *employ* and *decide*, just as the words *company* or *corporation* do. Instead, when it refers to a car, *BMW* tends to occur as an object of *drive*, or to be modified by *new* or *expensive*, just as other objects do.

In a first attempt to improve the recognition of systematically related senses, we used verb-object, verb-subject and adjective-noun relationships extracted from German newspaper text using the Lexicalized Probabilistic Context-Free Grammar developed by Schulte im Walde (2003), to characterize the nouns in the corpus. We then build a graph of nouns by linking any two of them based on feature overlap. The degree to which two words share features is measured using Lin (1998)'s information-theoretic word similarity measure.

Word	**Sense clusters**	**Class label**
jersey	israel colombo guernsey luxembourg denmark malta greece belgium sweden turkey gibraltar portugal ireland mauritius britain cyprus netherlands norway australia italy japan canada kingdom spain austria zealand england france germany switzerland finland poland america usa iceland holland scotland uk	European country
	bow apron sweater tie anorak hose bracelet helmet waistcoat jacket pullover equipment cap collar suit fleece tunic shirt scarf belt crucifix	garment
head	voice torso back chest face abdomen side belly groin spine breast bill rump midhair hat collar waist tail stomach skin throat neck speculum	body part
	ceo treasurer justice chancellor principal founder pres- ident commander deputy administrator constable librarian secretary governor captain premier executive chief curator assistant committee patron ruler	person
squash	football gardening rounders running theatre snooker gymnastics bowling mini wrestling chess netball area fishing gymnasium lounge table soccer bar playground baseball cup basketball city ball league volleyball bath room fencing cricket weather hockey polo walking dancing boxing garden sauna terrace court golf union swimming solarium croquet tennis rugby handball development athletics archery badminton music horse pool whirlpool	sport
	apple rum mint lime cola honey coconut lemonade chocolate lemon bergamot pepper eucalyptus acid rind	foodstuff
oil	heat coal power water gas food wood fuel steam tax heating kerosene fire petroleum dust sand light steel telephone timber supply drainage diesel electricity acid air insurance petrol	object
	tempera gouache watercolour poster pastel collage acrylic	paint
lemon	bread cheese mint butter jam cream pudding yogurt sprinkling honey jelly toast ham chocolate pie syrup milk meat beef cake yoghurt grain	foodstuff
	hazel elder holly family virgin hawthorn	shrub
cherry	cedar larch mahogany water sycamore lime teak ash hornbeam oak walnut hazel pine beech alder thorn poplar birch chestnut blackthorn spruce holly yew laurel maple elm fir hawthorn willow	wood
	bacon cream honey pie grape blackcurrant cake banana	foodstuff

Table 4.1: Some test words and their senses as extracted by the word sense discovery algorithm.

A class of regular polysemous nouns in German is the set of nominalizations ending in "-ung", most of which have both a "process" and a "state" reading, where the latter is often the result or the object of the former. For example, the word form *Bemalung*, just as its English translation *painting*, denotes as well the activity of painting as its result, and the word form *Lieferung* (*delivery*) may refer to both the process of delivering and the thing being delivered.

We adjust the word sense discovery algorithm to the new way of linking words based on shared grammatical relations as follows. In Step 1, instead of discriminating against the *words* contained in previously extracted clusters, we discriminate against the *features* characterizing them. This means, we rebuild the local graph surrounding the target word w, so that we consider only those words which have a non-significant feature overlap with the already extracted word clusters.

We applied the new modified algorithm to the "-ung" nominalizations occurring in the corpus. Table 4.2 lists some of the results. Again, the clusters shown for each test word are the first two in the list of extracted clusters.

The algorithm discriminates well between the "activity/process" and "object" readings of the test words. Furthermore, the grammatical relationships characterizing a cluster provide valuable clues about which meaning a particular instance of the target word refers to.

For example, the first cluster generated for the test word *Leitung* relates to *Leitung* "pipe/powerline", i.e. to a physical object. The cluster members share the following grammatical relations:

- **subj-verb**: *lecken (leak), einfrieren (freeze), durchschmoren (melt down), brechen (break apart), etc.*
- **obj-verb**: *verlegen (lay), legen (lay), verlaufen (run), explodieren (explode), reparieren (repare), reißen (break), auswechseln (exchange), erneuern (renew), etc.*
- **adj-noun**: *defekt (defective), beschädigt (damaged), undicht (leaky), verrostet (rusty), isoliert (insulated), etc..*

On the other hand, the second cluster, which characterizes the "function, duty" sense of *Lieferung*, is defined by the features

- **subj-verb**: *zufallen (evolve upon), obliegen (be incumbent upon), zuwachsen (accrue), erfordern (require)*
- **obj-verb**: *übernehmen (accept), wahrnehmen (perform), zuweisen (assign), übertragen (delegate), erfüllen (perform), etc.*
- **adj-noun**: *wichtig (important), undankbar (thankless), exekutiv (executive), dankbar (thankful), sozial (social), etc..*

These syntagmatic features associated with each of the two clusters can help to disambiguate *Leitung* in sentences such as

Word	Sense clusters	Class label
Leitung	Rohr Trasse Pipeline Wasserleitung Kabel Behälter Faß Gasleitung Abwasserkanal Schlauch etc.	Vorrichtung
	Kompetenz Rolle Verantwortung Funktion Aufgabe	Pflicht
Erkrankung	Vorfall Verkehrsunfall Vorgang Einbruch Zwischenfall Tod Straftat Überfall Diebstahl Unglück	Vorfall
	Virus Erreger Panik Unruhe Infektion Krankheit Fieber Epidemie Seuche Angst etc.	Krankheit
Verbindung	Verbundenheit Atmosphäre Kommunikation Austausch Freundschaft Zusammenhang Beziehung Tradition Kooperation Zusammenarbeit etc.	Umstand
	Gift Dampf Stoff Dioxin Schadstoff Asbest Kohlenwasserstoff Plutonium Chemikalie Lösungsmittel etc.	Substanz
Verletzung	Verkehrsunfall Beschädigung Rauchvergiftung Knochenbruch Riß Verwundung Gesundheitsschaden Fraktur Trauma Schnittwunde etc.	Menge
	Verhalten Brauch Verstoß Vergehen Verbrechen Delikt Handlung Betrug Menschenrechtsverletzung Mord etc.	Geschehen
Bedienung	Betreuung Behandlung Versorgung Hilfe Berücksichtigung Unterbringung	Geschehen
	Krankenschwester Passant Frau Verkäuferin Bürger Dame Nachbarin Herr Leute Kellner	Lebewesen
Drogenberatung	Jugendarbeit Schutz Überwachung Betreuung Kontrolle Versorgung Überprüfung Aufsicht Beratung Untersuchung etc.	Geschehen
	Hof Gebäude Kindergarten Spielplatz Einrichtung Station Hallenbad Pforte Türe Krankenhaus etc.	Bauwerk
Versteinerung	Einführung Angleichung Beitritt Abriß Ausstieg Privatisierung Währungsunion Reduzierung	Geschehen
	Müll Kot Abfall Überbleibsel Exkrement Brot	Substanz

Table 4.2: Examples of German nominalizations ending in -ung and their learned senses.

- *Der Physiker Albert Einstein übernimmt die Leitung des Kaiser-Wilhelm-Instituts für Physik.*
- *Der Netzwerkausfall war auf eine beschädigte Leitung der Telekom zurückzuführen..*

We have given some examples to illustrate the potential usefulness of this approach to the induction of word senses. We are aware that a more thorough evaluation is needed. Our current approach is time-consuming as word senses are learned for each word separately. In the following, we will investigate global approaches to inducing word senses which analyze the whole word graph at once.

Chapter 5

Using curvature for ambiguity detection and semantic class acquisition

5.1 A linguistic interpretation of curvature

The aim of this chapter is to understand the concept of curvature, which was introduced in Section 2.3, from a semantic point of view. The definition of curvature is based on purely graph-theoretic arguments. The question addressed in this chapter is how this definition relates to the meanings of the words in the word graph.

Figure 5.1 shows the neighborhood of the word form *head* in the word graph. Two highly clustered regions, corresponding to word communities, are clearly visible: one referring to the "body part" sense and another referring to the "person in a leading position" sense of *head*.

There is a high density of triangles and thus a high degree of transitivity within each area of meaning, but transitivity breaks down at community boundaries. Words within an area of meaning, i.e. words with a precise meaning, thus have high curvature. Ambiguous words such as *head*, which connect otherwise unrelated areas of meanings, instead have low curvature because transitivity generally holds only between neighbors referring to the same sense of a word. The more meanings a word has, the less triangles it occurs in, and consequently, the lower is its curvature.

In terms of geometry, this means that neighborhoods of specific unambiguous words can be embedded in a highly positively curved surface and thus require little space (forming a ball-like structure). The neighborhood of a semantically fuzzy word, on the other hand, can only be accommodated on a surface of negative curvature which takes up much more room in semantic space. Clusters of high curvature words correspond to cohesive semantic communities around which the space of word meanings is spherical in nature, i.e. very compact. Ambiguous words live in between such word communities. Their neighborhoods take up more space. Ambiguity locally molds the semantic space into a saddle.

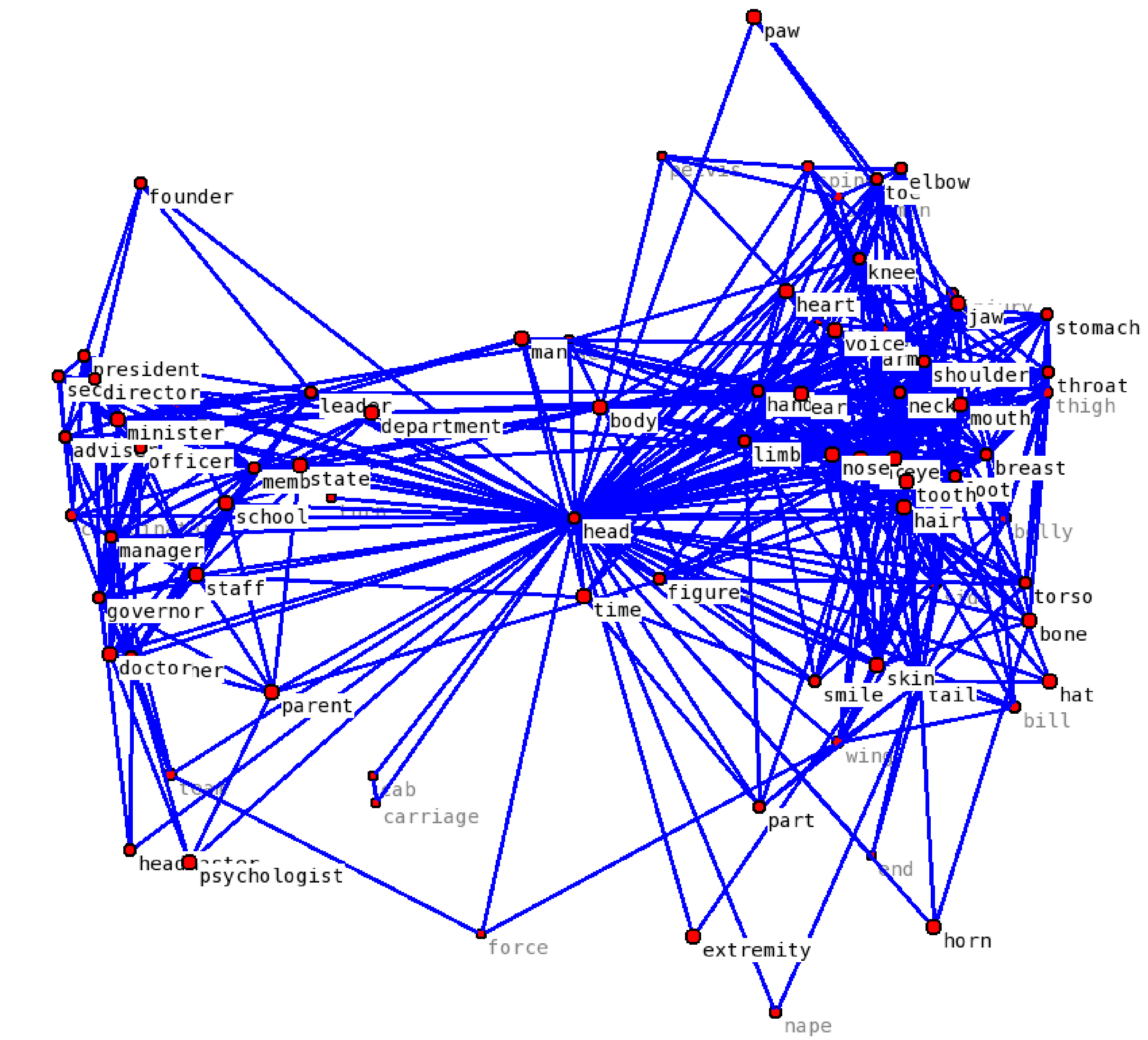


Figure 5.1: The neighborhood of the ambiguous word form *head*. We can identify two sphere-like clusters of highly connected nodes corresponding to different meanings of *head*. These are connected by a small number of low curvature nodes.

5.2 Curvature as a measure of ambiguity

Word frequency is often used as a measure of a word's information content or ambiguity (Resnik, 1999; Lin, 1998). The idea, which is rooted in information theory (Shannon, 1948), is that rare (unexpected) words yield much more information than very frequent (expected) ones.

Most high frequency words have a very general scope and are often used in a figurative sense. The familiar word *head*, for example, in addition to being used in its original meaning of "top part of the human body", is often employed to refer to the top or chief part of all kinds of concrete or abstract things including nails, spears, bones, departments, schools and movements. Similarly, the word *fruit* denotes the "fruits of a tree" or the state of "being in fruit", but it can also be used metaphorically to refer to the outcome of something as in "the fruit of labor".

Whereas ambiguity tends to increase with word frequency, frequency alone is not a very reliable indicator of ambiguity. There are many examples of rare words which have multiple meanings. Among the most ambiguous low-frequency words are acronyms and abbreviations, such as *ge*, which may refer either to a chemical element (*Germanium*) or to an American company (*General Electrics*). Other examples are names of companies or products which are borrowed from the pool of already existing words (e.g., *condor* and *python*). We shall see in the following that curvature better quantifies ambiguity than measures based on word frequency alone.

Among the most frequent words covered by the graph model are names of countries. Figure 5.2 is a scatter plot of word frequency versus curvature for the words in the graph. Names of countries (indicated by black dots) have considerably higher curvature than other words of comparable frequency. The curvature measure correctly recognizes that, despite their high frequencies, names of countries are very informative, that is unambiguous words. The outlier is the word *monaco* which may not seem ambiguous, but which indeed has several different meanings in the BNC: country, city, 14th century painter and 20th century tenor (cf. Figure 5.3).

Just as names of countries are set apart from other frequent words by high curvature values, we find many family names among the low frequency - low curvature words. In addition to different persons sharing the same name, family names are often taken from names of colors (Green, Black, White, Brown), places (Wolverton, Adlington, Barnham), professions (Smith, Taylor, Marshall), natural objects (Hill, Bush, Wood, Moore), animals (Crane, Swift), or first names (John, Martin), which makes them highly ambiguous.

To confirm the empirical finding that curvature better quantifies ambiguity than word frequency, we check how well each of the two measures predicts the words' "true" ambiguity index as determined by their WordNet polysemy counts (number of different senses). Instead of word frequency, i.e. the total number of occurrences of a word, one might also measure a word's level of ambiguity as its degree in the word graph, i.e. the number of different words it occurs with.

We consider all words in the graph model which are listed in WordNet and compute a ranking of these words for each of the three measures: curvature, frequency and degree. The

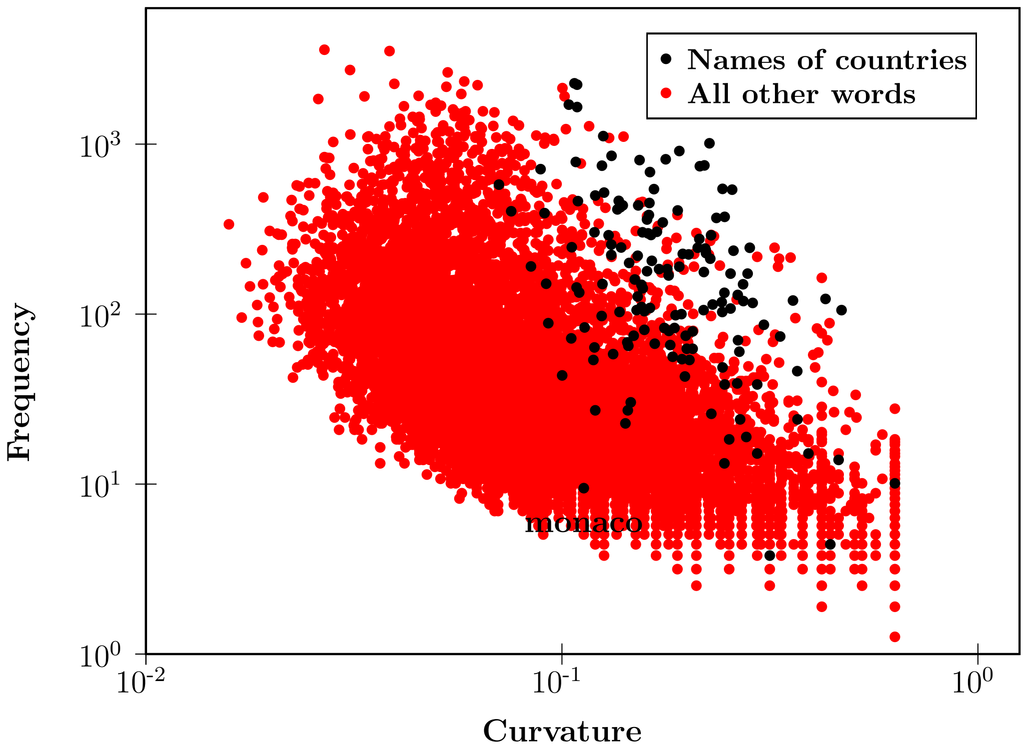


Figure 5.2: Scatter plot of curvature vs. frequency. Note that countries (black dots) have substantially higher curvature values than other words with similar frequencies, meaning that they are very specific. The outlier is *monaco*.

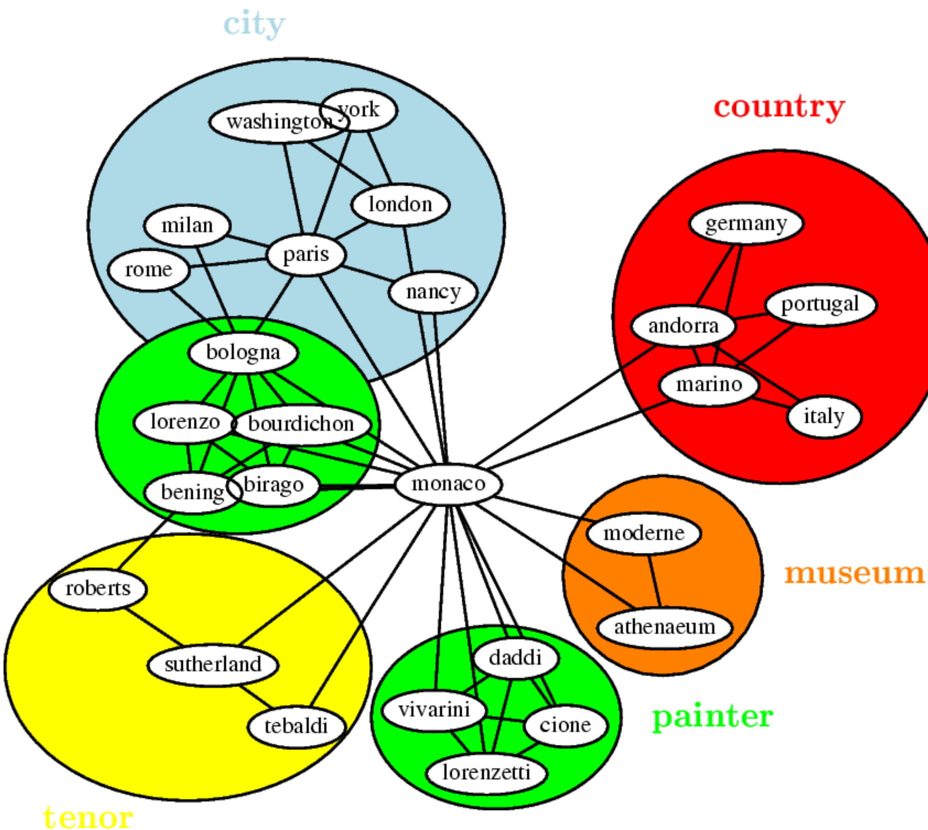


Figure 5.3: Local neighborhood of the word form *monaco*.

different rankings are compared to each other as well as to the ranking based on the WordNet polysemy counts which serves as the benchmark. Table 5.1 contains the correlations between the different rankings as estimated by the Pearson coefficient (Pruscha, 1996). We find that, with a negative correlation of -0.538, curvature is more strongly related to the WordNet-defined benchmark than frequency (0.475) or degree (0.480), and thus the better predictor of ambiguity [1].

	WN	Freq	Deg	Curv
WN	1.000	**0.475**	**0.480**	**-0.538**
Freq		1.000	0.963	-0.865
Deg			1.000	-0.884
Curv				1.000

Table 5.1: Rank correlations between any two out of WordNet polysemy count, word frequency, degree and curvature.

The curvature approach to measuring ambiguity shares some similarity with the approach proposed by Sproat and van Santen (1998), in that curvature also quantifies ambiguity based on the semantic cohesiveness of the target word's neighborhood. Words with a very tightly-knit neighborhood are assigned smaller ambiguity scores than words whose neighborhood is rather fuzzy.

5.3 Curvature as a clustering tool

Besides being a useful tool for measuring ambiguity, the curvature measure immediately gives rise to an extremely powerful tool for separating the word graph into groups of words with similar meanings.

The high average curvature of the nodes in the word graph is indicative of the presence of community structure. This means that whereas the word graph as a whole is sparsely connected, it contains many highly connected regions. These node communities correspond to semantically homogeneous groups of words, i.e. words denoting the same or similar concepts.

Ambiguous words function as "semantic hubs". They blur community boundaries by connecting together different communities which evoke unrelated concepts. As we have seen in the previous section, the curvature measure allows us to locate the ambiguous words by their low curvature values. To reveal the community structure underlying the word graph, we attack the low-curvature hubs. The word graph naturally fragments into isolated islands of high-curvature nodes. These correspond to semantically coherent clusters of unambiguous words. More precisely, we compute the curvature of the nodes in the graph

[1] The strength of the relationship is given by the absolute value of the correlation coefficient. The sign merely defines the direction of the relationship.

and delete all those nodes and their adjacent links whose curvature falls below a given threshold. This causes the word graph to break into several disconnected pieces (clusters).

The particular choice of the curvature threshold regulates the trade-off between the coverage and the coherence of the clustering. Words failing to pass the curvature threshold are not contained in the resulting clustering and hence are not classified. Imposing a high curvature threshold therefore results in low coverage, but at the same time yields small coherent clusters. In contrast, the more lenient the curvature threshold, the more words exceed it, i.e. are classified, but the bigger and more heterogeneous the clusters tend to be.

5.3.1 Determining the curvature threshold

In order to find a suitable curvature threshold θ, we examine how changing θ affects the fragmentation of the graph. A quantity which characterizes the degree of fragmentation in a graph is the proportion of nodes belonging together in the largest fragment (connected component): The smaller the largest component, the more fragmented is the graph.

Erdös and Rényi (1960) found that randomly wired graphs (any two nodes are linked with a certain probability p) show an abrupt change in their connectivity as the wiring probability p passes a critical value p_0. Below the critical threshold, the graph consists of a multitude of small isolated components. But as p approaches p_0 a giant cluster, comprising the vast majority of the nodes, suddenly emerges, and the graph is said to *percolate*.

Even more interesting is to study percolation in *real-world* graphs. Most real-world networks are characterized by the existence of a giant component. By observing the behavior of this giant cluster as nodes and/or edges are deleted, it is often possible to make inferences about the stability of the network.

Nodes belonging to the same connected component of a graph can be reached from each other. That is, the size of the largest component measures the degree of communication between the nodes and thus strongly influences the functionality of the network.

The giant component shrinks with the number of nodes (links) being disabled and eventually disappears. Its behavior depends very much on the order in which nodes (links) are removed. For example, random removal of nodes often leads to a gradual decrease in the size of the giant component. Removing the most connected nodes first, instead leads to an abrupt collapse of the giant component, i.e. communication breakdown, as the number of removed nodes exceeds a critical value (the percolation threshold).

The study of the percolation properties of a network has many applications, for example, in epidemiology, where it can help in the design of vaccination strategies aimed at protecting the population at risk against an infectious disease. Vaccination of a person is modeled as removal of a node from the social network, and the percolation threshold corresponds to the number of vaccinations necessary to prevent the disease from spreading. If people with many social contacts are vaccinated first, the disease can be controlled much more efficiently and effectively.

Seeking guidance on the choice of the curvature threshold, we will study the percolation properties of the word graph as we vary the curvature threshold, which is our control parameter of ambiguity. If there occurs a sharp transition in the connectivity of the word graph, then the curvature value at which it occurs corresponds to the critical amount of ambiguity which has to be eliminated in order to break the word graph into coherent meaning clusters.

We use $S_{\text{rel}}(\theta)$ to denote the relative size of the largest component for a given value of θ, which is simply the size of the largest component divided by the total number of nodes in the graph. Figure 5.4 shows the behavior of $S_{\text{rel}}(\theta)$ as a function of the curvature threshold θ (black line). Low curvature thresholds θ give rise to a large giant component, whereas high θ-values result in many small isolated clusters. The most substantial change in fragmentation occurs within the interval $0.1 \leq \theta \leq 0.3$, where slight changes in the curvature threshold θ are associated with significant changes in $S_{\text{rel}}(\theta)$. The curve levels off at $\theta = 0.35$.

For comparison, the figure also includes the fragmentation values which are obtained by removing the same number of nodes at random rather than by increasing order of curvature (red line) [2]

When nodes are removed in random order, the largest component shrinks only very slowly. Even after a significant number of nodes has been removed, the largest cluster still covers a substantial fraction of the graph. In order to break the graph, nearly all nodes have to be removed. On the other hand, if low curvature nodes are removed first, the largest cluster disintegrates very quickly. This supports our intuition that it is the low curvature nodes, corresponding to ambiguous words, which are chiefly responsible for holding the word graph together.

Until now, we have only examined the behavior of the *largest* component, neglecting all other components. However, studying the small components can also be illuminating. We do this by counting the number of clusters containing at least 5 nodes, which we denote by $N_5(\theta)$, and by recording this number for different values of θ (cf. Figure 5.5). For small values θ (≤ 0.1), $N_5(\theta)$ is extremely small, which means that the removal of only a negligible number of the lowest curvature nodes causes only small pieces to break off the largest component. As θ increases, bigger clusters suddenly start falling off. However, these clusters are quickly broken apart themselves as θ gets even larger. Table 5.2 summarizes properties of the clustering for different values of θ. These include the size ($S_{\text{abs}}(\theta)$) and density of the largest cluster, the size of the second largest cluster ($S^{(2)}_{\text{abs}}(\theta)$), as well as the number $N_5(\theta)$ of clusters of size ≥ 5. We calculate the density of the largest cluster as the ratio of the number of links to the number of nodes it contains. The density of the largest cluster is a measure of its coherence.

An optimal curvature threshold θ has to be large enough to ensure that the largest cluster has internal coherence and no longer dominates the graph. At the same time, θ shouldn't be as large as to cause the small components to split.

Whereas $N_5(\theta)$ reaches its maximum at $\theta = 0.25$, the giant component disappears no

[2]The values are averaged over several runs. The apparent dip at $\theta = 0.675$ is due to a large number of nodes having a curvature which is ≥ 0.65 but < 0.675. When θ increases from 0.65 to 0.675 a much larger number of nodes is removed at once than when moving from $\theta = 0.625$ to $\theta = 0.65$ or from $\theta = 0.675$ to $\theta. = 0.7$. If plotted against the number of nodes instead of against θ, $S_{rel}s$ plots as a smooth curve.

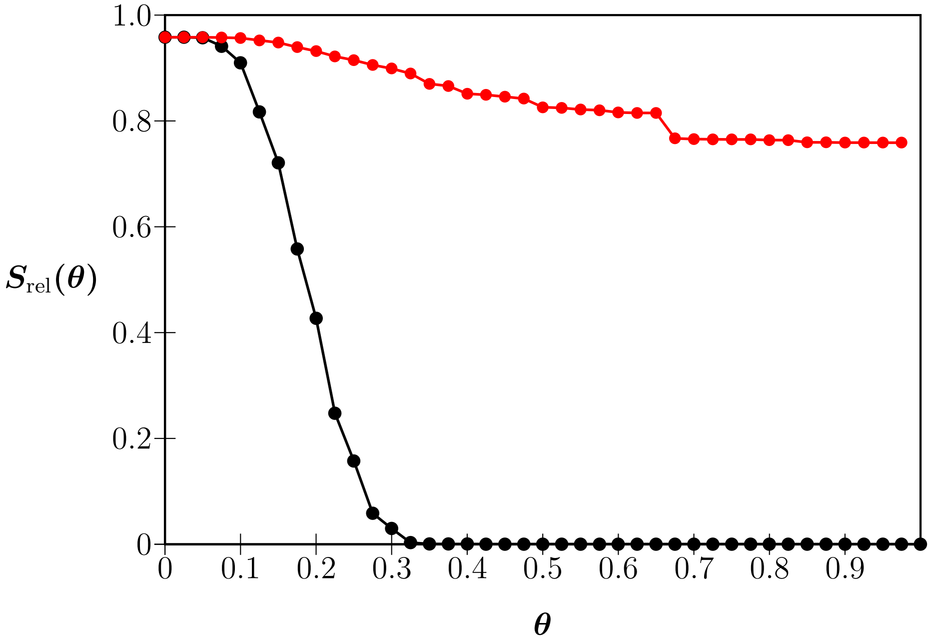


Figure 5.4: Relative size $S_{\mathrm{rel}}(\theta)$ of the largest cluster as a function of the curvature threshold θ. The red line shows the effect of removing the same number of nodes at random.

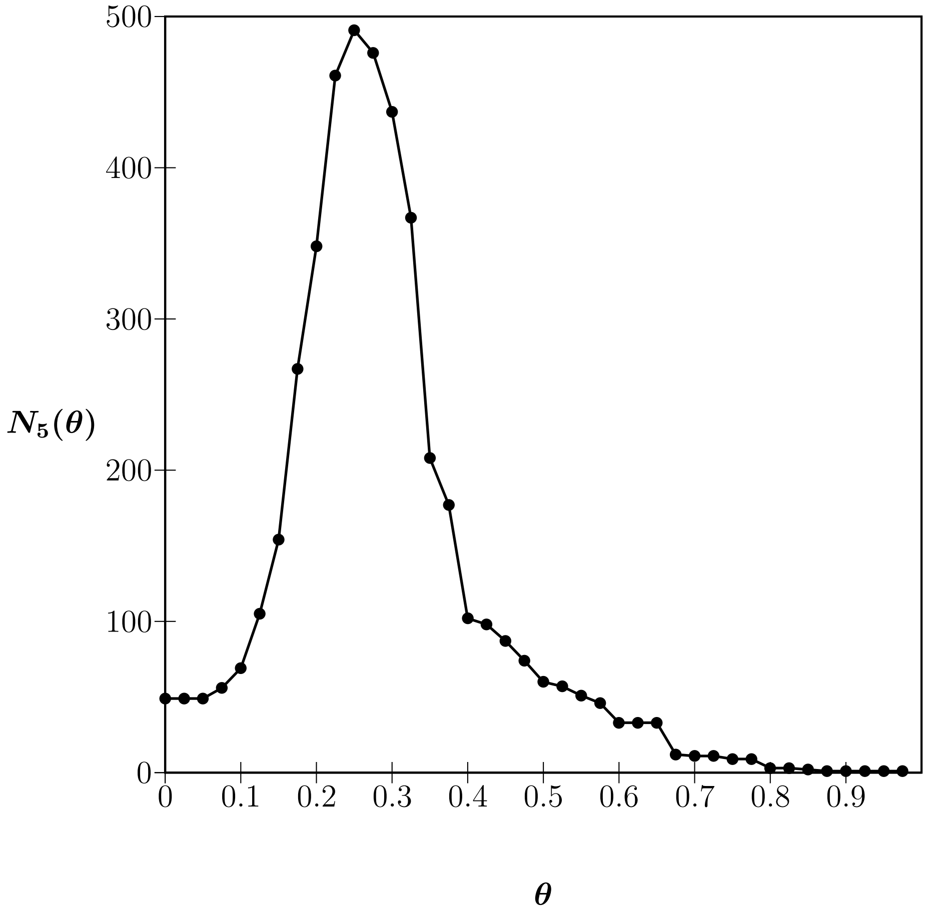


Figure 5.5: The number of clusters comprising at least five words as a function of θ.

sooner than at $\theta = 0.325$. This is however the point at which it is sparsest, i.e. least coherent. We decide to trade the coverage of the clustering for its accuracy and choose $\theta = 0.35$ as the threshold for curvature clustering to ensure homogeneous clusters. Inspection of the clusterings associated with different values of θ further confirms this choice.

Using $\theta = 0.35$ as the curvature cut-off point, we obtain a graph which comprises $28,763$ nodes. The majority of the nodes ($16,731$) are isolated nodes (not connected to any other nodes). The remaining $12,032$ nodes are distributed across $4,770$ clusters of size ≥ 2.

θ	#nodes	$S_{\text{abs}}(\theta)$	$S^{(2)}_{\text{abs}}(\theta)$	density	$N_5(\theta)$
0.2	39,809	17,006	32	2.70	348
0.225	37,471	9,277	42	1.95	461
0.25	36,080	5,681	44	1.68	491
0.275	34,191	1,997	241	1.37	476
0.3	33,061	986	121	1.26	437
0.325	31,496	96	79	1.10	367
0.35	28,763	20	19	1.45	208
0.375	28,296	14	13	1.50	177
0.4	26,618	13	10	1.85	102

Table 5.2: Some properties of the curvature clustering associated with different curvature thresholds θ.

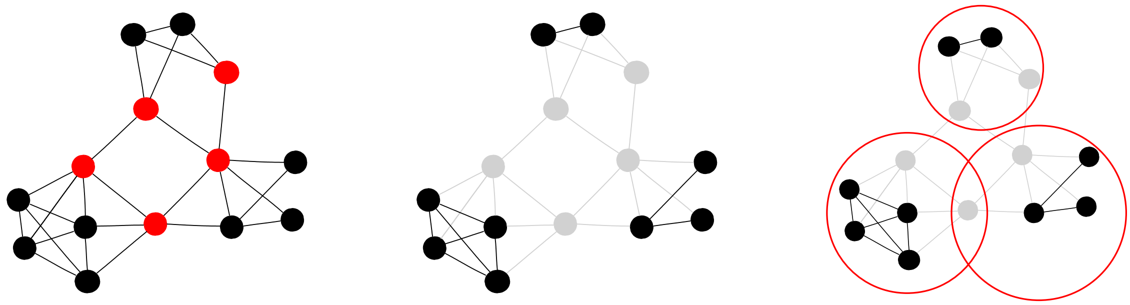
5.3.2 Allowing for overlapping clusters

Curvature clustering in the current form covers only the high curvature words, that is, words with a well-defined meaning. Information on the meaning of the ambiguous low curvature words is not provided. By augmenting each of the curvature clusters with those nodes in the original word graph which are directly attached to it, we can expand the clustering to cover also words with a less precise meaning. Moreover, since a node may attach to more than one cluster, this process allows for multiple cluster membership. This means that an ambiguous word can simultaneously belong to several clusters corresponding to its various meanings. As an example, *mouse* links to both the (*scsi, trackball, joystick*) and the *pika, terrapin, marmot, hamster, gerbil* cluster (cf. Table 5.8), and is hence added to both of them. The steps of the extended curvature clustering algorithm (also illustrated in Figure 5.6), are the following:

1. Compute the curvature of all nodes in the graph.

2. Remove all nodes whose curvature falls below a certain threshold ($\theta = 0.35$). This decomposes the word graph into a set of non-overlapping clusters.

3. Expand each of these crisp clusters with its immediate neighboring nodes to produce a "soft" clustering of the word graph. Only neighbors with at least two links to the cluster are added.



(a) Identify hubs (red). (b) Delete hubs and their links. (c) Add neighbors to clusters.

Figure 5.6: Illustration of the curvature clustering.

Expansion of the 4,770 clusters resulting from Step 2 with their neighbors (Step 3), turns the initially crisp clustering into a fuzzy clustering which now covers 16,446 words (as opposed to 12,032 before expansion). We try to avoid spurious cluster members by adding only nodes to a cluster which are neighbors of at least two of the cluster members.

Table 5.8 lists some of the enriched clusters. Original cluster members (before expansion) are shown in black, cluster neighbors added in Step 3 are shown in gray font. The cluster cores behave similarly to the committees in Pantel and Lin's *Clustering by Committee* approach (Pantel and Lin, 2002) (cf. Section 9.2.3). Words in the core of a cluster (original cluster members) are very specific and well-defined. The new cluster members, on the other hand, have a wider range of meaning or are even ambiguous. For example, *applewood* and *fruitwood* both have a single specific meaning, namely they are both kinds of *wood*. *Cherry*, *pine* and *oak*, instead, can also refer to a *tree*, and *cherry* has yet another meaning, namely that of *fruit*.

Words in the core tend to be the less well-known representatives of a semantic category. If we were to list the world religions, we would probably start with the most common ones: *islam*, *buddhism*, *hinduism*, *christianity* and *judaism*. Only later in the list, if at all, we would come up with *jainism*, *sikhism* and *vaisnavism*. It seems that because of their low profile (i.e. high information content), these words are especially well-suited to determine which other words should belong to the category.

Since only the core members decide which new words are added to the cluster, even if a word is mistakenly accepted, this won't cause "infection" (Roark and Charniak, 1998), meaning that the neighbors of the wrongly-admitted words won't be added to the cluster as they are not linked to the cluster core. It is also important to note that the cluster extension step doesn't affect the purity of closed semantic categories like, for example, the cluster composed of the four nucleotide bases, (*adenine, uracil, guanine, thymine*).

scsi trackball joystick adaptor mouse modem interface disk
pika terrapin marmot hamster gerbil donkey dog mouse hedgehog rat sheep cat rabbit
applewood fruitwood cherry ivory pine oak
jainism sikhism vaisnavism islam buddhism hinduism christianity judaism
osmium iridium rhodium ruthenium silver platinum gold
rubella cytomegalovirus poliomyelitis enteritis mumps diphtheria measles tetanus pox disease tuberculosis disorder polio cough
cuboid octahedron icosahedron tetrahedron dodecahedron cone cylinder pyramid cube
adenine uracil guanine thymine
collarbone plexus forearm sternum groin wrist head heart chest neck rib spine nose hand
tartar mongol bedouin yakut tatar evenki turk pole jew russian eskimo greek people
slough maidenhead london reading sheffield
agraphia dyslexia dysgraphia reading spelling
pantaloon lapel anklets sash boot collar button dress jacket scarf shirt
nutritionist dietitian dietician expert physiotherapist doctor representative psychologist
cerise umber ultramarine red green ochre violet blue
brasserie bistro cafe bar bars restaurant
tempera gouache pen oil paint
glycerine lanolin hexane fragrance oil
kerosene gasoline fuel oil
nutmeg tofu tablespoon breadcrumb paprika allspice anise sultana peppercorn cassia tabasco treacle pepper yoghurt egg garlic margarine clove oil almond
alstroemeria gladiolus freesia gypsophila carnation rose dahlia lily chrysanthemum flower daisy
laptop portable adapter notebook workstation desktop pc device
stalactite stalagmite breccia formation cave rock

Table 5.3: Some examples of clusters resulting from the curvature approach.

5.4 Evaluation against WordNet

In this section, we will complement the qualitative analysis of the curvature scheme with a more formal quantitative evaluation. In order to assess the validity of the curvature scheme, we need to measure how well a word's cluster affiliations agree with its different meanings. To do this objectively, we would need large amounts of sense-tagged data. However, sense-tagged corpora are both few and small. For this reason, we have chosen to resort to WordNet as the "gold-standard" sense inventory against which we evaluate the performance of the curvature algorithm.

We are aware that this evaluation procedure is not ideal as the word senses covered by the BNC do not map one-to-one to WordNet synsets. Not all sense distinctions made by the creators of WordNet necessarily appear in the corpus. Similarly, the corpus contains word meanings which are not represented in the WordNet lexicon. In particular, corpus and lexicon will differ in terms of sense granularity [3]. Nevertheless, we expect an evaluation against WordNet to provide insight on the strengths and limitations of the curvature approach.

Naturally, the target senses of non-WordNet words are unknown. For this reason, we perform evaluation on $\mathcal{W}_{\text{wn}}$, the graph which is induced by the set of nouns which are known to WordNet. $\mathcal{W}_{\text{wn}}$ is a subgraph of the full word graph $\mathcal{W}$ and consists of $21,082$ nodes (words) and $617,947$ edges (relationships). Application of the curvature clustering algorithm to $\mathcal{W}_{\text{wn}}$ results in 730 (partly overlapping) clusters which together cover $3,578$ words.

5.4.1 Precision and Recall

An ideal clustering tool would assign words to clusters in such a way that, for each word, there is a one-to-one correspondence between its cluster affiliations and its WordNet senses. In the following, we will describe an evaluation procedure which quantifies the degree to which the curvature scheme deviates from this optimal clustering, using the criteria of *precision* and *recall* which are traditionally used in evaluating the performance of search engines.

We define precision and recall of an individual word as follows. The precision of a word is the proportion of retrieved *true* meanings out of the *total* number of retrieved meanings. The recall of a word is the ratio of retrieved true meanings to the total number of true meanings:

$$Precision(w) = \frac{\text{\# retrieved true meanings of } w}{\text{\# retrieved meanings } \ w}. \tag{5.1}$$

$$Recall(w) = \frac{\text{\# retrieved true meanings of } w}{\text{\# true meanings of } w}. \tag{5.2}$$

[3] WordNet is known to make extremely subtle sense distinctions in certain domains.

In the following, we will illustrate the two evaluation measures with some examples. We find different types of correspondences between a word's clusters on one side and its WordNet synsets on the other side. The word *mouse* shows an ideal match of clusters to WordNet senses. It occurs in exactly two clusters, each of which maps to one of its two WordNet senses (cf. Figure 4.1). Consequently, *mouse* has both a precision and recall of 100%. In Section 5.4.2, we will give a detailed description on how we determine whether a cluster corresponds to a WordNet synset. For now, we will simply assume that we have such a tool available which tells us whether a cluster actually represents a true WordNet sense of a word.

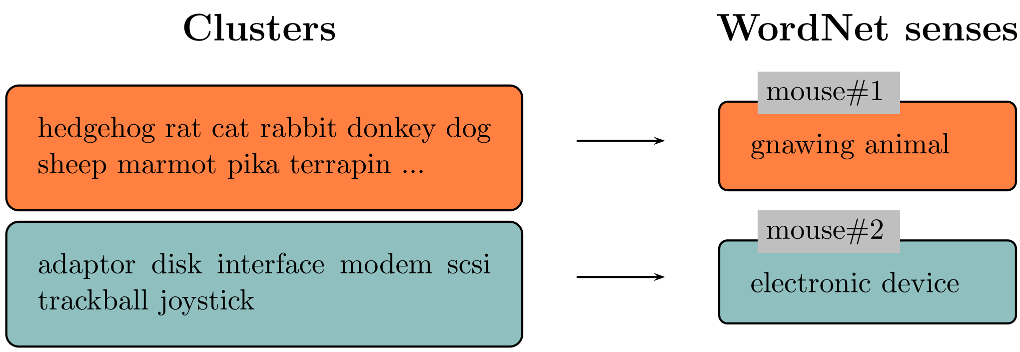


Figure 5.7: The clusters of *mouse* map one-to-one to its WordNet senses.

For most words, we observe only a partial match between a word's clusters and its different meanings in WordNet. Consider the word *oil* (cf. Figure 5.8). Just as in the case of *mouse*, we succeeded in finding both WordNet meanings of *oil*, which means *oil* has maximum recall, too. However, two of the clusters are left unmatched, resulting in a precision of less than 100%[4]. More precisely, the curvature scheme retrieves five senses of *oil* three of which can be accurately mapped to WordNet senses of *oil*. Hence, the precision of *oil* is 3/5. In fact, two of the clusters map to the *same*, "lipid" sense of *oil*. Counting both of the clusters as correctly identified senses of w, unfairly raises precision. To penalize for duplicate senses, we consider only one of the clusters as being correctly identified, and the other as being an incorrectly extracted sense of *oil*[5]. Hence, the revised precision of *oil* is 2/5.

A word whose precision is maximal but recall isn't, is the word *recorder* which appears in two clusters, one mapping to its "recording machine" sense, the other mapping to its "musical instrument" sense. However, *recorder* has two additional senses in WordNet which we failed to discover. We correctly extracted two of the four WordNet senses of *recorder*, which means a recall of 0.5.

To put it simply, the precision of a word is obtained from a diagram such as those

[4]The "food" sense of *oil* is not encoded in WordNet, although it probably should. Consequently, the precision of *oil* is judged lower than it actually is.

[5]We adopt this approach from Pantel and Lin (2002).

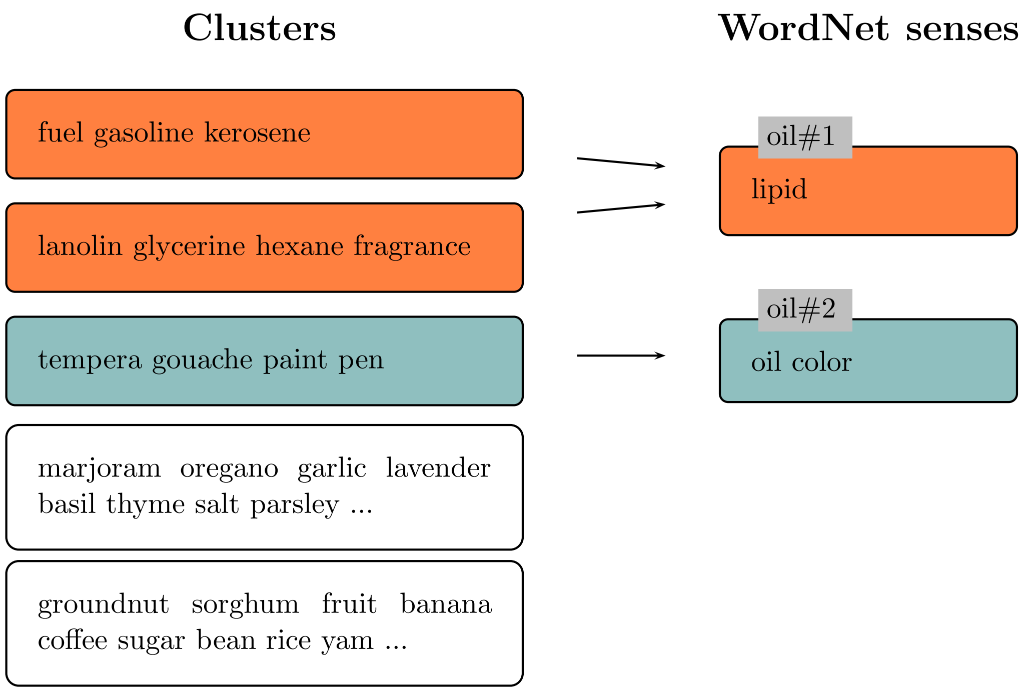


Figure 5.8: The clusters *oil* appears in and their mappings to WordNet senses. There are two clusters which map to the same WordNet sense *lipid*. Two other clusters, reflecting the *food* aspect of *oil*, cannot be mapped to a WordNet sense of *oil*.

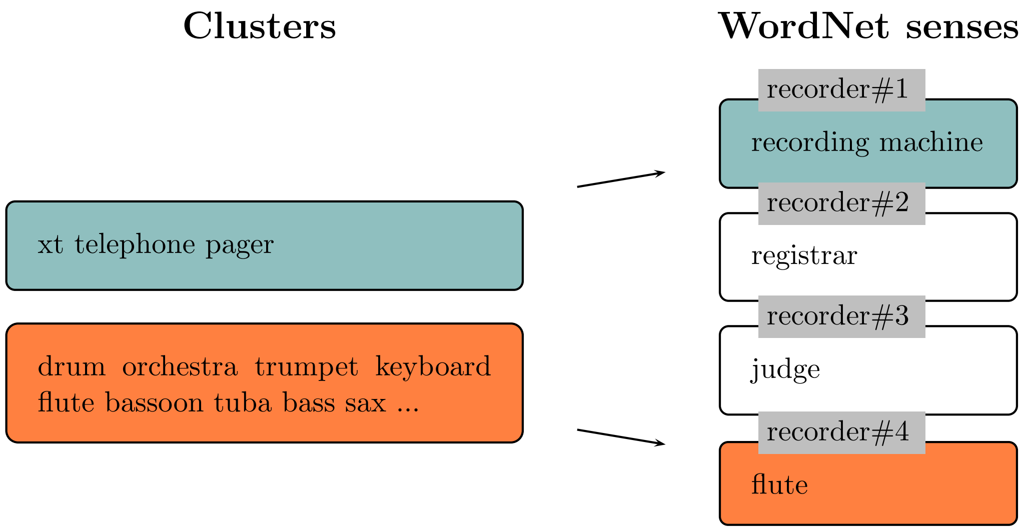


Figure 5.9: The clusters *recorder* appears in and their mappings to WordNet senses. Curvature clustering failed to discover the *registrar* and the *judge* sense of *recorder*.

shown in Figures 5.7, 5.8 and 5.9 by counting the number of different colors (not counting white as a color) and dividing by the total number of clusters (the number of boxes on the left side of the diagram). Similarly, the recall of a word is the number of different colors divided by the total number of its WordNet senses (the number of boxes on the right side). We define precision and recall of a clustering scheme simply as being the average recall and precision of all words:

$$Precision = \frac{1}{\#w} \sum_{w} Precision(w) \tag{5.3}$$

$$Recall = \frac{1}{\#w} \sum_{w} Recall(w) \tag{5.4}$$

A single measure of the overall efficiency of a system is the F-score which is the harmonic mean of precision and recall:

$$F = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{5.5}$$

It assumes high values when both precision and recall are high. When precision and recall are close to being equal, the F-score is near the arithmetic mean of the two. The bigger the imbalance between the two, the more F deviates from the arithmetic mean, penalizing systems which have high precision at the expense of low recall, or high recall at the expense of low precision.

5.4.2 Mapping clusters to WordNet senses

We employ the methodology of Pantel and Lin (2002) to check whether a cluster in which a word participates does indeed reflect a *true* sense, that is a *WordNet* sense of the word.

WordNet provides a set of candidate senses for each word w which we will denote by S_w. Similarly, we use C_w to denote the set of clusters of which w is a member. We map each of the clusters $c \in C_w$ to the most similar candidate sense $s \in S_w$, where similarity between a cluster c and a WordNet sense s is measured as the average similarity of each fellow cluster member $\tilde{w}$ of w in c with s:

$$sim(c, s) = \sum_{\tilde{w} \in c, \tilde{w} \neq w} sim(\tilde{w}, s).$$

Similarity between a word $\tilde{w}$ and a word sense s, in turn, is defined as the maximum similarity between s and a sense $\tilde{s}$ of $\tilde{w}$:

$$sim(\tilde{w}, s) = max_{\tilde{s} \in S(\tilde{w})} \ sim(\tilde{s}, s).$$

Finally, the similarity between two word senses (i.e. WordNet synsets) is computed as proposed by Lin (1998) and implemented by (Pedersen et al., 2004) in the WordNet::Similarity package:

$$sim(\tilde{s}, s) = \frac{2 * logP(lcs)}{logP(\tilde{s}) + logP(s)}, \tag{5.6}$$

where lcs is the lowest common subsume of $\tilde{s}$ and s in the WordNet hypernymy tree. The (empirical) probability $P(s)$ of a synset is defined as the relative frequency of all its descendant synsets in a reference corpus, the BNC in our case. The intuition behind this definition of similarity is rooted in information theory: The higher the probability of encountering a concept (synset) in the corpus, the lower is its information content. Synsets whose lowest common subsumer is very informative (infrequent) are deemed more similar than synsets which share only a highly frequent and hence uninformative superordinate term in common.

As an example of a mapping of a cluster to a WordNet sense, let us consider the word *beauty* which occurs in the cluster $\{$*beauty, pleasantness, tranquility*$\}$. *Beauty* has the following senses in WordNet:

- beauty#1: The qualities that give pleasure to the senses.
- beauty#2: A very attractive or seductive looking woman.
- beauty#3: An outstanding example of its kind.

To determine which of these senses to map the cluster to, we compute the similarities of each of the senses of *beauty* with each of the possible WordNet senses of *tranquility* and *pleasantness*. These are listed in Table 5.4. The similarity between a sense of *beauty* and the word *tranquility*, resp. *pleasantness*, is given by the maximum of the two similarity values in the corresponding table cell, e.g.

$$\begin{aligned} sim(\text{tranquility}, \text{beauty\#1}) &= max_{i=1,2}\ sim(\text{tranquility\#i, beauty\#1}) \\ &= max(0.327, 0) = 0.327. \end{aligned}$$

The bottom row of the table contains the similarities $sim(c, \text{beauty\#i})$, $i = 1, 2$, or 3, between the cluster and the different senses of *beauty* which are obtained by averaging over the maximum scores of a column's cells, e.g.

$$\begin{aligned} sim(c, \text{beauty\#1}) &= 0.5 * (sim(\text{pleasantness\#2, beauty\#1}) + sim(\text{tranquility\#1, beauty\#1})) \\ &= 0.5 * (0.384 + 0.327) = 0.356. \end{aligned}$$

The sense with the highest similarity to the cluster is *beauty#1* (0.35), and thus the sense to which the cluster is being mapped.

Mappings $c \mapsto s$ for which $sim(c, s) \geq \theta$ are said to correspond to correctly inferred word senses. Naturally, the higher the similarity threshold θ, the more clusters will be rejected as wrongly identified word senses.

5.4.3 Results and Conclusions

Figure 5.10 shows precision, recall and F-score of the curvature scheme using different similarity thresholds θ. Naturally, as θ increases, all three measures decrease, since more and more clusters fail to pass the "validity" test. Not surprisingly, recall is lower than

	beauty#1	beauty#2	beauty#3
pleasantness#1	0	0	0.231
pleasantness#2	0.384	0	0
tranquility#1	0.327	0	0
tranquility#2	0	0	0.280
sim$(c, .)$	**0.356**	0	0.256

Table 5.4: Similarities between the different senses of *beauty* and those of *pleasantness*, resp. *tranquility* as measured by Formula 5.6.

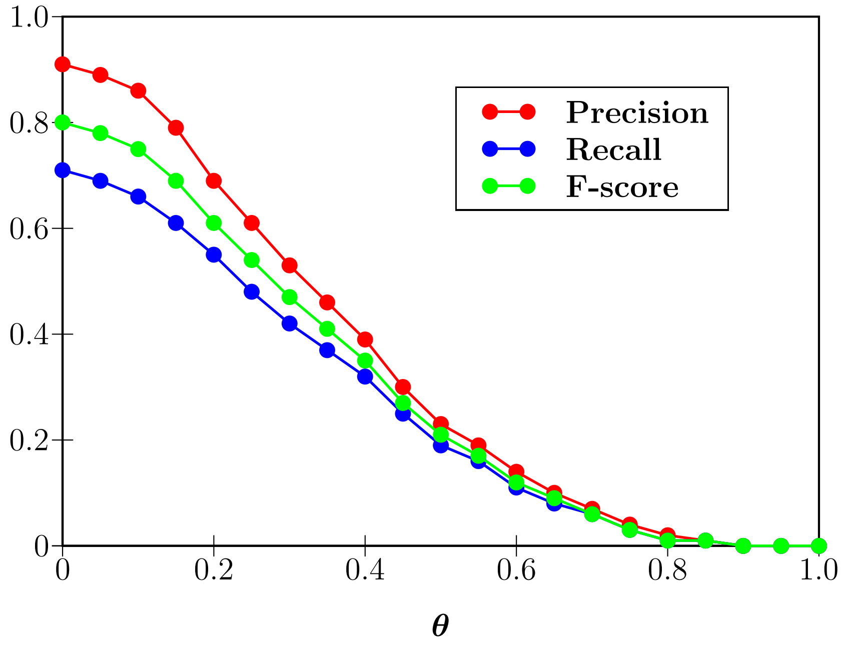


Figure 5.10: Precision, recall and F-score using different similarity thresholds θ.

precision. In any event, we didn't expect all the many and fine-grained sense distinctions used in WordNet to show up in the corpus.

Imposing a similarity threshold of $\theta = 0.25$, we obtain a precision of 61.1% which is slightly higher than the precision of 60.8% reported by Pantel and Lin (2002) for their word sense discovery system and the same θ-value. Pantel and Lin achieve a higher recall than we do (50.8 vs. 48.4). However, the numbers are not directly comparable, as the set of target senses used in computing recall differ. While we take all the words' WordNet senses as the "true" senses to be discovered, Pantel and Lin use only a subset of these as the set of target senses to be retrieved. Their set of target senses consists of the collective "true" results (correctly identified WordNet senses) of several clustering algorithms. Hence, the lower recall value observed for the curvature scheme may simply reflect the stricter approach we follow to measure the recall of our system.

It has to be noted that the fairly high performance of curvature clustering comes at the expense of low coverage. The curvature scheme covers only a relatively small number of nodes (3,578 altogether), since many of the more general and ambiguous nouns do not pass the curvature threshold or do not attach directly to a cluster. Consequently, they are left uncategorized.

Tables 5.5 and 5.6 break down the number of correctly inferred word senses (using $\theta = 0.25$) by a word's WordNet polysemy count, respectively by the number of clusters a word belongs to. Each cell (i, j) of Table 5.5, contains the number of words which occur in j clusters i of which correspond to true and distinct WordNet senses. Words falling into cell (i, j) therefore have a precision of (i/j). Similarly, each cell (i, j) of Table 5.6 contains the number of words which have j senses in WordNet i of which the curvature scheme was able to identify, which means the recall of a word falling into cell (i, j) of Table 5.6 is (i/j).

# discovered senses	**# clusters**											**Total**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**14**	
0	1152	58	13	4	5	2	0	0	0	0	0	1234
1	2047	154	53	24	6	3	1	1	0	0	0	2289
2		20	7	10	3	4	1	1	0	1	1	48
3			1	1	2	1	0	1	1	0	0	7
Total	3199	232	74	39	16	10	2	3	1	1	1	3578

Table 5.5: Frequency table of the number of correctly identified word senses broken down by the number of clusters a word occurs in.

The words covered by the curvature scheme are the high curvature ones, the majority of which have only one sense in WordNet which is reproduced quite reliably. We expect the set of high curvature words to contain many "novel" words, since most newly coined words relate to very specific domains, and are introduced to name, for example, newly discovered phenomena or new technologies. The purpose of the following sections is to estimate the potential of the curvature scheme to learn the meanings of novel words.

# discovered senses	# WN senses									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	715	226	110	58	39	22	20	17	12	5
1	1355	474	188	93	49	41	35	8	9	15
2		11	4	10	9	6	3	1	1	0
3			0	0	0	1	2	1	0	3
Total	2070	711	302	161	97	70	60	27	22	23

# discovered senses	# WN senses										Total
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**29**	**32**	
0	1	3	3	0	0	2	1	0	0	0	1234
1	9	1	3	3	2	1	0	1	1	1	2289
2	0	2	0	0	1	0	0	0	0	0	48
3	0	0	0	0	0	0	0	0	0	0	7
Total	10	6	6	3	3	3	1	1	1	1	3578

Table 5.6: Frequency table of the number of correctly identified word senses broken down by the number of a word's WordNet senses.

5.5 Uncovering the meanings of novel words

We believe that curvature clustering is particularly well-suited to learning the meanings of previously unseen words. These abound in specialized vocabularies, such as those of medicine or technology, which grow extremely quickly. Medical neologisms, for instance, include names of newly emerging diseases, novel treatments or medications. Most of these newly-coined terms are unambiguous and have a very concise meaning. They are found among the high-curvature words in the word graph which are well-covered by the curvature clustering.

In this section, we investigate how well curvature clustering predicts the meanings of novel words. We follow the evaluation procedure proposed by Ciaramita and Johnson (2003) who trained a noun classifier on an earlier version of WordNet which was then tested on a later release.

The curvature clustering contains 242 nouns which appear in WordNet-1.7.1 but not yet in the earlier WordNet-1.6 release. This constitutes our test set of "novel" words. We find that the curvature clustering algorithm assigns each of these novel nouns to exactly one cluster. In fact, the majority of these words has exactly one sense in WordNet-1.7.1. The following table shows the frequency distribution of the number of WordNet-1.7.1 senses of these test words. The most polysemous test word according to WordNet-1.7.1 is "Robinson" which is listed with 7 senses, each of which refers to a different personality.

Table 5.8 shows some of the test words and the cluster they were assigned to. Non-core cluster members (words which weren't covered by the initial hard clustering, but joined the cluster in the final soft clustering step) are shown in the lighter gray font.

# of WN-1.7.1 senses	1	2	3	7
# of test words	221	18	2	1

Table 5.7: Distribution of the test words' polysemy counts in WordNet-1.7.1.

Test word	**Cluster**
vldl	hdl ldl triglyceride cholesterol lipoprotein
nephropathy	neutropenia splenomegaly thrombocytopenia failure function anaemia jaundice
gastroenterologist	radiologist oculist doctor surgeon
bruegel	durer breughel holbein veronese rembrandt rubens titian
dysgraphia	dyslexia agraphia spelling reading
coursework	doctorate dissertation assignment assessment work essay examination exam
etruscan	phoenician scythian gaul greek
trackball	joystick scsi adaptor disk modem mouse interface
carolina	arkansas wyoming kentucky oklahoma california north mexico georgia colorado virginia texas pennsylvania louisiana kansas tennessee missouri nebraska
tchaikovsky	berlioz dvorak mendelssohn symphony mozart elgar beethoven wagner
pledgee	grantee mortgagee mortgagor assignee lessee purchaser lessor court landlord interest surety father tenant person society buyer
bikers	camper backpacker cyclist walker
kandinsky	grosz kirchner klee schmidt marc corinth
biosphere	hydrosphere earth lithosphere atmosphere
kosovo	montenegro slovenia karelia kirgizia estonia uzbekistan byelorussia belorussia slovakia kirghizia byelarus moldavia rumania kirgizstan tajikistan dalmatia turkmenia adriatic moldova turkmenistan ...
halon	chlorofluorocarbon cfc dioxide chemical
rompers	slacks pullover sneaker underpants jodhpur fatigues slipover anorak trousers shirt pair clothes red sweater waistcoat jean boot trainer balaclava dress jacket marks shorts skirt knickers hair sock cardigan jumpers shoe slipper blouse
tagliatelle	kohlrabi courgette pasties broccoli flan radish quiche omelette caramel kale nob macaroon fruitcake cauliflower meatball noodle dish sausage turkey lettuce pepper green dessert sprout pea ...
hypnotherapy	autosuggestion psychotherapy meditation therapy counselling analysis
sikhism	vaisnavism jainism hinduism buddhism
physiotherapist	dietitian dietician nutritionist psychologist expert doctor representative

Table 5.8: Examples of test words and the clusters they appear in.

We now evaluate to which extent the meanings predicted by the curvature clustering algorithm coincide with the senses of the test words in WordNet-1.7.1. We do this by mapping each of the test words into the WordNet-1.7.1 noun hierarchy using its fellow cluster affiliates and the class labeling algorithm described in (Widdows, 2003). The class labeling algorithm returns the WordNet synset which best subsumes the test word's fellow cluster members. We chose to give more weight (a factor of 1.5) to core cluster members as they tend to describe a cluster's meaning more reliably.

The label is chosen so that it subsumes as many of the cluster members as closely as possible. The sense of a test word is correctly predicted by the clustering if it is directly subsumed by the class label. Some parts of the WordNet noun hierarchy go down into extreme and arbitrary detail, and even if the sense label computed is not an immediate hypernym of the test word's true sense, it may still constitute a sensible label. For example, the class labeling algorithm tells us that *Heisenberg* is a *physicist*. The *physicist* synset is two and not just one level above *Heisenberg* in the hypernymy tree, but, nevertheless, pinpoints the true meaning of *Heisenberg*, who belongs to the *nuclear physicist* synset, quite accurately.

In some cases, we have to walk both up and down in the hierarchy to trace a test word to its label. Figure 5.11 shows some of the observed paths between test word (red) and label (blue). The paths have different lengths and consist of a different number of up and down steps. All four labelings seem quite reasonable though. The fact that the computed sense label doesn't subsume the true sense of the test word in the latter two examples is to blame on the structural inconsistency of WordNet: According to WordNet, a *gastroenterologist* isn't a *medical specialist*, but all other kinds of physicians, e.g. *neurologists*, *psychiatrists*, *rheumatologists*, *orthopedists*, etc., are. Similarly, it's not immediately obvious why a *therapist* is not a kind of *specialist*.

In general, however, we expect a good sense label to be separated from the test word by only a few steps in the WordNet hypernymy tree. The farther the label is above the test word, the less telling it is. Down steps, however, are even less desirable. In particular, the path should involve less down than up steps as the semantic category of a word can't be more specific than the word itself. In summary, good labelings are characterized by short paths which involve few down steps.

Figure 5.12 shows the distribution of the number of up and down steps in the shortest paths between test word and label. Each cell (i, j) in the grid contains the number of test words which are separated from their label by i up and j down steps. The proportion of test words falling into a cell is also indicated by the degree of gray shading. Cell $(1, 0)$ contains the test words which are directly subsumed by the label they were assigned. For these 87 test words (36%) the meaning discovered by the curvature clustering coincides with a sense listed in WordNet (this constitutes the optimal case). Another 25 test words (10%), are subsumed at a distance of 2 by their label, which means the discovered sense is one WordNet level more general than the "true" WordNet sense. There is one instance where the assigned label and the test word are the same WordNet synset (cell $(0, 0)$). The test word in this case is *bridges* which, in its singular form was already present in WordNet-1.6, but which entered WordNet-1.7.1 as also being the name of Harry Bridges,

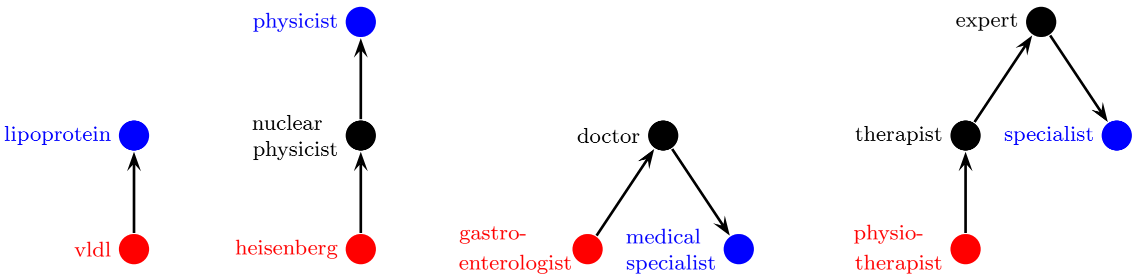


Figure 5.11: Some examples of paths linking a test word to its assigned label in the WordNet noun hierarchy.

a United States labor leader. The curvature clustering assigns *bridges* to the same cluster as *footbridge, underpass, flyover, wall, bridge, line*, which contains also the singular form *bridge*, and from which it infers the correct sense of the common noun *bridges* rather than the meaning which is associated with the proper noun *Bridges*. As the shading shows, the test words are concentrated largely in the lower-left cells which correspond to short paths. Furthermore, there is only a relatively small number of test words for which the assigned sense label is more specific than the test word. These are found in the cells which lie above the stepped line.

We manually inspected the clusters and their labels and found that in most of the cases of far apart word-label pairs, we did in fact discover a correct sense of the test word. Many of the discrepancies between word and label can be attributed to structural inconsistencies and missing senses in the WordNet lexicon. More precisely, we noticed the following causes of long test word-label paths:

1. The sense discovered by the curvature clustering does not appear in WordNet, but is nevertheless a valid sense of the test word.

2. WordNet is missing some of the senses of the other cluster members, resulting in an inappropriate cluster label.

3. The long path is due to extremely subtle sense distinctions in WordNet. This is particularly the case with the vocabularies of medicine and the natural sciences which have an exceedingly fine granularity level in WordNet.

4. The sense distinctions made in WordNet are inconsistent.

Some miscategorizations, however, are due to mistakes on our side, caused by

5. Incorrect treatment of compound nouns during data preparation.

6. Failure to recognize a correct sense of the word.

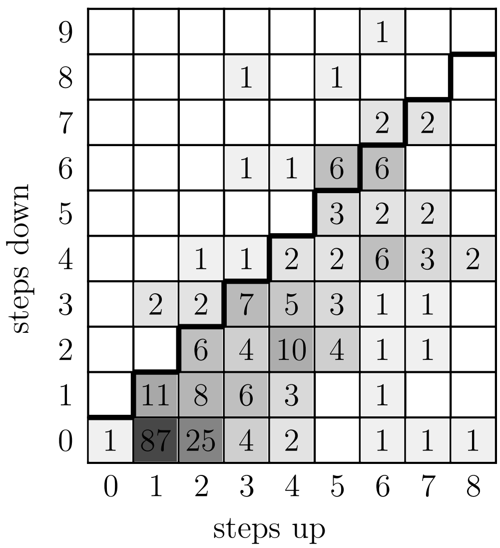


Figure 5.12: Frequency histogram of the number of up and down steps along the shortest path from a testword to its label.

Type 1: Examples of correctly inferred word senses which are not captured in WordNet are the “travelers” sense of *bikers*, the “device” sense of *traverser*, the “town” sense of *Harlow* and the “author” sense of *Spearpoint*. Each of these test words is shown in Table 5.9 together with its fellow cluster members, the cluster label and its WordNet sense.

Test word	**Fellow Cluster Members**	**Cluster label**	**WN sense label**
bikers	camper backpacker cyclist walker	traveler	youth subculture
traverser	capstan crane	lifting device	mover
harlow	harrisburg haifa	port	actress
spearpoint	lyricist singer	author	point

Table 5.9: Examples of correctly learned word senses absent in WordNet.

The WordNet gloss for the noun *bikers* is

> *Originally a British youth subculture that evolved out of the teddy boys in the 1960s; wore black leather jackets and jeans and boots; had greased hair and rode motorcycles and listened to rock'n'roll; were largely unskilled manual laborers.*

WordNet is missing the much more general and very common sense of *biker* which is simply a person riding a motorcycle. The cluster label *traveler* comes quite close to this meaning of *bikers*. Similarly, *traverser* is encoded in WordNet as being a *person* ("someone who moves or passes across"). However, *traverser* also has a technical meaning, referring to a transfer table, which is a laterally movable piece of track which is used to move a train from one track to another. Naturally, WordNet can't have full coverage of named entities. The clustering reveals that *harlow* was not only the name of a famous actress but is also the name of a town in Essex, England, and that besides being literally interpretable, *spearpoint* also refers to the songwriter with the same name.

Type 2: Some clusters closely correspond to a correct sense of the test word. This sense, which is shared with the other cluster members, is also a WordNet sense of the test word, but it is missing in some of the other cluster members' WordNet sense profiles. As a result, the class labeling algorithm produces an inadequate cluster label. An example of a test word which was wrongly labeled for this reason is the noun *bulimia*. It occurs in a cluster together with the other core member *overeating* and the non-core members *anorexia* and *disorder*. *Overeating*, since it is a core member, has a bigger influence on the cluster label than the other two cluster members. WordNet lists only a single sense for *overeating*, namely that of "mortal sin". It does not include the "eating disorder" sense of *overeating*, and as a consequence, *bulimia* receives the incorrect label *mortal sin*.

We observe the same type of mislabeling for the noun *palau*. It occurs in the same cluster as *yap* and *manus* which belong to the core of the cluster, and *island*, which is a non-core member. WordNet is ignorant of the "island" sense of both *yap* and *manus*. But, as it happens, the WordNet senses of *yap* ("mouth") and *manus* ("hand") are very closely related, both being body parts. For this reason, the labeling algorithm infers that *palau* is a body part, too, where in fact it is a group of islands in the Pacific Ocean.

Type 3: For some of the test words, the discovered sense and the WordNet sense are very close in meaning but nevertheless distant in WordNet. The granularity of sense distinctions varies a lot and arbitrarily across topic areas. Whereas certain branches of the hypernymy tree make rather coarse-grained sense distinctions, others go into great detail. The length of a path separating two WordNet synsets reflects not only their semantic relatedness but also to some extent the level of detail of a domain. This variation in sense granularity often causes the shortest path connecting a test word and its label to be extremely long and is the reason why many of the discovered senses are given low confidence.

The test word *camcorder*, for example, has the fellow cluster members *vcr*, *tape*, *player* and *machine* from which the class labeling algorithm computes the label "tape recorder". The WordNet sense of *camcorder* is "television camera" and is seven links (4 steps up and 3 steps down) away from the computed sense label "tape recorder". In comparison, *Christmas tree* and *highway* are only five links (3 steps up, 2 down) apart from each other (cf. Figure 5.13).

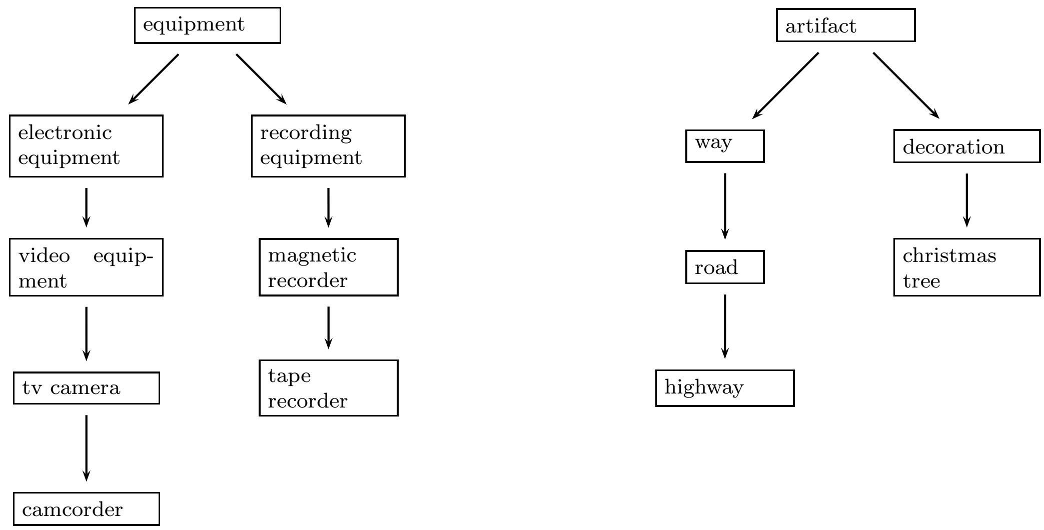


Figure 5.13: *Camcorder* and *tape recorder* are more distant in WordNet than *highway* and *christmas tree*.

Type 4: Many of the long paths between a test word and its label are due to such inconsistencies in the way WordNet assigns senses to words. The noun *scythian*, for example, forms a cluster together with other ethnic groups, *phoenician*, *etruscan*, *gaul*, *greek*, and the cluster receives the label "European". According to WordNet, a Scythian is a "nomad" who in turn is a "wanderer" who is a "traveler" and so on. There is no sense of *Scythian* in WordNet which specifies the geographical origin of these people. This is however the case for other ethnic groups. According to WordNet, a *Bedouin* is an "Arab", a *Tartar* is a "Mongol", and a *Maya* is a "North American Indian". It seems that WordNet, in order to be consistent, should include a "Eurasian" sense for *Scythian*, which is close to the computed sense label "European".

In many cases, the name of an ethnic group is also the name of their language. WordNet is very inconsistent with respect to this regular People/Language polysemy. For example, WordNet lists the "language" sense of *Thracian* and *Illyrian*, but is missing their "person" sense. As a result, *Dardanian* which forms a cluster together with *Thracian* and *Illyrian* receives the false label "Indo-European language" where in fact *Dardanian* refers only to a person ("Trojan: a native of ancient Troy").

Type 5: In building the word graph, we decided to strip off a noun's adjectival and nominal modifiers to increase the graph's community structure (degree of clustering). However, this preprocessing step treats a compound noun as if it were a noun phrase consisting of the head preceded by a set of nominal modifiers. Many compound nouns, such as *South Carolina* or *Upper Volta*, for example, which do not have a head, mistakenly become *carolina* and *volta* respectively. The meanings of these corrupted compounds are very different from the original ones. *Carolina*, for example, forms a cluster together with

> *arkansas wyoming kentucky oklahoma california north mexico georgia colorado virginia texas pennsylvania louisiana kansas tennessee missouri nebraska*

The labeling algorithm assigns this cluster the label *American State*. Whereas North and South Carolina are indeed American States, *Carolina* isn't. Its single WordNet sense is "geographical region" which is not only quite distant from but even more general than the computed label "American State". Similarly, *volta* appears in a cluster composed of many, mostly African, countries:

> *togo gambia cameroun qatar senegal antilles nauru lesotho berzelius niger malawi tahiti caliphate burundi tanganyika myanmar tobago cameroon micronesia benin kiribati surinam guyana gabon vanuatu bhutan congo gatt botswana emirate maldives mauritania tuvalu ...*

Naturally, this cluster receives the label *African country*. This accurately describes the meaning of *Upper Volta* (the former name of Burkina Faso). *Volta*, on the other hand, has only one meaning in WordNet which refers to the physicist Alessandro Volta and which is very different from our discovered meaning of *African country*.

Type 6: In some cases, the sense discovered simply isn't a valid sense of the test word. This happens, for example, with the test word *gorbachev* which is mistakenly assigned to a cluster of countries. Another example of a wrongly labeled test word is *hrt* (hormone replacement therapy) which occurs in a cluster with topically rather than paradigmatically related words:

Test word	**Fellow Cluster Members**	**Cluster label**	**WN sense label**
gorbachev	montenegro slovenia karelia kirgizia estonia uzbekistan byelorussia belorussia slovakia kirghizia byelarus ...	European country	statesman
hrt	puberty menarche menopause pregnancy time age	time period	therapy

Chapter 6

From clustering words to clustering word associations

6.1 Intuition

Until now, we have tried to extract semantic classes in the word graph by clustering the nodes representing words into groups of similar words based on shared links. As we have seen, many of the nodes are an amalgamation of multiple meanings. The links, which correspond to pairs of related words, on the other hand involve much more contextual information than the nodes, which makes them far less ambiguous.

In this chapter, we will investigate an alternative approach to inducing word meaning from the word graph which treats the *links* (word relationships) rather than the *nodes* (isolated words) as the objects to be clustered.

Since most of the links address a particular sense of the words they connect (cf. Yarowsky (1993)'s one-sense-per-collocation principle), we naturally avoid having to deal with fuzzy cluster boundaries caused by ambiguity. At the same time, a word can be present in more than one cluster as its relationships can be distributed across different clusters.

This is very much in the style of Schütze (1998) who induces word senses by grouping together the occurrences of a word into clusters which can be seen as representing its senses. Similarly, we will interpret the different clusters in which a word's relationships occur as characterizing the different meanings of the word.

6.2 Link graph transformation

To shift the focus from nodes to links, we transform the links of the original word graph $\mathcal{W}$ into nodes of a new graph $L(\mathcal{W})$. This allows us to reduce the problem of clustering the edges of $\mathcal{W}$ to the conjugate (and already familiar) problem of clustering the nodes of the new graph $L(\mathcal{W})$. The conjugate graph $L(\mathcal{W})$ is constructed by

1. Creating a vertex v_e for each edge e in the original graph $\mathcal{W}$.

2. Putting a link between two (new) vertices v_e and v_f of $L(\mathcal{W})$ if the corresponding edges e and f belong to a common triangle in $\mathcal{W}$.

We will refer to this new transformed graph $L(\mathcal{W})$ as the *link graph* associated with $\mathcal{W}$. The link graph transformation is illustrated in Figure 6.1.

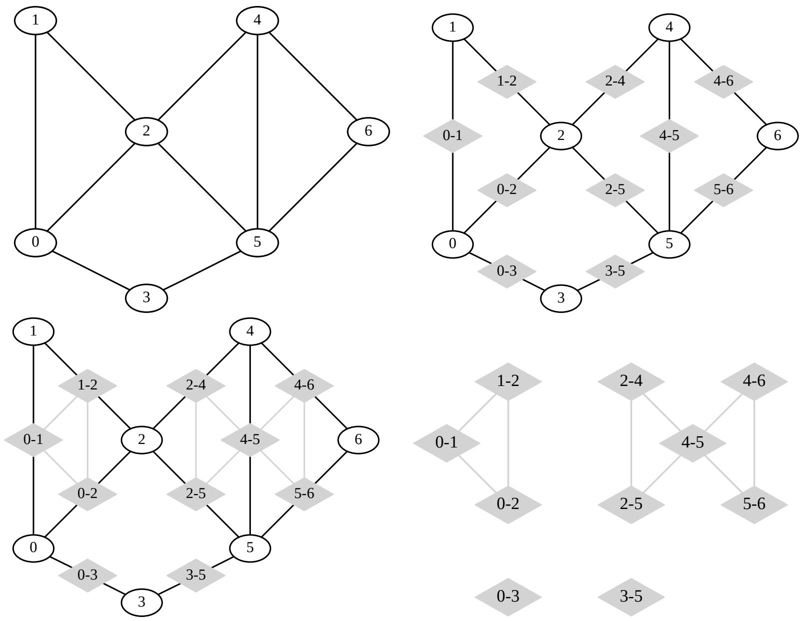


Figure 6.1: From $\mathcal{W}$ to $L(\mathcal{W})$. The original graph, new nodes v_e, new links, the graph $L(\mathcal{W})$.

The idea behind joining the newly created vertices according to the rather restrictive Rule 2 is that we only want to join semantically compatible word associations. Two links in $\mathcal{W}$ which cooccur in a triangle correspond to word associations which do not only share a word but, due to transitivity, share the same sense of that word. Similarly, two adjacent links which do not form sides of the same triangle are likely to refer to different senses of the word (node) they have in common.

This idea is illustrated in Figure 6.2 which shows two triangles in the word graph which meet at the node representing the word form *squash*. Links incident to the *squash* node which appear in the same triangle (i.e. have the same color) refer to the same sense of *squash*: The red links address the "vegetable", the green the "sport" sense of *squash*. In contrast, links which belong to different triangles (i.e. are differently colored) refer to different senses of *squash*.

Figures 6.3 and 6.4 show the neighborhood of the ambiguous word *organ* before and after application of the link graph transformation. In the original word graph, the different meanings of *organ* are not immediately identifiable since some nodes in the graph (*movement* and *body* in addition to *organ*) are highly ambiguous and cause fuzzy sense boundaries. The nodes in the transformed graph, on the other hand, have a sharp-cut

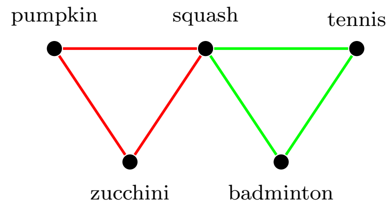


Figure 6.2: Two triangles involving the word form *squash*. Transitivity of meaning holds only within, not across, triangles.

meaning and refer to either the "body part" or the "musical instrument" sense of *organ*. Transition to the link graph naturally divides the initially fuzzy neighborhood of *organ* into semantically consistent pieces corresponding to its different senses.

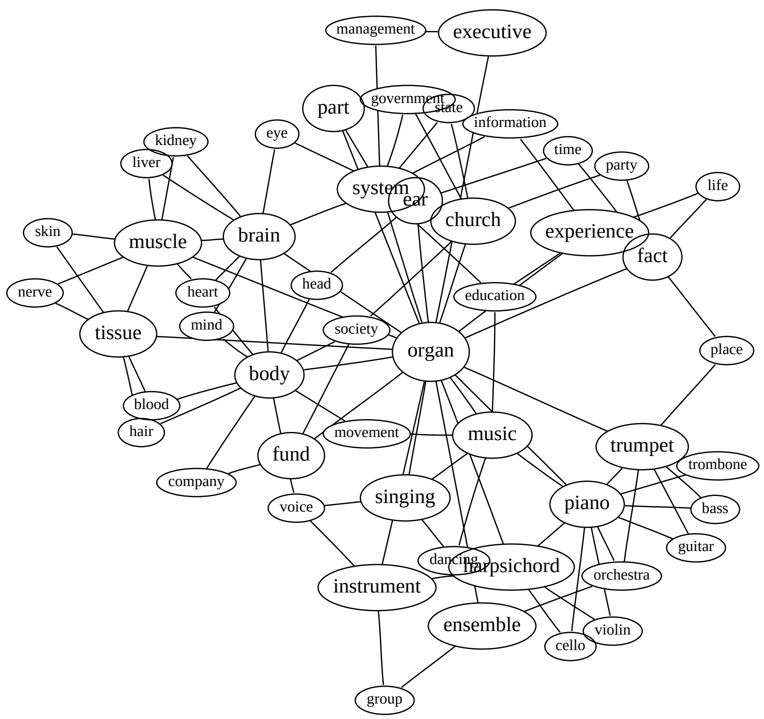


Figure 6.3: Local word graph around the word form *organ* based on the original word graph $\mathcal{W}$. Unrelated areas of meaning (body parts, musical instruments) are connected to one another.

6.3 Some properties of the link graph

The number of nodes in the transformed graph $L(\mathcal{W})$ is obviously equal to the number of links in $\mathcal{W}$. This means that the link-transformed graph is considerably larger than the

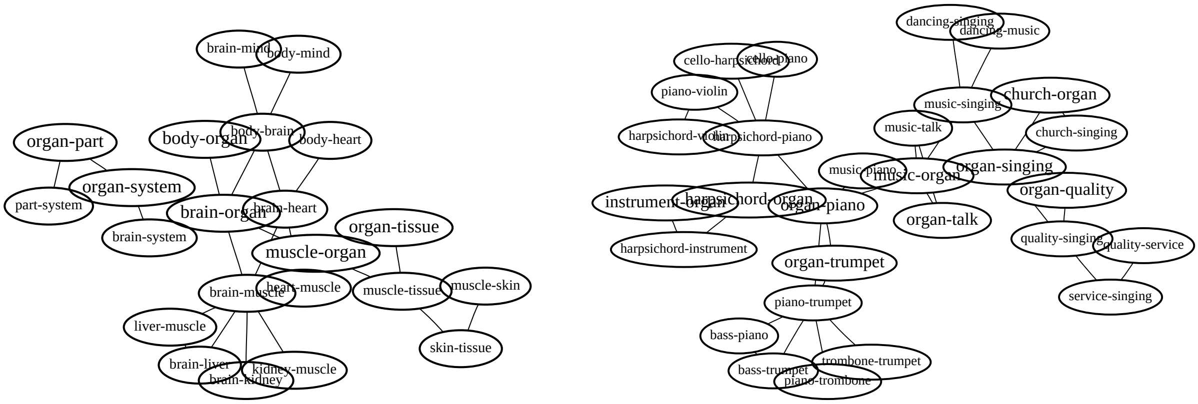


Figure 6.4: *Organ*'s associated link graph. Contexts belonging to the "body part" and "musical instrument" sense are neatly separated.

original graph. Each triangle in $\mathcal{W}$ gives rise to three unique links in $L(\mathcal{W})$ (cf. Figure 6.1). The number of links in $L(\mathcal{W})$ is thus three times the number of triangles in $\mathcal{W}$.

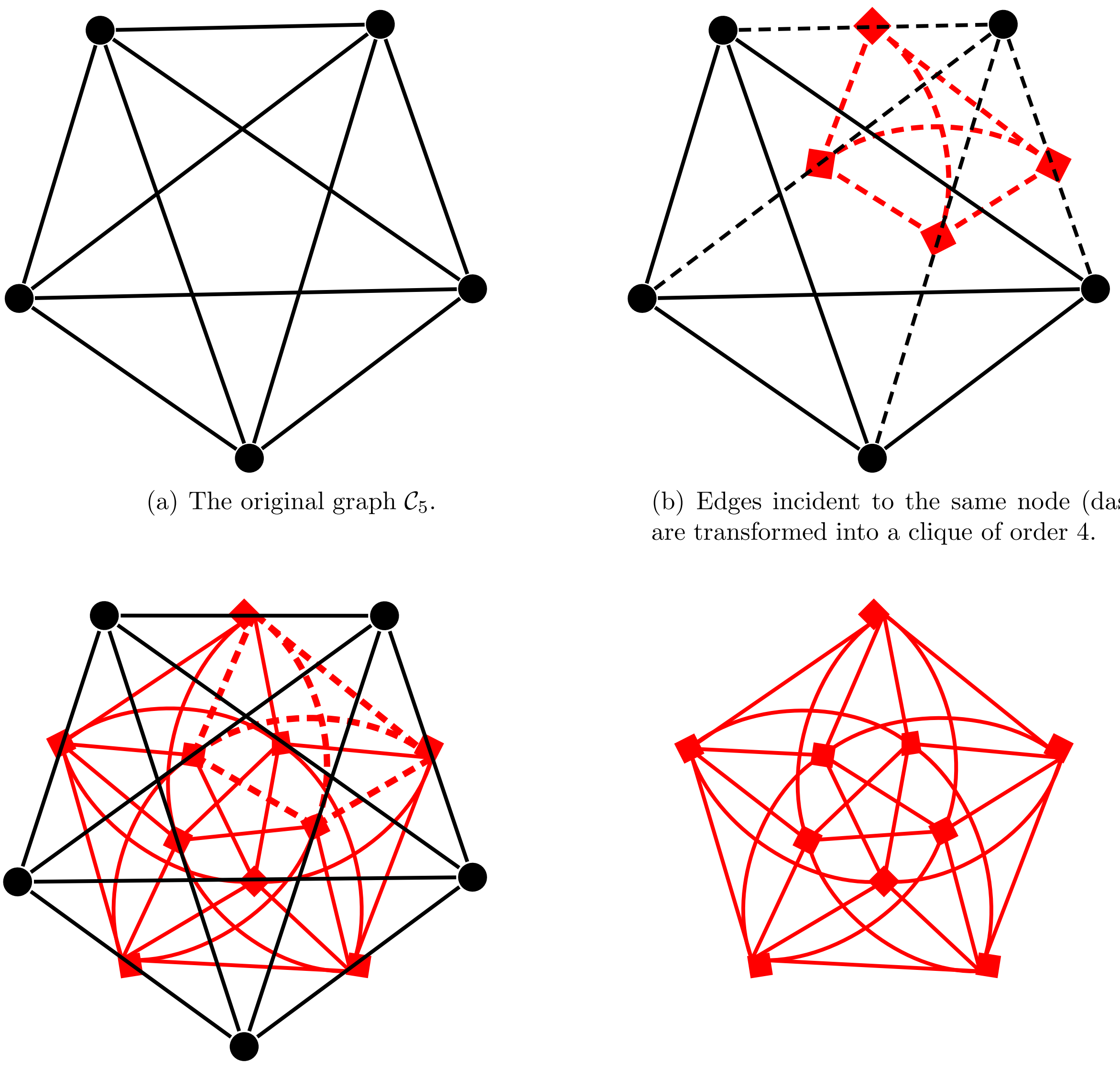
To get an idea of how triangles are arranged in the link graph, let us examine how a clique $\mathcal{C}_k$ (complete graph containing k nodes, cf. Section 2.1) behaves under the link graph transformation. The process is illustrated in Figure 6.5. The $k-1$ edges incident to a node u in $\mathcal{C}_k$ are transformed into a clique of order $k-1$ since any two of them belong to a common triangle. Any edge $e=(u,v)$ in $\mathcal{C}_k$ is therefore transformed into a node v_e which is a member of two $(k-1)$-cliques, one associated with each of its end points, and thus has a degree of $2(k-2)$.

Let us have a look at Figure 6.6. The neighbors of v_e in $L(\mathcal{C}_k)$ correspond to those links in $\mathcal{C}_k$ which are incident to either u (the red links) or v (the blue links). There are $\binom{k-2}{2}$ links among v_e's red neighbors and $\binom{k-2}{2}$ links among its blue neighbors since both sets of equally colored links are transformed into a clique of order $k-1$. In addition, for each red link incident to u, there is exactly one blue link incident to v which has the same endpoint. Therefore, there is an additional $k-2$ links between the red and blue neighbors of v_e in $L(\mathcal{C}_k)$, and the new node v_e in $L(\mathcal{C}_k)$ has a curvature of

$$
\begin{aligned}
curv(v_e) &= \frac{2\cdot\binom{k-2}{2}+(k-2)}{\binom{2\cdot(k-2)}{2}} \\
&= \frac{(k-2)\cdot(k-3)+(k-2)}{2(k-2)\cdot(2(k-2)-1)/2} \\
&= \frac{k-2}{2k-5} \quad ,
\end{aligned}
$$

which is $\geq \frac{k-2}{2k-4}=0.5$ for $k\geq 3$. As k becomes larger, $curv(v_e)$ approaches this lower bound.

We conclude that the image of a highly clustered area in $\mathcal{W}$ also shows a considerable degree of clustering in $L(\mathcal{W})$. It seems therefore that the community structure present in the original graph $\mathcal{W}$ is largely preserved under the link graph transformation.



(a) The original graph $\mathcal{C}_5$.

(b) Edges incident to the same node (dashed) are transformed into a clique of order 4.

(c) Repeat procedure for each node in $\mathcal{C}_5$.

(d) The transformed graph $L(\mathcal{C}_5)$.

Figure 6.5: The image of a clique under the link graph transformation. The edges incident to a node are transformed into a clique which is one order of magnitude smaller than the original clique.

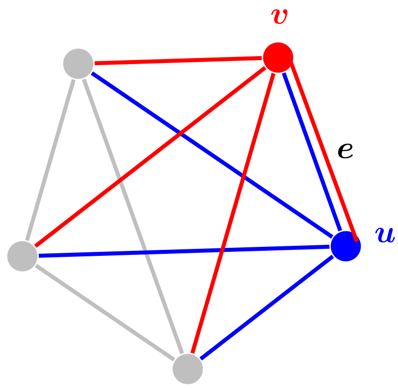


Figure 6.6: Links incident to the same endpoint of an edge e (i.e. equally colored links) are linked to one another in $L(\mathcal{C}_k)$. Similarly, links incident to opposite endpoints of e (i.e. differently colored links) are linked in $L(\mathcal{C}_k)$ if they share their other endpoint.

6.4 Clustering the link graph

In order to keep the link graph a manageable size, we start off with the subgraph $\mathcal{W}_{\mathrm{wn}}$ of the word graph $\mathcal{W}$ which is formed by the nouns known to WordNet. Furthermore, we drop weak links. We consider a link in $\mathcal{W}_{\mathrm{wn}}$ to be weak if its log-likelihood weight (cf. Section 3.5) falls below a certain threshold λ.

The parameter λ regulates not only the manageability of the link graph, but also affects the coverage and purity of the link graph. High values of λ result in lower coverage of words and word meanings, but in higher significance of the links.

We use $\mathcal{W}_\lambda$ to denote the subgraph of $\mathcal{W}_{\mathrm{wn}}$ which consists of those links whose log-likelihood weight is greater than λ. Table 6.1 summarizes the basic statistical properties of $\mathcal{W}_\lambda$ and its associated link graph $L(\mathcal{W}_\lambda)$ for different values of λ.

The nodes of the transformed graph are highly contextual units of meaning and thus far less ambiguous than the original nodes. We can therefore use a hard clustering algorithm, such as Markov Clustering (cf. Section 4.2), to divide the set of new nodes into (non-overlapping) classes of similar word relationships. These clusters of *relationships* can be translated back into clusters of *words* which consist of the words the relationships are composed of. The back transition from word relationships to words naturally yields a soft clustering of the original word graph. We interpret the different clusters a word appears in as representing its different senses.

6.5 Evaluation

We apply Markov Clustering to each of the transformed graphs $L(\mathcal{W}_\lambda)$, with λ ranging from 3 to 12. High values of λ imply lower coverage, but also increased significance of the links. In the following, we evaluate the clusterings associated with different values of the control parameter λ and investigate how λ interacts with the quality of the clustering.

λ	$\mathcal{W}_\lambda$			$L(\mathcal{W}_\lambda)$		
	#nodes	#edges	#triangles	#nodes	#edges	#triangles
3	19,916	289,215	635,941	289,215	1,907,823	4,864,389
4	19,611	227,829	422,436	227,829	1,267,308	2,708,276
5	19,182	180,966	297,521	180,966	892,563	1,643,345
6	18,623	145,383	217,300	145,383	651,900	1,042,900
7	17,885	118,967	165,407	118,967	496,221	720,323
8	16,957	98,886	130,379	98,886	391,137	529,919
9	15,949	83,672	105,484	83,672	316,452	405,356
10	14,870	71,735	87,613	71,735	262,839	323,121
11	13,743	62,202	73,564	62,202	220,692	261,724
12	12,716	54,681	62,780	54,681	188,340	216,392

$\cdot 3$

Table 6.1: The number of nodes, links and triangles before and after the link graph transformation for different log-likelihood thresholds λ.

In order to assess the quality of a clustering we use the same evaluation methodology as for the curvature clustering (cf. Section 5.4). There, a sense of a given word form was considered correctly recognized if the word form occurred in a cluster whose similarity to the sense exceeded a certain threshold θ. The control parameter θ determines the rigor of the evaluation. The following data is based on a similarity threshold of $\theta = 0.25$ (as used by Pantel and Lin (2002)).

Figure 6.7 shows precision and recall (black dots) for different clusterings associated with different log-likelihood thresholds λ. Since the coverage of the clustering changes with λ, we compute a clusterings *global* recall. This means, we take the set of all WordNet senses of the words contained in the largest graph (which is the one associated with the smallest value of λ, i.e. $\mathcal{W}_3$) as the set of word meanings to be discovered.

The precision-recall curve is monotonically decreasing. Raising the log-likelihood threshold λ improves precision but decreases recall, which means that word meanings are extracted more reliably but less completely. Removing more and more low-weight edges reduces not only noise but also removes more and more semantic information. The green curve shows the precision and recall values obtained by applying Markov Clustering directly to the untransformed word graph. Whereas precision of the baseline is considerably higher, its recall is much lower.

Figure 6.8 shows the corresponding F-scores (see Section 5.4.1 for the definition), which measure the trade-off between precision and recall. The F-measure increases with λ until it reaches its maximum at $\lambda = 7$ and then decreases. This means that for $\lambda \geq 8$ the increase in the quality of the links is outweighed by the loss of valuable semantic information. As

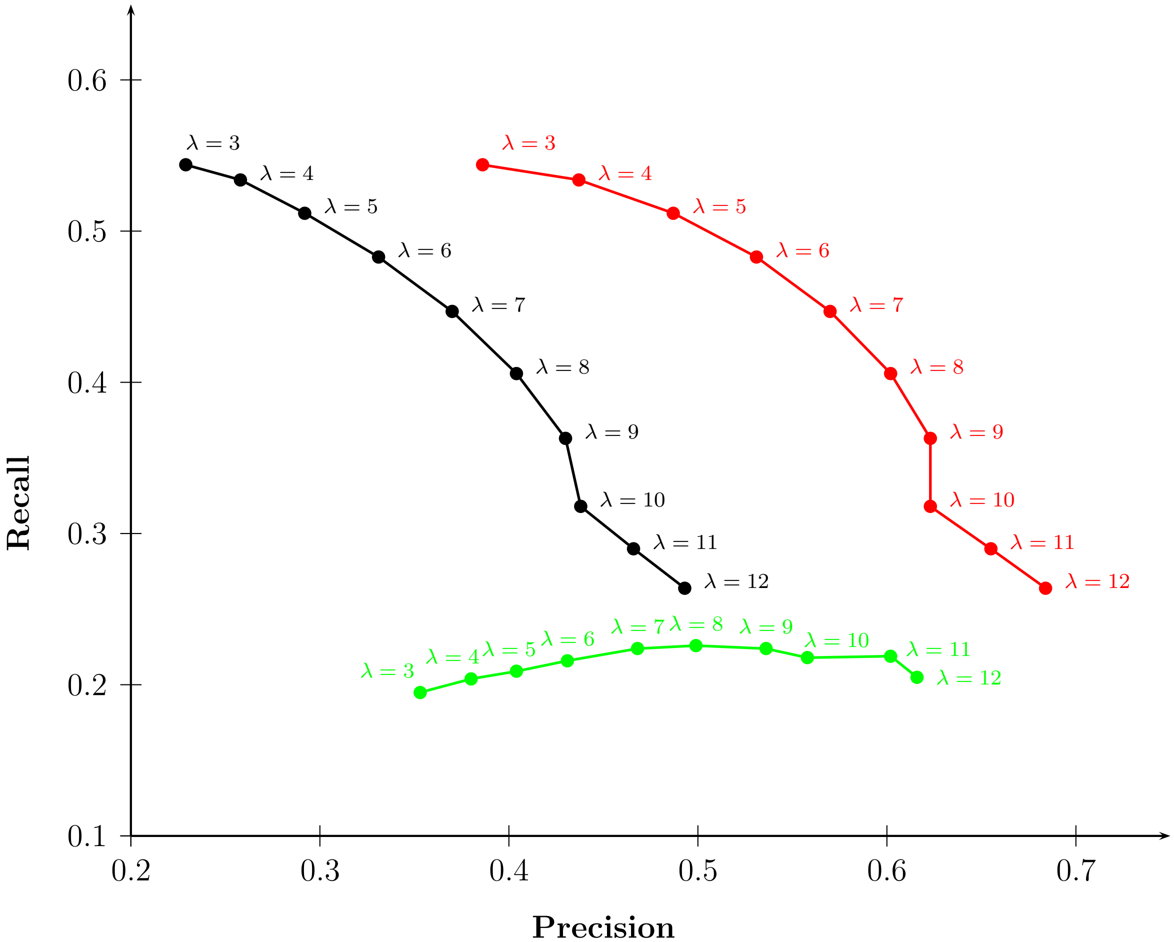


Figure 6.7: The black dots correspond to precision and (global) recall of the clusterings associated with different log-likelihood thresholds λ. The red points are obtained if the clustering is not penalized for multiply discovered senses. The green points correspond to the baseline which is Markov Clustering of the (untransformed) word graph.

can be seen, the corresponding baseline F-scores (green dots) are much lower.

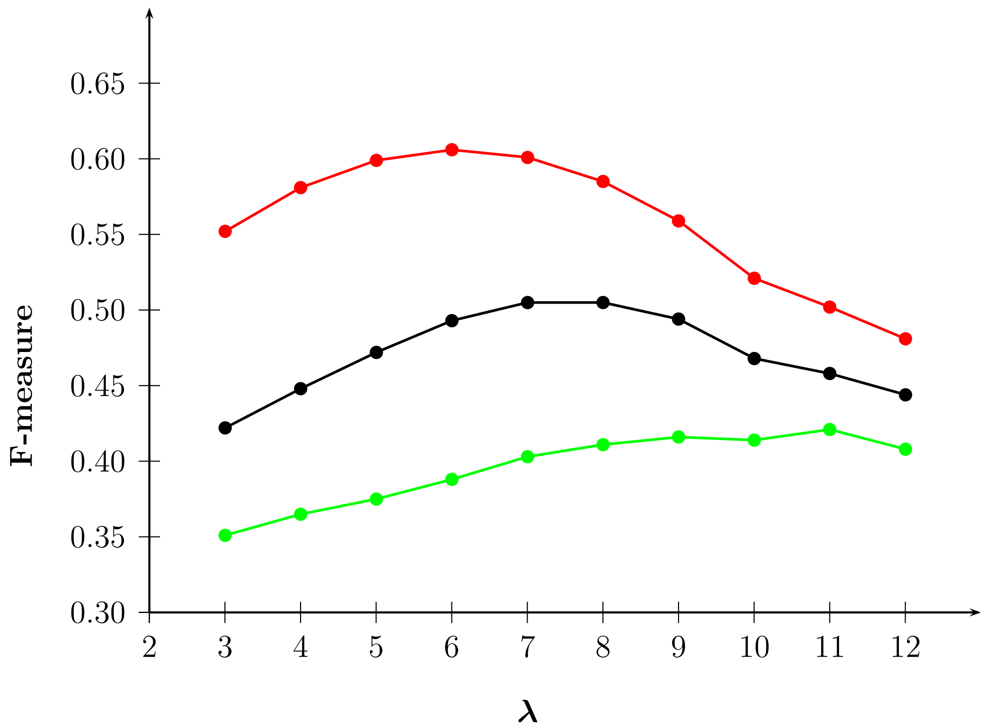


Figure 6.8: Strict (black) and relaxed (red) F-measure of the clusterings associated with different log-likelihood thresholds λ.

The low precision of the clustering ($\leq 50\%$, even for $\lambda = 12$) is mainly due to repeatedly discovered senses of which only one is counted as a correctly discovered sense while all others are considered to be wrongly detected senses.

Due to our strict rule for connecting the new nodes v_e in $L(\mathcal{W})$, the link graph is quite sparse. Many adjacent links which do not belong to a common triangle may still correspond to similar word associations. The third missing edge which would complete the triangle may simply not have been observed in the text. Due to sparse connectivity, Markov Clustering divides the link graph into a large number of rather small pieces. This means that a word's relationships are distributed across many different clusters, many of which do in fact represent the same sense.

If we relax the definition of precision (cf. Section 5.4.1) so that each correctly discovered sense (whether discovered before or not) counts as such, then precision increases considerably (red dots in Figure 6.7) and even exceeds the precision of the baseline clustering.

We think that a fine-grained clustering is a lot more useful than a coarse-grained clustering which lumps different senses together. Still, we would like to be able to tell the more meaningful of a word's cluster affiliations from the less informative ones. Moreover, we would like to identify those clusters which represent *different* senses of a word.

6.6 Pairs of contrasting clusters

The clusters are composed of conjugate nodes v_e corresponding to word pairs. A word may participate in more than one conjugate node within a cluster. The more conjugate nodes it participates in, the stronger is its affiliation to the cluster, and the more reliably the cluster characterizes a meaning of the word.

We define a word w's *affinity* to a cluster c as follows. Suppose the cluster c is composed of n words. We sort these words according to how many conjugate nodes in c they participate in, and we use $\mathrm{rank}_c(w)$ to denote the rank of w in this sorted list of words ($1 \leq \mathrm{rank}_c(w) \leq n$). The *affinity* of w to c is given by

$$aff_c(w) = \frac{n - rank_c(w)}{n}, \tag{6.1}$$

which assumes a value between 0 and 1. The last word in the sorted list of cluster members receives an affinity score of 0, the first one receives a score of $(n-1)/n$. This ensures that words appearing early in *short* lists are not favored too much over words appearing early in *long* lists.

The word *humour*, for example, is a member of the two clusters shown in Figure 6.9. Each cluster member is followed by the number of conjugate nodes it participates in (Freq) and its resulting rank in the frequency-sorted list of cluster members (Rank). Whereas *humour* ranks first in the left cluster, it is among the lowest ranking members of the right cluster (Rank=9.5). The two clusters consist of 13 and 11 members, and Formula 6.1 yields the following cluster affinities

$$\begin{aligned} aff_{c_{593}}(humour) &= (13-1)/13 = 0.923 \\ aff_{c_{802}}(humour) &= (11-9.5)/11 = 0.136. \end{aligned}$$

The above defined affinity score may assign high scores to several clusters representing the *same*, prevalent sense of a word. In the following, we describe a way of identifying pairs of clusters representing *different* senses of a target word. We call such pairs of clusters *contrasting pairs*. Given a word w, we measure the *contrast* between two of w's clusters as follows:

$$contr_w(c_1, c_2) = \frac{aff_{c_1}(w) * aff_{c_2}(w)}{sim(c1, c2)}, \tag{6.2}$$

where $sim(c1, c2)$ is a measure of the similarity of two clusters. We measure the similarity between two clusters c_1 and c_2 as the similarity of their vector representations in a vector space model built using Latent Semantic Analysis (cf. Section 9.2.4). Contrast is a trade-off between the dissimilarity of two clusters and their relevance to the target word. Two clusters c_1 and c_2 which show high contrast are thus good indicators of different senses of w.

As an example, let us consider the word *delivery*. The table on the left hand side of Figure 6.2 lists three of its clusters and its affinity to them computed according to Formula 6.1. The table on the right hand side shows the inter-cluster similarities $sim(c_1, c_2)$ (black)

Cluster 593		
Word	Freq	Rank
humour	10	1
irony	4	3
wit	4	3
fantasy	4	3
sarcasm	3	6
satire	3	6
fun	3	6
fairytale	2	9
charm	2	9
pathos	2	9
pain	1	12
adventure	1	12
honesty	1	12

Cluster 802		
Word	Freq	Rank
gentleness	8	1
courtesy	4	3
sensitivity	4	3
patience	4	3
meekness	2	6
grace	2	6
kindness	2	6
humility	1	9.5
hospitality	1	9.5
humour	1	9.5
warmth	1	9.5

Figure 6.9: Two of the clusters the word *humour* occurs in, sorted by the number of relationships they are involved in (Freq). Whereas *humour* ranks first in Cluster 593, it ranks low (9.5) in Cluster 802.

as well as the resulting cluster contrasts (red). Using Formula 6.2, the contrast between Clusters 944 and 1426, for example, equates to

$$\begin{aligned} contr_{delivery}(c_{944}, c_{1033}) &= \frac{aff_{c933}(delivery) * aff_{c1033}(delivery)}{sim(c_{944}, c_{1033})} \\ &= \frac{0.64 * 0.90}{0.28} \\ &= 2.05. \end{aligned}$$

Note that the two least similar clusters (c_{944} and c_{1426} which have the lowest similarity of 0.14) are not the most contrasting ones. The affinity of *delivery* to cluster c_{1426} is considerably weaker than its affinity to cluster c_{1033}, which indicates that c_{1426} less reliably describes a correct meaning of *delivery*. Hence, the contrast measure assigns the pair (c_{944}, c_{1033}) a higher value.

We tested our approach on the set of ambiguous words which were used by Yarowsky (1995) in the evaluation of his word sense disambiguation system. These words and the senses given by Yarowsky are shown in Table 6.3. The results are presented in Table 6.4. Each row corresponds to a contrasting pair of clusters (Cluster 1 vs. Cluster 2). For each of the test words, the three top scoring (that is most contrasting) cluster pairs are shown. Only the twelve top most frequent members of a cluster are listed.

The two words missing are *poach* and *axes*. *Poach* is a verb and hence not covered by our model. The word *axes* does not occur in our model due to an oversight during preprocessing. We mistakenly didn't account for words which have more than one WordNet base form

No.	Cluster	Affinity
944	pregnancy labour *delivery* child lactation abuse service birth foetus marriage depression	0.64
1033	*delivery* ordering transfer payment quantity date quality sum service price	0.90
1426	purchaser auctioneer seller price *delivery* vendor investor manufacturer supplier	0.39

Cluster similarity / contrast

	c_{1033}	c_{1426}
c_{944}	0.28 2.05	0.14 1.74
c_{1033}		0.55 0.64

Table 6.2: Computing the contrasts between three of the clusters the word *delivery* appears in. The left table contains the clusters and affinity scores. The table on the right lists the cluster-cluster similarities (black). The contrast between two clusters is a trade-off between high word-to-cluster affinities and low cluster-cluster similarity (red values).

Word	Sense1	Sense2
bass	fish	music
tank	vehicle	container
space	volume	outer
crane	bird	machine
duty	tax	obligation
sake	benefit	drink
drug	medicine	narcotic
palm	tree	hand
motion	legal	physical
plant	living	factory
axes	grid	tools
poach	steal	boil

Table 6.3: Test words used by Yarowsky.

Test word	Cluster 1	Cluster 2
bass	ray codling flounder whiting pout dogfish dab eel bass conger pollack cod	bass guitar amplifier deck tuner player keyboard equaliser drummer rhythm combo headphone
	ray codling flounder whiting pout dogfish dab eel bass conger pollack cod	treble bass boost clef percussion tone
	ray codling flounder whiting pout dogfish dab eel bass conger pollack cod	soprano tenor alto bass mezzo cello baritone sax contralto conductor
tank	howitzer tank machine weapon mortar soldier carrier artillery gun	reservoir wells bore tank pipes user tunnel watercourse
	tank transporter carrier artillery troop	fish hood tank men heater cabinet time woman parent industry equipment sink
	howitzer tank machine weapon mortar soldier carrier artillery gun	oxygen helium hydrogen nitrogen methane tank monoxide sulfur dioxide cylinder chlorine deuterium
space	space facility comfort privacy enjoyment pleasantness warehouse resource seclusion service heating peace	space void gap form vacuum street chance air light sunlight
	space facility comfort privacy enjoyment pleasantness warehouse resource seclusion service heating peace	space comparison shape marker weight
	space facility comfort privacy enjoyment pleasantness warehouse resource seclusion service heating peace	world universe mankind man einstein physicist space scotsman language dimension mystic life
crane	crane truck digger excavator skid generator shed wheel trailer conveyor driver lorry	heron grebe curlew swan flamingo crane plover spoonbill wildfowl pelican duck waterfowl
	crane truck digger excavator skid generator shed wheel trailer conveyor driver lorry	swift shelley schopenhauer kafka schiller rousseau conrad johnson donizetti tasso crane barrie
	heron grebe curlew swan flamingo crane plover spoonbill wildfowl pelican duck waterfowl	capstan crane winch traverser
duty	piety patriotism loyalty duty love value authority cruelty kindness squalor obedience bigotry	tax taxis duty fee burden interest levy charge contribution cost benefit expense
	piety patriotism loyalty duty love value authority cruelty kindness squalor obedience bigotry	damage injunction account duty breach trespass plaintiff rule development congestion tort court
	piety patriotism loyalty duty love value authority cruelty kindness squalor obedience bigotry	custom excise vat immigration paye duty revenue department palette contribution commissioner squad
sake	fate jesus sake	competitiveness growth profits materialism success profitability acquisitiveness sake productivity investment
	honour glory flop challenge respectability sake pride excitement shame	competitiveness growth profits materialism success profitability acquisitiveness sake productivity investment
	fate jesus sake	honour glory flop challenge respectability sake pride excitement shame
drug	antibiotic drug hormone anaesthetic steroid painkiller food penicillin zidovudine immunization substance vaccine	drug gun terrorist launcher explosive allegation ira obesity firearm
	pill drug booze contraceptive antihistamine alcohol medicine cigarette	drug gun terrorist launcher explosive allegation ira obesity firearm
	drug treatment medicine ect woman development	drug gun terrorist launcher explosive allegation ira obesity firearm

Test word	Cluster 1	Cluster 2
palm	wrist palm fingertip finger toe knuckles grin thigh knee shoulder ear heel	palm banana papaya beach cassava mango pool
	wrist palm fingertip finger toe knuckles grin thigh knee shoulder ear heel	portico tower building palm stone porch block hollyhock entrance steps cloister apse
	portico tower building palm stone porch block hollyhock entrance steps cloister apse	palm banana papaya beach cassava mango pool
motion	amendment motion bill revision addendum resolution repeal provision vote speed modification	motion time region eternity shape colour sound figure position
	motion relativity optic electromagnetism space heat	motion time region eternity shape colour sound figure position
	amendment motion bill revision addendum resolution repeal provision vote speed modification	texture shape colour size appearance colours flavour pallidum movement shine form sound
plant	plant house storefront life interest men woman quality man merchandise parent building	insect bird mammal butterfly plant mollusc spider fish wildlife creature protist beast
	plant house storefront life interest men woman quality man merchandise parent building	plant organism substrate fossil crustacean decor rock seedling worm protist
	accessory plant merchandise spare planter tool equipment shop	animal plant soil phytoplankton zooplankton cargo being crop vegetation men microbe moss

Table 6.4: The three most contrasting pairs of sense clusters for the test words used by Yarowsky.

none of which coincides with the word. The word *axes* (which has the base forms *ax*, *axe* and *axis*) happens to be one of these few left-out words. The results are based on a log-likelihood threshold of $\lambda = 6$. This is the highest threshold λ for which the other 10 test words can still be classified.

Our method discriminates well among the different senses of these test words, and contrasting clusters almost always correspond to different senses of a word.

Apart from making a broad distinction between the "fish" and "music" sense of *bass*, the clustering algorithm breaks down the "music" sense of *bass* even further into *bass* "musical instrument", *bass* "clef" and *bass* "singer". Similarly, the algorithm discovers both the "(military) vehicle" and "container" sense of *tank*, but further distinguishes between different types (and sizes) of containers: fish, gas and water tank. A finer sense distinction than the one proposed by Yarowsky is also made for the word form *space*. The contrasting meanings discovered are *space* "universe", *space* "volume" and *space* "area".

Our method correctly recognizes that *crane* has the two contrasting meanings of "bird" and "machine". In addition, *crane* is placed into a cluster which is composed mainly of writers' names, which accurately reflects the fact that *crane* is also a family name and, in particular, the name of a famous writer.

For the test word *duty*, the first and the third contrasting pair represent the sense distinction "responsibility" and "tax" given by Yarowsky. But our algorithm finds the additional "legal obligation" sense of *duty* which is defined in an online dictionary of legal terms [1] as

> **Duty:** In negligence cases, a "duty" is an obligation to conform to a particular standard of care. A failure to so conform places the actor at risk of being liable

[1] http://www.edgarsnyder.com/resources/terms/d.html

to another to whom a duty is owed for an injury sustained by the other of which the actor's conduct is a legal cause.

We failed to find the rather rare "beverage" sense of *sake*, which is probably simply due to data sparseness. The clustering instead discriminates two different, namely a religious and a secular connotation of *sake*.

Our algorithm distinguishes between (legal) *drug* as in medicine and its illegal sense. It seems that the "narcotic" sense of *drug* is closely associated with criminal activities in the BNC, which explains the high contrast which is wrongly attributed to the second pair of clusters which are both related to the same, "narcotic" sense of *drug*.

The most contrasting clusters extracted for the word form *palm* quite accurately reflect the "tree" and "body part" sense given in Table 6.3. The second and third contrasting pairs, however involve the rather heterogeneous cluster {*portico, tower, building, palm, stone, etc.*} which is only vaguely similar to *palm* "tree".

The clustering covers both the "legal" and the "physical" sense of *motion*. However, the clusters corresponding to *motion* "movement" contain several words which are only distantly related to that sense. Similarly, our method successfully identified the "organism" sense of *plant*. None of the other clusters, however, clearly reflects the meaning of "industrial plant".

The contrast measure seems to be vulnerable to semantically heterogeneous clusters because of their low LSA similarity to all other clusters. By filtering out heterogeneous clusters, the recognition and discrimination between different word senses could be further improved.

The within-cluster frequencies of the words in a cluster give some indication of the cluster's homogeneity or heterogeneity. Consider the following three clusters which all contain the word form *plant*. The number in parentheses following each word is the number of times it is present in the cluster.

1. plant (24), house (3), storefront (2), authority (1), mother (1), union (1), staff (1), merchandise (1), development (1), man (1), woman (1), wife (1), building (1), government (1), parent (1), number (1), money (1), life (1), family (1), germany (1), body (1), society (1), quality (1), men (1), age (1), interest (1)

2. rose (9), potentilla (8), flower (8), larkspur (6), greenery (4), mimosa (4), astrantia (4), plant (4), hydrangea (4), daisy (2), strawberry (2), hellebore (2), hawthorn (2), campanula (1), iris (1), foliage (1), cornflower (1), tree (1)

3. fortification (7), dockyard (5), bridges (4), harbour (3), canal (2), castle (2), mills (2), factory (2), plant (2), aqueduct (1).

Whereas Cluster 1 is characterized by the dominant presence of a single word (which is *plant*), Cluster 2 and Cluster 3 contain a core of *several* frequent words.

Remember that each cluster is a set of nodes in the link graph $L(\mathcal{W})$, and thus corresponds to a set of links in the original word graph $\mathcal{W}$, that is, to a subgraph of $\mathcal{W}$. Figure

6.10 shows the subgraphs of $\mathcal{W}$ associated with each of the three above clusters. Cluster 1 corresponds to a star-shaped graph which shows hardly any degree of transitivity. It contains merely a single triangle. The graphs associated with Clusters 2 and 3, instead show a much higher degree of clustering, meaning transitivity. The degree of clustering, that is the density of triangles within a cluster, reflects the diversity or homogeneity of the cluster members. Highly transitive clusters correspond to semantically coherent clusters, clusters with a low level of transitivity correspond to semantically heterogeneous clusters.

First sifting out heterogeneous clusters, may improve the quality of the contrasting cluster pairs.

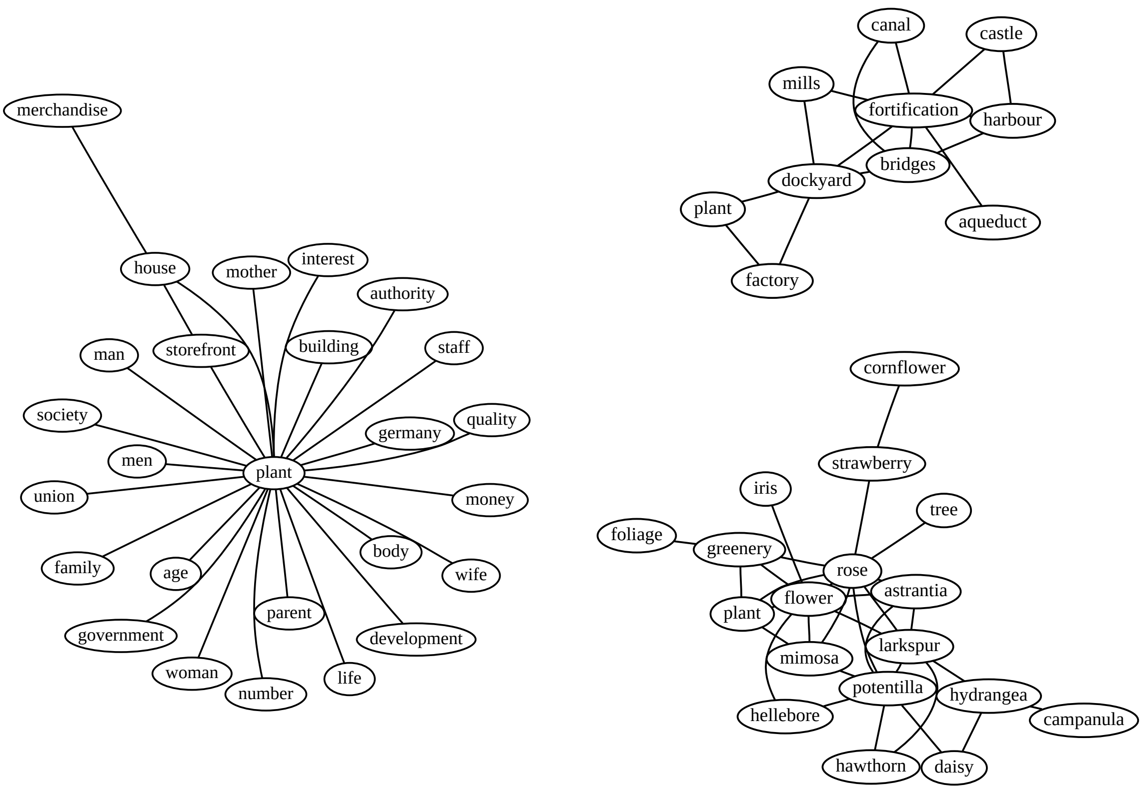


Figure 6.10: The subgraphs of $\mathcal{W}$ associated with three of the clusters the word form *plant* belongs to.

6.7 Conclusion

Link graph clustering has been shown to successfully discriminate between a word's different, even closely related, senses. Our algorithm computes clusters of relationships between words as opposed to clusters of words and splits the words as a "byproduct" which can then belong to several clusters which correspond closely to word meanings.

We have introduced the measure of contrast to tackle the problem of too finely grained clusters. The contrast measure seems quite effective in picking out pairs of clusters which are representative of different meanings of a word. Certainly, a more general measure of contrast, which allows to extract sets of more than two mutually contrasting clusters, would be even more valuable.

A way of extending contrast from pairs to larger groups of clusters may be by considering the clusters a word appears in as the nodes of a graph which are connected if their contrast is higher than a certain threshold. Maximal sets of mutually contrasting clusters then simply correspond to the maximal cliques of the graph. Figure 6.11 shows the graph of cluster contrasts associated with the word form *line* using a threshold of 0.5. Each triangle in the graph corresponds to a set of three mutually contrasting clusters.

Many of the nodes represent the same meaning of *line*, and hence, many links represent the same meaning contrast. For example, the link connecting Clusters 1 and 2, and the link connecting Clusters 1 and 4 both correspond to the distinction between the "geometric object" and "fishing line" sense of *line*. The problem of determining which of the clusters correspond to the same sense of *line* can be formulated as a (well-studied) *vertex-coloring problem* (Bollobás, 1998): Color the vertices of the graph in such a way that any two adjacent vertices are differently colored. Differently colored vertices are conflicting clusters. Equally colored vertices, on the other hand, are likely to correspond to the same meaning of *line*. Figure 6.11 shows one possible coloring of the graph. The coloring of a graph is, however, not necessarily unique. For example, vertices 5 and 6 may as well have been colored orange.

1. *line point click axis polygon*
2. *reel line rod winch knot gaffe handle hook*
3. *station men woman building bridge line government road*
4. *bait tackle line lure hook*
5. *circle dot spike line keyhole*
6. *contour curve line shape*
7. *junction road line signpost lamppost links railway car train canal*

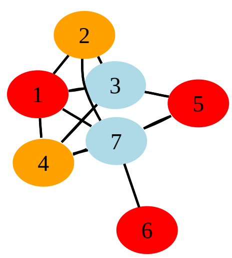


Figure 6.11: The graph of contrasts between clusters containing the word form *line*. Each node corresponds to a cluster. Two clusters are linked to one another if there is a significant contrast between them. A coloring of the graph which assigns different colors to adjacent nodes, may be interpreted as a partition into different senses of *line*.

The link graph transformation is just one out of a variety of other graph transformations. Graph transformations have been examined mainly from theoretical perspectives (Prisner, 1995; Bagga, 2004), but have also been applied to real data (Nacher et al., 2004; Eckmann et al., 2004). It would be interesting to study (both experimentally and theoretically) the effect of applying other graph transformations to the word graph or to other graphs built from other kinds of linguistic data.

Chapter 7

Idiomaticity in the word graph

7.1 Introduction

In order to speak and understand a language proficiently, extensive knowledge of its vocabulary and syntax alone is not sufficient. The English language (as well as many other languages) is rich in well-established word combinations whose comprehension and proper employment non-native speakers find extremely difficult to master. These established word combinations vary in their syntactic rigidity as well as in their semantic transparency and range from institutionalized phrases (preferred ways of expressing an idea), such as *ladies and gentlemen* (as opposed to *gentlemen and ladies*) and *bumps and bruises* (and not *bumps and scratches*), to fully-idiomatic expressions, such as *pull someones leg* or *kick the bucket*. We will refer to the family of such established word combinations as *multi-word expressions* (Sag et al., 2002).

Multi-word expressions do not only pose a challenge to second language learners, but also to computers trying to understand (let alone generate) natural language. Both lexicography and natural language understanding would greatly benefit from automated methods for their detection.

A natural application of the word graph is the automatic identification and partial classification of list-like multi-word expressions. These include coordinations which have become well-established among the native speakers of a language (*fish and chips*; *pipes and drums*), and many of which have assumed a meaning which can no longer be predicted from the meanings of their constituents (*smoke and mirrors*; *carrot and stick*). Lists are also a very popular way of naming things. Many names of products (*Macaroni and Cheese*, *Head and Shoulders*), book and movie titles (*The Beauty and the Beast*; *The Cook, The Thief, His Wife & Her Lover*), names of ministries or organizations (*Organization for Security and Co-operation in Europe*; *Ministry of Economy, Trade and Industry*), names of rock bands and sport teams (*Love and Rockets*, *Cherry and White*), etc. take the form of lists.

One of the main characteristics of established lists is their limited commutability. Changing their order makes them sound awkward (?*gentlemen and ladies*) and may even involve a (sometimes drastic) shift in meaning (?*breakfast and bed*). In the word graph, limited commutability of a list is reflected in the corresponding links being traversed preferably in one of the two possible directions.

7.2 Asymmetric links

For assembling the word graph, we decided to simply put a link between two words if they cooccurred in a list. By ignoring the order of the words, we implicitly assumed that lists are permutable. In terms of the word graph, this means that links are reversible. They simply connect nodes, but do not have a direction. However, certain lists have a strong preference for one order over the other and violate this symmetry/reversibility assumption. Lists such as *chalk and cheese*; *lock, stock and barrel*; *Romeo and Juliet*; *stars and stripes*; *Father, Son and Holy Ghost*; *bread and butter* and *mother and father* are almost always stated in a specific order. The undirected word graph studied so far doesn't account for these word order preferences.

7.2.1 Testing the reversibility (symmetry) of links

We can divide the set of coordinations a pair of words (w_1, w_2) occurs in into those in which w_1 precedes w_2, and those in which w_2 precedes w_1. We will denote the two sets with S_1 and S_2 and their sizes with $n_1 = |S_1|$ and $n_2 = |S_2|$. If S_1 and S_2 are approximately equally sized, we can conclude that there is no preferred order for w_1 and w_2, which means that the link connecting the two is reversible. A strong imbalance between n_1 and n_2, on the other hand, suggests that there is a bias towards one of the two possible orders, i.e., the link is directed.

To check whether a pair of words favors one order over the other, we apply the *Pearson Chi-Square test* (Pruscha, 1996). The Chi-Square test examines how well categorical data[1] fits a certain expectation (null hypothesis).

Imagine a categorical random variable X which falls into k different bins (categories) $c_1, \ldots, c_k$ with specified proportions (probabilities) $p_i = P(X \in c_i), i = 1, \ldots, k$. An independent sample $X_1, \ldots, X_N$ of the random variable X, will on average fall $E_i = Np_i$ times into bin c_i, $i = 1, \ldots, k$.

The Chi-Square statistic tells us how well the hypothesized category probabilities p_i fit the data. This is done by comparing the actually observed bin sizes O_i with the expected bin sizes $E_i = Np_i$:

$$\chi^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \tag{7.1}$$

The bigger χ^2, the more the observed category frequencies depart from the expected ones. Squaring the discrepancies $O_i - E_i$ prevents negative and positive differences from

[1] Data which can be divided into a small number of specific groups/categories. Examples of categorical variables are marital status (single/married/widowed/divorced), sex (male/female), employment status (yes/no), type of treatment (drug/placebo) or lexical categories (noun/verb/adjective/adverb/preposition).

canceling each other out. Furthermore, the normalization by the expected values makes the discrepancies comparable across categories. Assuming that all bins are equally likely, i.e.,

$$H_0: \ P(c_i) = p_i = \frac{1}{k}, \ i = 1, \ldots, k \quad , \tag{7.2}$$

as the number of trials N increases, χ^2 approaches the Chi-Square distribution χ^2_{k-1} with $k-1$ degrees of freedom.

Large values χ^2 are unlikely to occur under H_0 and indicate that the discrepancies between observed and expected frequencies cannot be explained by chance alone.

When there are only two categories (e.g., male/female or healthy/sick), many statisticians recommend using *Yates' correction* (Evert, 2004). The discrepancies $O_i - E_i$ in Formula 7.1 are substituted with the weakened differences $|O_i - E_i| - 0.5$:

$$\chi^2 = \sum_{i=1}^{k} (|O_i - E_i| - 0.5)^2 / E_i \tag{7.3}$$

This adjustment makes the value of the statistic smaller, and hence makes it more difficult to establish significance.

Extracting directed (non-reversible) links: In our case, there are two categories,

c_1: w_1 precedes w_2

c_2: w_2 precedes w_1

Our null hypothesis is that both cases are equally likely:

$H_0 : p_1 = p_2 = 0.5$

In other words, w_1 and w_2 can swap places in any list they co-occur in. The pair (w_1, w_2) occurs in $N = n_1 + n_2$ lists. If the null hypothesis is true, we expect the lists to split up into even halves ($E_1 = N * p_1 = N/2$ and $E_2 = N * p_2 = N/2$). The actually observed category frequencies are given by $O_1 = n_1 = |S_1|$ and $O_2 = n_2 = |S_2|$, and formula 7.3 becomes:

$$\begin{aligned}
\chi^2 &= \frac{(|O_1 - E_1| - 0.5)^2}{E_1} + \frac{(|O_2 - E_2| - 0.5)^2}{E_2} \\
&= \frac{(|n_1 - N/2| - 0.5)^2 + (|n_2 - N/2| - 0.5)^2}{N/2} \\
&= \frac{(|n_1 - N/2| - 0.5)^2 + (|N - n_1 - N/2| - 0.5)^2}{N/2} \\
&= \frac{4 * (|n_1 - N/2| - 0.5)^2}{N}
\end{aligned}$$

Since we are dealing with two categories, the test statistic χ^2 is approximated by the Chi-Square distribution with one degree of freedom.

Our experiments are based on a subgraph $\mathcal{W}'$ of the word graph $\mathcal{W}$ which is obtained as follows. We consider only those nodes (words) which were observed at least five times in a list, and the links between them. Furthermore, we drop all links whose log-likelihood weight is less than ten. We compute the Chi-Square statistic for each of the remaining links in the graph and filter out those links for which χ^2 exceeds the critical value $\chi^2_{0.01,1} = 6.63$.[2] This means that the category frequencies of n_1 and n_2 would have less than a 1% chance of occurrence if w_1 and w_2 were indeed commutable.

For example, consider Table 7.1 which lists the observed category frequencies and resulting χ^2-values of pairings of *chalk* with three other words. We found 27 lists containing both *chalk* and *cheese*. In each of these lists, *chalk* appeared before *cheese*. The extremely uneven category frequencies $O_1 = 27$ and $O_2 = 0$ give rise to a χ^2 value of 27 and a rejection of H_0. In the case of *chalk* and *flint*, the category frequencies are much more balanced ($O_1 = 5$ and $O_2 = 4$). In fact they are very close to expectation ($E_1 = E_2 = 4.5$) which is reflected in the very small χ^2-value of 0.11 which is well below the critical value. The word pair (*chalk, limestone*) seems to occur preferably in the order stated. The value of the test statistic χ^2 is close to the critical value, but the discrepancy in the observed category frequencies $O_1 = 10$ and $O_2 = 2$ may be simply due to randomness.

w_1	w_2	n_1	n_2	χ^2
chalk	cheese	27	0	27
chalk	flint	5	4	0.11
chalk	limestone	10	2	5.33

Table 7.1: The χ^2-values for three different links attaching to the *chalk* node.

Application of the cut-off criterion $\chi^2 \geq \chi^2_{0.01,1}$ results in $2,956$ word pairs with a strong preference for a specific order. Table 7.2 is an exemplary assortment of these asymmetric links which we manually assembled such as to cover the main factors influencing word order.

We identify two main causes for asymmetry, namely

- Semantic constraints affecting the order of the words, and
- Idiomaticity.

In the remainder of this chapter, we will examine both of these phenomena in more depth.

7.2.2 Semantic constraints

Certain systematic rules, which are listed below, seem to regulate the order of words in lists. The examples following each of the rules are taken from our list of asymmetric links.

[2] $\chi^2_{\alpha,k}$ is the smallest value for which $P(Z \geq \chi^2_{\alpha,k}) < \alpha$, where Z is a random variable which is distributed according to the Chi-Square distribution with k degrees of freedom.

w_1	w_2	n_1	n_2	χ^2
man	woman	2965	423	1907.25
bed	breakfast	433	2	427.04
life	death	313	24	247.84
strength	weakness	297	23	234.61
ladies	gentleman	247	5	232.4
success	failure	279	17	231.91
north	south	270	14	230.76
day	night	555	172	201.77
fish	chip	209	8	186.18
bread	butter	188	2	182.08
rise	fall	175	10	147.16
spring	summer	156	4	144.4
birth	death	120	7	100.54
doctor	nurse	207	45	104.14
landlord	tenant	127	12	95.14
parent	guardian	68	1	65.06
rock	roll	58	0	58
odd	end	44	0	44
hue	cry	40	0	40
twists	turn	40	1	37.1
childhood	adolescence	39	1	36.1
beauty	beast	35	1	32.11
chalk	cheese	27	0	27
son	grandson	29	0	29
god	goddess	31	3	23.06
spring	autumn	74	26	23.04

Table 7.2: A subset of the graph's links together with their χ^2 values.

From now on, we will use the notation $x > y$ to express that word x tends to appear before word y.

1. **male > female:** man > woman; brother > sister; boy > girl; prince > princess; hero > heroine.
2. **old > young:** father > son; son > grandson; woman > child; wife > daughter; host > hostess.
3. **big > small:** church > chapel; hotel > restaurant; street > alley; university > college; town > village.
4. **prototypical > nonprototypical:** apple > cherry; beef > mutton; biology > zoology; iron > manganese; wool > nylon; oil > resin.
5. **early > late:** spring > autumn; morning > afternoon; beginning > end; arrival > departure; shampoo > conditioner.
6. **strong/important/valuable > weak/less important/less valuable:** landlord > tenant; teacher > pupil; cardinal > bishop; doctor > nurse; gold > silver; friend > acquaintance.
7. **positive > negative:** love > hate; rise > fall; joy > sorrow; boom > slump; winner > loser; friend > foe.
8. **second requires first:** horse > rider; father > son; pregnancy > birth; war > aftermath; question > answer; marriage > divorce; crime > punishment; input > output; dog > owner.
9. **more specific > less specific:** pot > container; theft > offence; car > vehicle mathematics > science; farmer > worker; director > person; cat > animal; heroin > drug.
10. **human > nonhuman:** people > animal; child > pet.
11. **animate > inanimate:** man > machine; people > situation; women > politics.
12. **concrete > abstract:** body > soul; food > aid; bed > sleep; hardware > software.

These constraints overlap largely with those identified by Müller (1998) and Benor and Levy (2006) in their studies. Some of the semantic constraints are more salient than others. Whereas we observe a tendency to name males before females, strength (Rule 6) seems to override this male first bias, as can be seen in the examples *queen* > *duke* and *landlady* > *lodger*. This is also observed in parenting relationships where females appear before males (*mother* > *father*; *mum* > *dad*; *grandma* > *grandpa*). Perhaps it is usually the mother who is bringing up the children, which would make her more important than the father as far as parenting is concerned.

Several of the above listed relationships, such as maturity (Rule 2), temporal or logical precedence (Rule 5), dominance (Rule 6), specificity (Rule 9) and size (Rule 3) are hierarchical in nature. Rather than being scattered over the word graph, they attach to each other, forming fully-directed subgraphs, some of which are illustrated in Figures 7.1 through 7.4. Figures 7.1 and 7.2.2 are composed of links which are directed due to ranking according to both maturity/pedigree and gender. Relationships with temporal and/or logical order give rise to the directed subgraphs depicted in Figures 7.3 and 7.4.

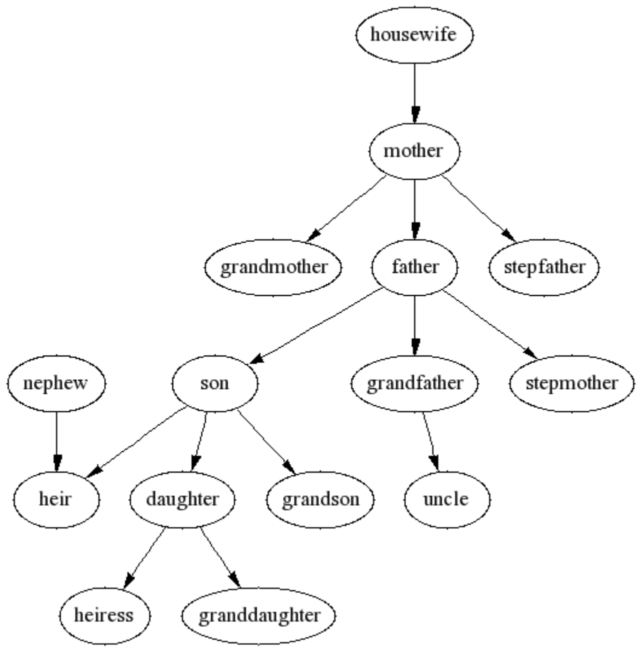


Figure 7.1: Relationships between family members.

The kinds of links which we have covered in this section are only restricted in terms of their word order which has become fixed over time or is the result of certain systematic semantic (sometimes also phonological) rules. As far as meaning is concerned, they are fully transparent.

However, a non-negligible number of the asymmetric links in the word graph arises from expressions which, in addition to having a rigid word order, constitute lexical units in their own right, often with a non-literal meaning. Their meaning cannot be deduced from the individual words they are composed of. These fall into the category of idioms, the semantically most opaque word combinations, and thus the most difficult expressions for non-native speakers of a language (and a computer) to understand and produce.

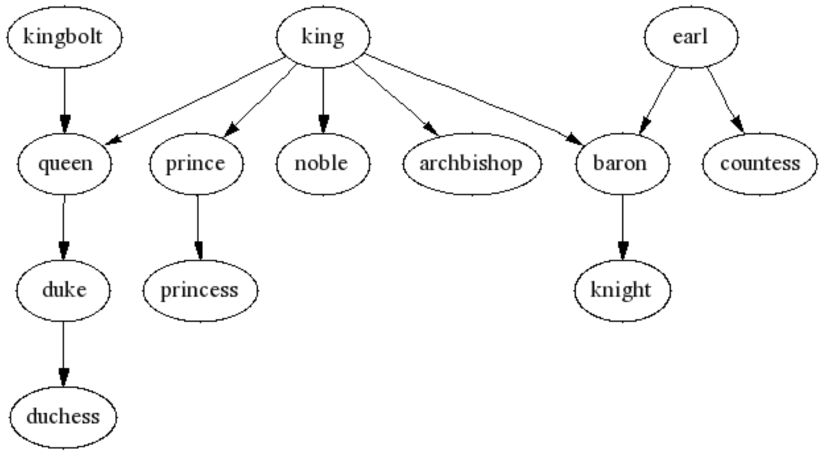


Figure 7.2: Hierarchical relationships between aristocrats.

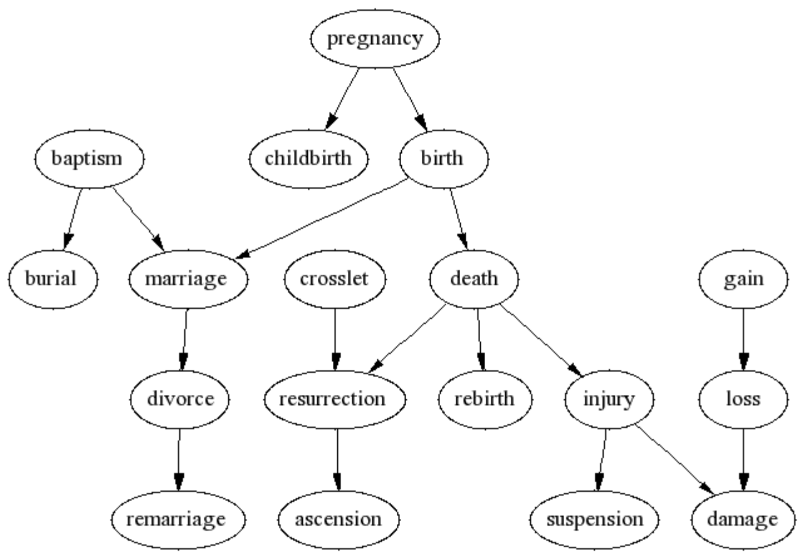


Figure 7.3: Different events in one's life and their temporal relationships.

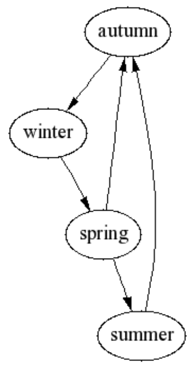


Figure 7.4: Chronology of the seasons.

7.3 Idiomatic constraints

Idioms are classically defined by the following two properties (Gläser, 1998; Burger, 1998):

1. **Lexicalization:** Idioms have become established among native speakers of a language (or within smaller communities). They constitute lexical units in their own right.

2. **Noncompositionality of meaning:** The meaning of an idiom can, if at all, only be partially predicted from the meanings of its parts.

Idiomaticity comes on a scale, and it is often extremely hard to decide whether an expression should be considered an idiom or not. Phrases which are often cited as prototypical idioms include *kick the bucket*, *spill the beans*, *pull someones leg* and *piece of cake*, but the definition given above applies also to expressions such as *stars and stripes*, *bed and breakfast* or *rock and roll*.

In addition to the defining characteristics 1 and 2, idioms often exhibit the following properties (Nunberg et al., 1994; Burger, 1998):

3. **Semantic anomaly:** The words involved have incompatible meanings. Examples are *chalk and cheese*; *smoke and mirrors*; *carrot and stick*.

4. **Limited syntactic flexibility:** Idioms often show little (if any) variation in form. In particular, the individual words within an idiom undergo little modification, and their order is rather rigid.

5. **Limited semantic flexibility:** In contrast to free word combinations, many idioms lose their original, i.e., their idiomatic meaning if individual words are replaced with (near-) synonyms.

6. **Broad scope:** Many idioms describe very general and common situations (Nunberg et al., 1994) (e.g., working too much (*all work and no play*), arguing (*cut and thrust*), (something happening) rapidly (*in/by leaps and bounds*)).

Characteristics 3 to 6 are not necessary conditions for idiomaticity. Idioms often do show some greater of lesser degree of syntactic or semantic variability (*chalk and bloody cheese*; *bits and bobs/pieces*). Similarly, many idioms are in fact composed of semantically compatible words (*morning, noon and night*; *(play) cat and mouse*). Nonetheless, these additional characteristics provide important clues as to whether and to which degree an expression is idiomatic.

The remaining part of this chapter describes different approaches for making recognition of idioms automatic. So far, it is a collection of ideas which has yet to be explored in detail. In the following section, we present a combinatorial method for determining whether a link is idiomatic or not. It will rely solely on the link structure of its neighborhood in the word graph.

7.4 Link analysis

Many idioms respond highly sensitively, to semantic transformations (cf. Characteristic 5). Since the individual components of an idiom assume and contribute a (often metaphoric) meaning which is different from their literal one, they usually cannot be replaced with synonyms or near-synonyms.

Consider the idiom *carrot and stick*, for example. By substituting another vegetable, e.g., *tomato*, for *carrot*, the original idiomatic meaning gets lost [3].

In terms of the word graph, replacing a constituent word with a near-synonym means rewiring one of a link's ends to a neighboring node. Consider a coordination x *and* y. Each possible way of paraphrasing x *and* y using a (near-) synonym x' of x, or a (near-) synonym y' of y, gives rise to a triangle (x, y, x') or (x, y, y') (cf. Figure 7.5). This means, the more triangles a link occurs in, the more flexible it is in terms of lexical substitution.

This idea is illustrated in Figure 7.6 for the idiomatic expression *rhyme and reason*. The *rhyme* and *reason* nodes do not share a single neighbor and consequently do not co-occur in a single triangle. This means that neither of the two can be replaced by a semantic neighbor.

Besides reflecting the lexical substitutability of an expression, the triangle count indicates the degree of semantic incompatibility of the words which make it up (cf. Characteristic 3). Whereas the words in a list are usually semantically similar, idiomatic expressions, are often composed of apparently very dissimilar concepts. Typical examples are *chalk and cheese* and *theft and fire*. It seems that there is a strong correlation between the semantic transparency of a list and the semantic relatedness of its coordinated words.

[3] It's more imaginable that idiomaticity can be maintained by replacing *carrot* with *chocolate*, for example, since *chocolate* and *carrot* have similar metaphoric interpretations (namely as being a *reward*).

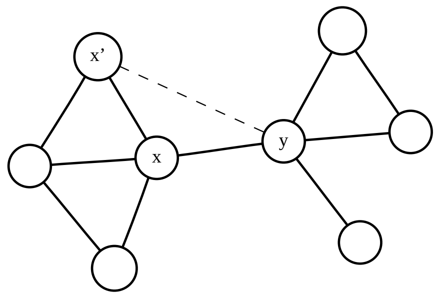


Figure 7.5: Lexical substitution in the word graph: If an expression x *and* y can be paraphrased as x' *and* y, where x' is a neighbor of x in the word graph, the link connecting x and y occurs in a triangle with corners x, y and x'.

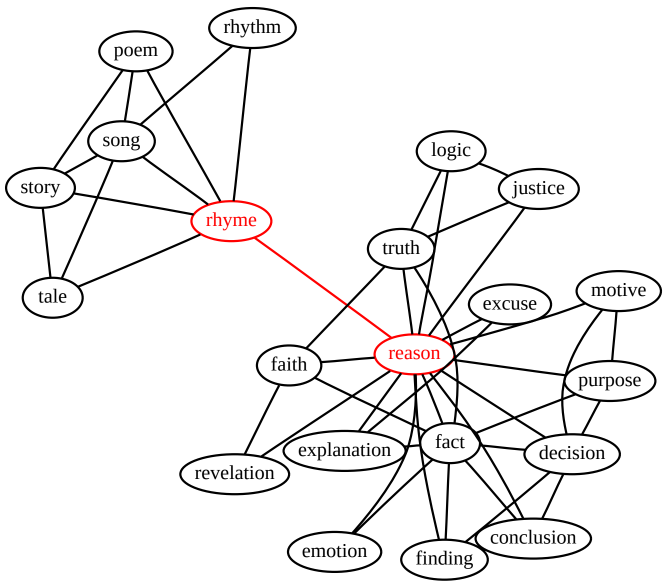


Figure 7.6: The neighborhood of the link connecting the words *rhyme* and *reason*.

In Chapter 5, we observed that ambiguous words correspond to *nodes* which connect different node clusters in the word graph. In the same vein, idiomatic links correspond to *links* connecting otherwise distant clusters. For example, the idiom *rhyme and reason* is the sole connection between the mutually exclusive semantic neighborhoods of *rhyme* and *reason*. The neighborhoods of semantically compatible words, such as *rhyme* and *song*, on the other hand, have considerable overlap (cf. Figure 7.7).

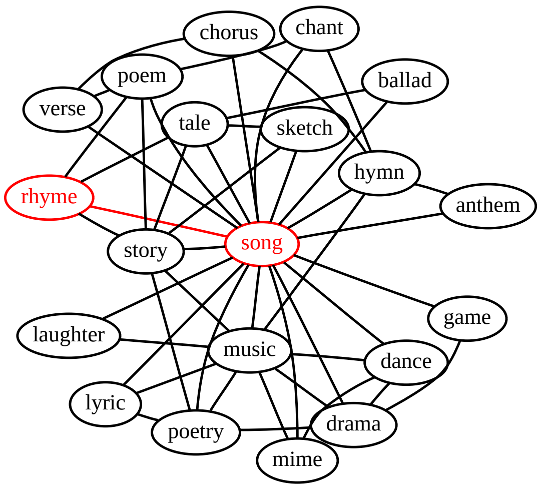


Figure 7.7: The neighborhood of the link connecting the words *rhyme* and *song*.

As we have seen, lists composed of two semantically clashing words correspond to edges which bridge between two mutually exclusive node clusters. Similarly, lists composed of three semantically incompatible words correspond to triangles linking three contradictory semantic neighborhoods.

Figures 7.8 and 7.9 show the network structure surrounding the two word triples (*man, woman, child*) and (*lock, stock, barrel*). Both triples have a strong preference for one word order (the one specified) over the other possible five. However, they differ a great deal in their analyzability. The meaning of *man, woman and child* is fully predictable and simply the composition of the semantically compatible individual meanings. The meaning of the expression *lock, stock and barrel*, on the other hand, cannot be derived from the individual meanings. Lists are usually semantically homogeneous, and whenever they are not, there is strong evidence that the list has a non-literal interpretation.

The difference in compatibility shows nicely in the topology of the graphs surrounding the two triples. Whereas the triangle (*man, woman, child*) is well-embedded in the dense subgraph of human beings surrounding it, the triangle (*lock, stock, barrel*) is much less well-integrated in its neighborhood. It sits in between different node communities corresponding to semantically distant categories.

Several questions remain open which need to be addressed in future work. Mere triangle counts do not adequately reflect idiomaticity. Naturally, links connecting frequent words are much more likely to appear in triangles than links between infrequent words. We therefore

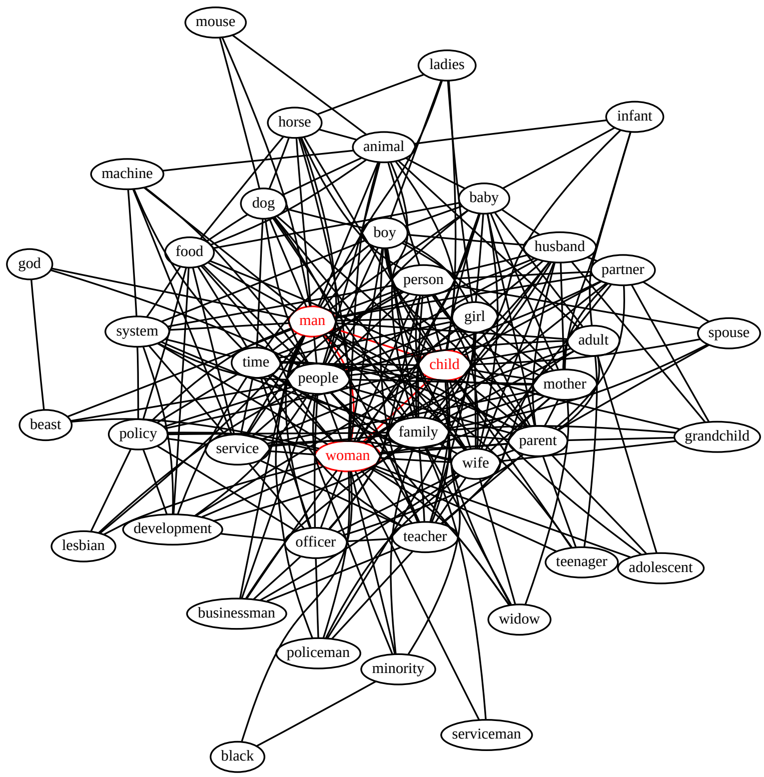


Figure 7.8: The neighborhood of the links connecting *man*, *woman* and *child*.

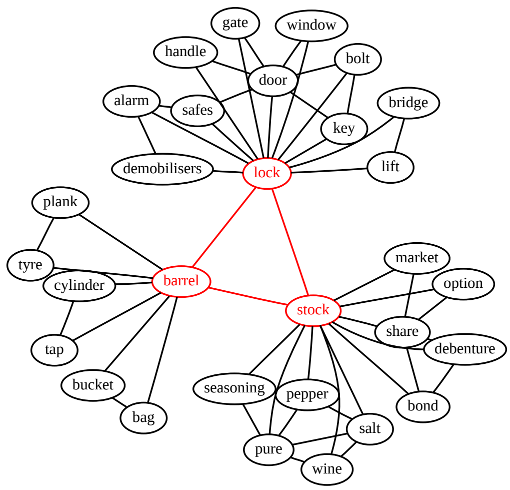


Figure 7.9: The neighborhood of the links connecting *lock*, *stock* and *barrel*.

need to find a (natural) way of adjusting the triangle counts to the frequencies of the words in the coordination.

A link between two words which form part of a list of n words, naturally occurs in at least $n-2$ triangles, irrespective of its idiomaticity. In order to properly deal with list composed of more than two words, we propose to assign weights to the triangles of the graph which reflect the degree of asynchrony among their three sides. The edges of highly synchronized triangles, such as *lock, stock* and *barrel* occur only simultaneously (within the same list). The triangle (*man, woman, child*) is much less synchronized. Its three links do often co-occur in the same list, but not always. There are many lists which involve only two of the three words. Eckmann et al. (2004) propose an information-theoretic measure of the degree of synchronization between the nodes in a dynamic graph which may be adapted to suit our purposes.

There are many idioms which are in fact composed of semantically similar words, such as, for example, *(cost an) arm and a leg* and *skin and bones*. Such links, in spite of their idiomaticity, occur in many triangles as their constituent words share many semantic neighbors.

The following sections are intended to provide a series of ideas on how idiom recognition can be improved using statistical techniques in addition to the combinatorial method described in this section.

7.5 Statistical analysis

Rather than presenting a set of fully implemented and tested techniques, this section is a collection of ideas for improving idiom recognition, which have yet to be explored further. The following subsections are devoted to the distinguishing characteristics of idioms listed in Section 7.3, and to their assessment.

7.5.1 Noncompositionality of meaning

We expect that the context surrounding an idiomatic expression is very different from the contexts the individual words occur in. Instead, the contexts of a fully compositional word combination are similar to the contexts of its constituent words. Consider the following example sentences:

1. *... closed galleries. The finances of the two museums are <u>chalk and cheese</u>, with taxpayers providing a mere 17% of ...*

2. *... party. With lots of beer and home-made <u>bread and cheese</u> and ham and some of my mother's special cakes ...*

3. *... removal of about 81 tonnes of topsoil, <u>clay and chalk</u> and the use of a total weight of timber, ...*

The topic of the sentence containing the idiomatic expression *chalk and cheese* is "finance", which is fully unrelated to the topics of the other two sentences, which each contain one of the individual words *cheese* and *chalk*. Instead, Sentence 2 which contains the free coordination *bread and cheese*, talks about food, as most sentences containing one of *cheese* or *bread* do. Similarly, Sentence 3 is similar to many contexts containing *chalk* or *clay*.

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) provides a framework for comparing the contexts of a word pair with those of the individual words. LSA yields a vector representation for the words in a corpus (which can be seen as a profile of its usage in the corpus), and the distributional similarity between two words is measured by the cosine of the angle enclosed by the corresponding vectors. Similarly, any context can be represented as a vector in the same LSA space, which is the (normalized) sum of the words in the context, and, just as similarity between words, similarity between contexts can be computed using cosine similarity.

7.5.2 Semantic anomaly

In Section 7.4, we have argued that links which occur in few if any triangles are composed of semantically anomalous words. Another way of measuring semantic incompatibility of the words in a list is by means of a statistical or taxonomy-based word similarity measure.

Distributional word similarity: The LSA similarity just mentioned can be used to measure the similarity between the words in a list. Latent Semantic Analysis has been used as a filter before. Cederberg and Widdows (2003), for example, improved the precision of pattern-based hyponymy extraction by using LSA similarity between the candidate hypernym and hyponym as a measure of confidence in the extracted relationship. And Baldwin et al. (2003) have used Latent Semantic Analysis to distinguish between compositional and non-compositional verb-particle constructions.

Table 7.3 shows examples of asymmetric word pairs (as extracted in Section 7.2.1) and their LSA similarities (based on an LSA model built from the BNC).

Whereas there are many idioms among the word pairs with low LSA similarity scores (e.g., *rhyme and reason*, *taste and pocket*), we find also several fully compositional word pairs among the lowest scoring ones. The reason is that many of them involve a very general or ambiguous word. The word *attention*, for example, is ambiguous between *attention* "care" and *attention* as in *to pay attention*. Rather than reflecting idiomaticity, the low LSA similarity score assigned to the word pair (*love*, *attention*) is the result of the ambiguity of the word form *attention*. The same phenomenon gives rise to the low LSA score between *background* and *interest*.

As was already discussed in the previous section, quite a lot of idioms, such as *(in every) nook and cranny*, *(to earn one's) bread and butter* and *(to cost an) arm and leg*, are semantically fully homogeneous, and hence their constituents are close to each other in the Latent Semantic Space. In fact, lots of idiomatic expressions, such as *bits and pieces* and *twists and turns*, make use of semantic repetitions, sometimes even using exactly the same words as in *run and run* or *out and out*.

Word pair	LSA sim
love attention	-0.23
rhyme reason	-0.07
taste pocket	-0.05
background interest	0.04
star stripe	0.12
family colleague	0.14
rank file	0.17
hat umbrella	0.25
bucket spade	0.26
totem taboo	0.34
cat mouse	0.36
spic span	0.55
echo bunnymen	0.57
jekyll hyde	0.63
kith kin	0.64
bits bobs	0.7
arm leg	0.73
gold silver	0.75
bread butter	0.79
jacket trouser	0.88
question answer	0.91
north south	0.93
nook cranny	0.95

Table 7.3: Examples of asymmetric word pairs and their LSA similarities.

Some idiomatic expressions are composed of words which do not occur in the corpus outside idioms. These include *spic and span*, *bits and bobs* and *kith and kin*. For example, since *bobs* is only observed in the context of *bits*, the LSA similarity of the two words is extremely high. The same problem occurs for named entities such as *Jekyll and Hyde* and *Echo and the Bunnymen* none of whose constituents occurs in isolation.

Taxonomy-based word similarity: Several measures of semantic relatedness have been proposed which rely on taxonomic information provided the WordNet database, and have been implemented by (Pedersen et al., 2004) in the WordNet Similarity package [4]. We have described one of them, the similarity measure developed by Lin (1998) earlier in Section 5.4.2 of Chapter 5.

Table 7.4 lists some of the asymmetric links and the WordNet similarities assigned to their constituent words using Lin's similarity measure.

Word pair	WN sim
lies statistics	0
bubble squeak	0
love duty	0
music lyric	0
colitis disease	0
bread circuses	0.062
pen ink	0.062
snake ladder	0.129
taste pocket	0.135
nook cranny	0.179
rhyme reason	0.254
ball chain	0.419
arms legs	0.884
bits pieces	1
twists turns	1

Table 7.4: Example word pairs and their WordNet similarities.

Again, we find some idioms among the word pairs with high WordNet similarity score which do not fulfill the assumption of semantic incompatibility.

Inspection of the results shows that semantic dissimilarity alone doesn't suffice for distinguishing idiomatic from non-idiomatic expressions. Nevertheless, semantic similarity measurements might prove useful as additional factors for extracting idioms.

We find several idioms among the word pairs which are assigned a WordNet similarity of zero (e.g., *lies and statistics*, *bread and circuses*). However, the set of zero similarity word pairs includes many fully compositional word pairs which are members of the same

[4] http://search.cpan.org/dist/WordNet-Similarity/

semantic field but nevertheless occur at very different places in the taxonomic tree. These include *love and duty*, *music and lyric* and *colitis and disease*. According to WordNet, *colitis* is an *inflammation* which is subsumed by the top node *psychological feature*. A *disease*, on the other hand, is encoded as an *illness* which is subsumed by the top node *state*. Consequently, the WordNet similarity measure wrongly judges these the word pair (*colitis*, *disease*) as being highly idiomatic.

7.5.3 Limited syntactic variability

Idioms are not only rigid in terms of word order, but also in terms of other structural variation, such as modification of the individual words. It seems that the syntactic flexibility of an expression and the transparency of its meaning are closely correlated (Fazly et al., 2005).

Highly idiomatic phrases are hardly ever modified. The word pair (*chalk*,*cheese*), for example, occurs 21 times in the corpus. Twenty of the occurrences take the form *chalk and cheese*, and only a single one takes the different form *chalk and bloody cheese*.

Fully transparent word combinations, on the other hand, are fully flexible syntactically. The coordinations containing the word pair (*bread, cheese*), for example, come in many different forms, such as:

- bread and the marvellous brown norwegian cheese
- cheese and fresh crusty bread
- bread and mousetrap cheese
- cheese and some bread
- bread and mouldy cheese
- bread and wine and cheese
- cheese and rye bread
- cheese , olive , sardine and bread.

We can think of a word pair's different syntactic forms as being the different outcomes $x_1, \ldots, x_n$ of a random variable X, whose probabilities $p_i = P(x_i)$ are given by their relative frequencies of occurrence. The degree of variability in a word pair's surface forms can be measured using *Shannon entropy*, which is a measure of the amount of randomness in a discrete probability distribution P_X, and which is defined as follows:

$$H(P_X) = -\sum_{i=1}^{n} p_i * \log(p_i) \tag{7.4}$$

$H(P_X)$ assumes the minimal possible value of zero if the probability distribution P_X is deterministic, i.e., if there exists a j such that $p_j = 1$ and $p_i = 0$ for all $j \neq i$. It is maximal when all possible outcomes x_i are equally likely, in which case it assumes a value of $log(n)$. Table 7.5 shows some example word pairs and their entropy scores.

Word pair	Entropy
pro con	0
hustle bustle	0
dustpan brush	0
benson hedge	0
part parcel	0.06
gin tonic	0.08
trial error	0.12
nook cranny	0.13
intent purpose	0.15
beauty beast	0.18
chalk cheese	0.23
bed breakfast	0.43
earl countess	0.44
trial conviction	0.47
disease recovery	0.54
melody accompaniment	0.65
bread butter	0.99
actor audience	1.06
rebellion revolution	1.25
chalk flint	1.66
cheese onion	2.01
bread cheese	2.73
bed mattress	2.95
fruit yoghurt	3.35
sand rock	3.69
bacon bread	4.25
shirt tie	5.31

Table 7.5: Example of word pairs and their entropy.

7.5.4 Variability of context

While free word combinations tend to occur in very specific contexts, idioms describe much more general situations and are distributed across a wide range of topics. Again, we will use the two coordinations *chalk and cheese* and *bread and cheese* to illustrate this idea. The free coordination *bread and cheese* occurs almost exclusively in contexts dealing with food and eating:

- *...party. With lots of beer and home-made bread and cheese and ham and some of my mother's special cakes ...*

- *... o'clock! Dinner time!" It was bread and cheese. Sam took his out into the sunshine. Rose ...*

- *... ate the contents of the saucepan and some bread and cheese and an apple, and drank most of the wine ...*

Sentences containing the idiomatic coordination *chalk and cheese*, instead cover very different domains ranging from finance to parenting:

- *... closed galleries. The finances of the two museums are chalk and cheese, with taxpayers providing a mere 17% of ...*

- *... that game to next Sunday's contest is to compare chalk and cheese. But if nothing else, the McKenna ...*

- *... parents describe their children as being as different as chalk and cheese. They can often be blind to ...*

The contexts surrounding an idiom tend to vary more than the contexts surrounding a fully compositional word combination. Nunberg et al. (1994) refer to this phenomenon as the proverbial character of many idioms:

> "Idioms are typically used to describe – and implicitly, to explain – a recurrent situation of particular social interest (becoming restless, talking informally, divulging a secret, or whatever) in virtue of its resemblance or relation to a scenario involving homey, concrete things and relations – climbing walls, chewing fat, spilling beans."

The diversity of the contexts an expression occurs in, in addition to the discriminating factors described in the preceding sections, gives helpful additional clues to its idiomatic character. Again, Latent Semantic Analysis provides a framework for measuring the diversity of a potential idiom's contexts: The more scattered the corresponding context vectors are in Latent Semantic Space, the more variable the contexts, and hence the bigger the scope of the expression. Similarly, similar contexts give rise to context vectors which point in very similar directions in LSA space, and are indicative of the expression having a very narrow scope.

In the following, we describe several methods for measuring the degree of dispersion of a set of context vectors in LSA space.

Each context vector corresponds to a point in Latent Semantic Space (namely to its "head"). The size, that is the volume or diameter, of the smallest box which contains all these points can be taken as indicating the degree of dispersion of the corresponding contexts[5]. It has to be noted that this approach may be sensitive to outliers in form of rare, peculiar contexts.

[5]This idea has been developed by Dominic Widdows and Aria Haghighi, but has not been published; hence this footnote instead of a reference.

Another way of assessing the dispersion of a set of context vectors in LSA space is by means of the measure of *circular standard deviation* (Fisher et al., 1987), which allows to measure the dispersion of circular data, that is data which lies on a circle, or more generally, a sphere. Since the context vectors resulting from Latent Semantic Analysis are all of unit length, their endpoints lie on the surface of a (high-) dimensional unit sphere.

For a given set of vectors $\underline{v_1}, \ldots, \underline{v_n}$, we can compute the average vector $\underline{v} = (1/n) \cdot \sum_i \underline{v_i}$. The length $\|\underline{v}\|$ of $\underline{v}$ assumes a value between zero and one. Highly dispersed context vectors give rise to a short average vector $\underline{v}$. If the context vectors point in similar directions, $\|\underline{v}\|$ is close to 1 (cf. Figure 7.10). The circular standard deviation of $\underline{v_1}, \ldots, \underline{v_n}$ is defined as

$$\sigma(\underline{v_1}, \ldots, \underline{v_n}) = \sqrt{-2 \cdot log\|\underline{v}\|}. \tag{7.5}$$

It is monotonically decreasing with $\|\underline{v}\|$, and thus returns small values for similar, and large values for highly dispersed context vectors.

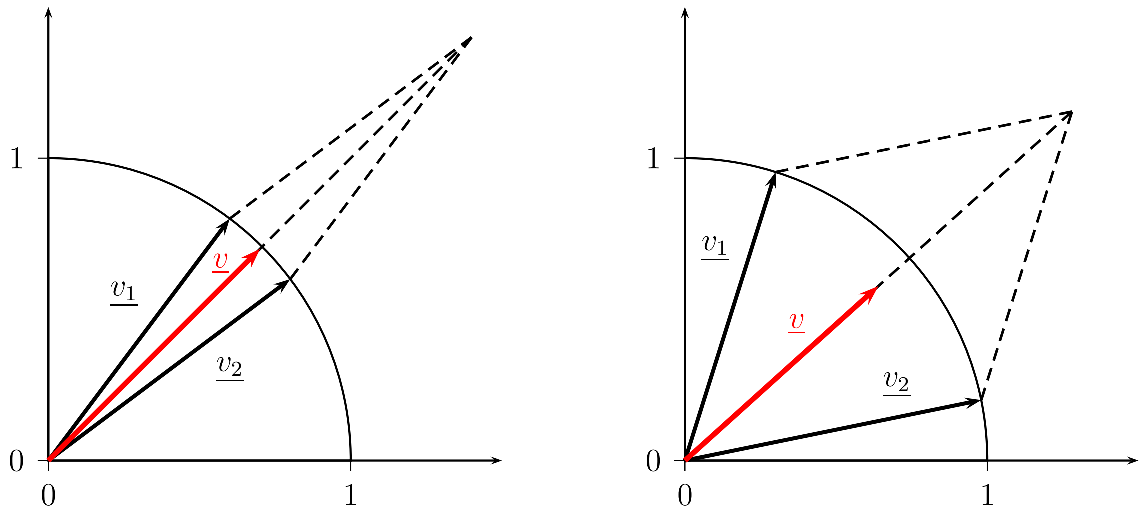


Figure 7.10: Two unit-length vectors pointing in *similar* directions give rise to a *long* average vector $\underline{v}$ (left). Vectors pointing in very *different* directions result in a *short* average vector.

Yet another way of measuring the variability of a set of contexts is by dividing the documents of the corpus into topic clusters using a standard clustering method, such as K-means clustering (Manning and Schütze, 1999)), and by examining how much the contexts are scattered across the topic clusters, for example, using a measure of entropy (Shannon, 1948).

Word pairs with a spiky distribution, i.e. which occur frequently in a small subset of clusters and which are absent in all other clusters, are likely to denote very specific, unidiomatic concepts. On the other hand, expressions which are evenly distributed across the topic clusters are likely to describe more general situations and are hence good candidates for idioms.

It has to be noted that variability of context is a distinguishing trait of "classical" proverbial idioms, such as *spill the beans*, *beat around the bush* or *piece of cake*. According to the definition given in Section 7.3, the class of idioms includes also specialized terms, such as *cut and paste* or *park and ride*, which are not fully compositional in meaning. In contrast to proverbial idioms, these expressions are highly specific. This means, they show low context variability, and will hence not be recognized as idiomatic expressions. The same applies to named entities, such as *Bubble and Squeak* (a traditional British dish made from leftovers) or *Snake and Leader* (a traditional Indian game), whose meanings are fully opaque, yet they denote highly specific concepts.

7.6 Conclusions

This chapter has explored the set of word pairs which give rise to asymmetric links in the word graph, that is, pairs of words which occur almost always in a particular order. Two causes for this asymmetry have been identified: Semantic and idiomatic constraints. We proposed several techniques, both combinatorial and statistical, for distinguishing idiomatic word pairs from compositional ones, which are based on syntactic and semantic characteristics of idioms. Many of these approaches have yet to be implemented and tested. The most difficult question which has to be addressed is how the proposed measures can best be combined.

Chapter 8

Synonyms in the word graph

In this chapter, we describe the use of the word graph for the automatic extraction of synonyms. Synonymy is a relationship between different words, more precisely, word *senses*, expressing the same concept. Examples of synonyms are *aristocracy* and *nobility*, *car* and *automobile*, *flora* and *vegetation*, and *chest* and *thorax*. Although alternate forms are not considered synonyms in the strict sense, we extend the term "synonym" to such forms as abbreviations (*refrigerator* and *fridge*; *bicycle* and *bike*), acronyms (*European Union* and *EU*; *television* and *TV*), and spelling variants (*moslem* and *muslim*; *urbanisation* and *urbanization*).

The availability of synonyms is fundamental to many applications in natural language processing, such as Information Retrieval, as it enables a retrieval system to paraphrase queries; and Document Summarization, as it helps to identify duplicate information to be fused.

8.1 Hypothesis

Most semantically related words do not only cooccur with the same words, but cooccur with each other, too. For example, look at this short passage from the BNC which contains three members of the semantic category of *vehicles*:

> Two drivers escaped injury when their *vehicles* collided near Thirsk. The accident happened on the southbound carriageway of the A19 and involved a *VW van* and a *Ford Sierra car*.

Synonyms, however, tend to defy this schema. Synonyms are alternative names for the same object or idea, and since we often prefer using one of the names over the others (or are simply not aware of other naming options), we tend to stick to the same name throughout an entire document (when writing) or conversation (when speaking). Consequently, synonyms scarcely appear in each others vicinity in the text. In analogy to the "one sense per discourse" principle of Gale et al. (1992), this argument may be put as follows: There is only one word per sense in a discourse.

This claim is a bit too strong, as we can well imagine an author making deliberate use of synonyms to avoid monotony from using the same word over and over. Yet, it becomes much more plausible if we restrict the very general notion of cooccurrence to mean cooccurrence in a specific grammatical context.

Lists, for example, contain synonyms only in a very small number of exceptional cases. As we have seen in Section 7, some expressions, such as *bits and pieces*, *twists and turns*, *hale and hearty* and *such and such* are composed of synonymous words (or even repetitions of the same word) to be more expressive, but these cases are extremely rare and mostly idiomatic.

Typically, each of the words in a list contributes some unique meaning to the meaning of the whole, which is not covered by the other list elements. Synonyms, however, constitute repeated, that is redundant semantic information.

In short, the assumption we make is that synonyms tend to occur with the same words in lists. But unlike other semantically related words, synonyms rarely occur together in the *same* list. The interpretation of this hypothesis in terms of the word graph is the following:

Hypothesis 1 *Synonyms correspond to nodes which are not (or only weakly) linked to one another, but which are (strongly) linked to the same other nodes.*

In other words, synonyms are weak first-order, but strong second-order neighbors in the graph. Figure 8.1 shows the section of the word graph surrounding the *pedestrian* and the *walker* node. It is important to note that the two nodes are not immediate neighbors in the graph. Yet, they are strongly held together by their shared neighbors, *cyclist*, *rider* and *motorist*.

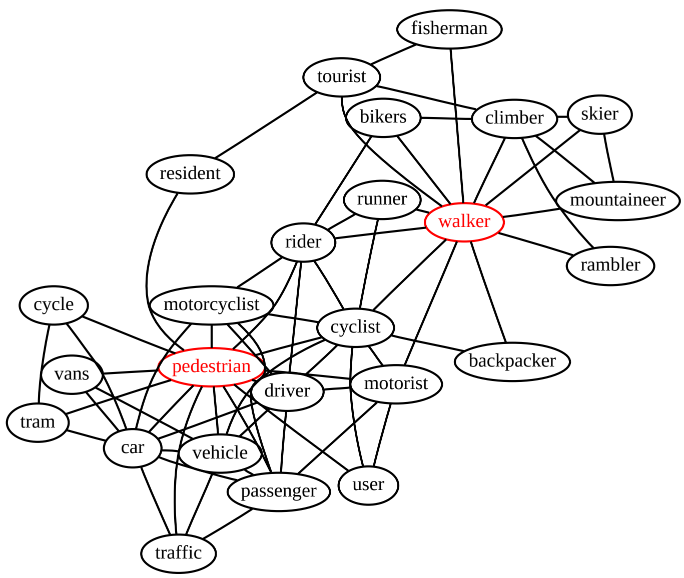


Figure 8.1: The neighborhoods of the *pedestrian* and the *walker* node. The two nodes are not linked to one another directly but have several neighbors in common.

8.2 Methodology

As mentioned earlier and exploited by the Markov Clustering algorithm (cf. Section 4.2), the Markov matrix $T_{\mathcal{G}}$ of a graph $\mathcal{G}$, that is its column-normalized adjacency matrix, contains the node-to-node transition probabilities of a random walk on the graph. More generally, the k-th power $T_{\mathcal{G}}^k$ of $T_{\mathcal{G}}$ lists the probabilities of moving from one node to another in k steps. The matrix $T_{\mathcal{G}}^k$ can be interpreted as the adjacency matrix of a weighted directed graph $\mathcal{G}_{T^k}$.

The first row of Figure 8.2 shows an example graph $\mathcal{G}$ and its adjacency matrix $A_{\mathcal{G}}$. By normalizing the columns of $A_{\mathcal{G}}$ so that their entries sum up to one, we obtain the Markov matrix $T_{\mathcal{G}}$ which, if viewed as adjacency matrix, gives rise to the graph $\mathcal{G}_T$ to its right. The widths of the arcs are proportional to the link weights encoded in $T_{\mathcal{G}}$, that is to the transition probabilities between the nodes they connect. The thicker an arc, the more likely a random walker will traverse it.

Squaring the matrix of transition probabilities $T_{\mathcal{G}}$ results in the second-order Markov matrix $T_{\mathcal{G}}^2$ and the associated second-order Markov graph $\mathcal{G}_{T^2}$, both shown in the third row in Figure 8.2 . The width of an arc leading form a node i to another node j reflects the probability of a random walk on the original graph $\mathcal{G}$ to reach node j at two steps from node i.

The transition from the first-order to the second-order Markov graph induces an interesting redistribution of transition probabilities: Some originally strong bonds (those between nodes 1 and 3, 1 and 4, 2 and 3, and 2 and 4) weaken. The previously unconnected nodes 1 and 2, on the other hand, develop strong ties to each other. According to Hypothesis 1, such pairs of nodes are good candidates for synonymy.

Nodes, such as 3 and 4, which are close first-order as well as second-order neighbors, are likely to correspond to members of the same semantic category. They occur not only with the same other words, but with each other, too.

A characteristic of the second-order Markov graph which stands out is that each node is connected to itself quite strongly via a self-loop. Nodes in the original graph, on the other hand, are not linked to themselves. In fact, this observation fits well with our hypothesis, as any word is naturally a synonym of itself.

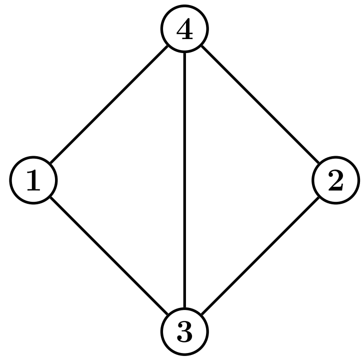
We now formalize Hypothesis 1. Let $T = T_{\mathcal{W}}$ denote the first-order, and $T^2 = {T_{\mathcal{W}}}^2$ denote the second-order Markov matrix associated with the word graph $\mathcal{W}$. The differences between 1-step and 2-step transition probabilities are encoded in the matrix $D := T^2 - T$. Since synonymy is a symmetric relationship, we expect a pair of synonyms w_i and w_j to show a considerable increase in transition probability in *both* directions between the corresponding nodes i and j. This means, we expect both of the entries d_{ij} and d_{ji} of D to assume large values.

The average increase in transition probability between each pair of nodes i and j is given by

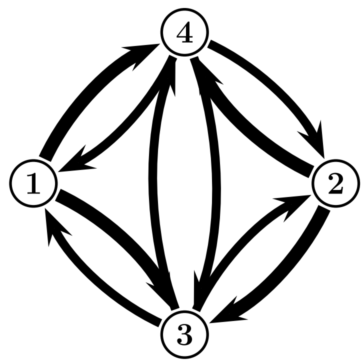
$$\bar{d}_{ij}\ (= \bar{d}_{ji}) \quad = \quad (1/2) * (d_{ij} + d_{ji}),$$

which assumes a value between -1 and 1, and corresponds to the (i, j)-element of the

$$A_{\mathcal{G}} \quad = \quad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 1 & 1 & 1 & 0 \end{array}$$



$$T_{\mathcal{G}} \quad = \quad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & .33 & .33 \\ 2 & 0 & 0 & .33 & .33 \\ 3 & .5 & .5 & 0 & .33 \\ 4 & .5 & .5 & .33 & 0 \end{array}$$



$$T^2_{\mathcal{G}} \quad = \quad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & .33 & .33 & .11 & .11 \\ 2 & .33 & .33 & .11 & .11 \\ 3 & .17 & .17 & .44 & .33 \\ 4 & .17 & .17 & .33 & .44 \end{array}$$

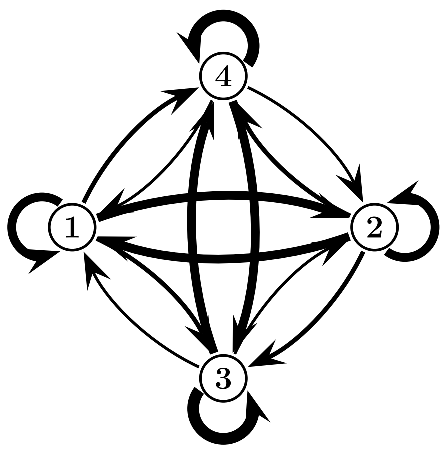


Figure 8.2: An adjacency matrix and its corresponding graph; the first-order Markov matrix and associated graph; the second-order Markov matrix and associated graph.

symmetric matrix

$$\bar{D} = (1/2) * (D + D^T) \tag{8.1}$$

where D^T is the transpose of D which is obtained by turning rows into columns and vice versa. Cells (i, j) of $\bar{D}$ with large values are indicative of a synonymy relationship between words w_i and w_j.

8.3 Evaluation experiment

In this section, we describe an experiment which tests the validity of our hypothesis.

8.3.1 Test set

The experiment is based on the same subgraph $\mathcal{W}'$ of the word graph $\mathcal{W}$ which was used in Chapter 7 for the extraction of idioms. $\mathcal{W}'$ consists of $n = 5,952$ nodes and $m = 62,184$ links, and the weight associated with a link is the log-likelihood score between the words it connects.

The link weights form the entries of the adjacency matrix $A_{\mathcal{W}'}$ from which we obtain the Markov matrix $T_{\mathcal{W}'}$ of first-order transition probabilities by normalizing the columns so that they each sum up to one.

We divide the set of all $\binom{5952}{2} = 17,710,176$ word pairs into two categories, synonyms and non-synonyms, where category membership is based on WordNet. In WordNet, synonymy information is captured in the WordNet synsets. A synset represents a certain concept and consists of its alternative names. Hence, two words are synonyms in WordNet if they are fellow members of a synset. For example, the following WordNet synsets encode the synonymy relations *pedestrian* $\overset{\text{syn}}{\sim}$ *walker*, *pedestrian* $\overset{\text{syn}}{\sim}$ *footer*, *walker* $\overset{\text{syn}}{\sim}$ *footer*, *footnote* $\overset{\text{syn}}{\sim}$ *footer* and *urbanization* $\overset{\text{syn}}{\sim}$ *urbanisation*.

{pedestrian, walker, footer}:
A person who travels by foot.

{footnote, footer}:
A printed note placed below the text on a printed page.

{urbanization, urbanisation}:
The social process whereby cities grow and societies become more urban.

Furthermore, we consider a word's derived forms [1], in particular singular and plural, to be synonymous.

$1,757$ of the $17,710,176$ word pairs fall into the category of synonyms. From these synonym pairs, we construct a test set of $2 * 1757 = 3,514$ test instances as follows. Each synonym pair (v, w) gives rise to two test instances. One in which v serves as the problem

[1] We obtain a word's derived forms using the `validForms` function of the WordNet::QueryData perl interface (`http://search.cpan.org/~jrennie/WordNet-QueryData/`).

word and w is the sought synonym (or the *target* as we shall call it), and one in which w is the problem word and v is the target. In what follows, we will refer to this test set as $Synonyms_{\mathrm{WN}}$.

Not all words which are synonyms in WordNet are necessarily synonyms in the BNC. *Gull* and *mug*, for example, both belong to the WordNet synset

{chump, fool, gull, . . . , mug}:
A person who is gullible and easy to take advantage of.

In the BNC, however, *gull* and *mug* assume meanings which are very different from the above synset and which are far from being interchangeable. This is illustrated in Figure 8.3 which shows the neighborhoods of the two words in the word graph.

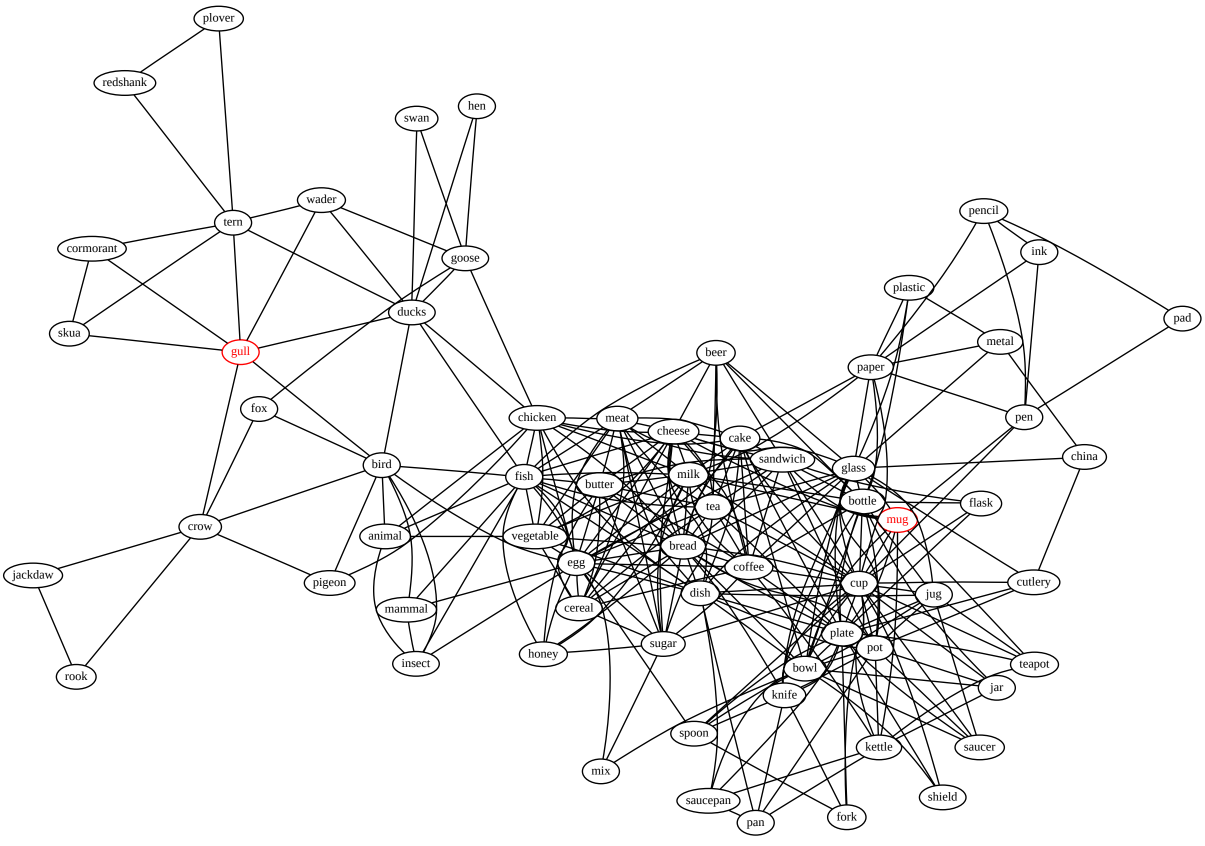


Figure 8.3: As the neighborhoods of *gull* and *mug* indicate, the two words have very different meanings in the word graph.

For this reason, we consider a second test set $Synonyms_{\mathrm{WN+LSA}}$ which is a subset of $Synonyms_{\mathrm{WN}}$ and consists of those problem-target pairs which, in addition to being WordNet synonyms, show high contextual similarity in the corpus. Contextual similarity is measured as the cosine similarity of the words' vectors in LSA space (cf. Section 9.2.4) and assumes values between 0 and 1. More precisely, for $\mathit{Synonyms}_{\mathrm{WN+LSA}}$ we drop all those problem-target pairs in the test set $\mathit{Synonyms}_{\mathrm{WN}}$ which have a cosine similarity of less than 0.2, since these WordNet-attested synonymy relationships seem to be inconsistent with the corpus. This results in a reduced test set $\mathit{Synonyms}_{\mathrm{WN+LSA}}$ which consists of $1,770$ problem-target pairs.

In the following section, we test whether and how well our method is able to reproduce the synonymy relationships between problem and target words.

8.3.2 Experimental set-up

The experiment is set up as follows. For each of the two test sets $\mathit{Synonyms}_{\mathrm{WN}}$ and $\mathit{Synonyms}_{\mathrm{WN+LSA}}$, and for each problem word therein, we compile a set of ten potential synonyms which includes the target, i.e. the "true" synonym. Given a problem word, our method provides a score for each candidate synonym which is a measure of the degree of synonymy between the problem word and the candidate. We sort the candidate synonyms in descending order of their scores and record the rank of the target. The highest scoring candidate has a rank of one, the lowest scoring a rank of ten.

The composition of the set of candidate synonyms determines the complexity of the task. We consider two different ways of choosing candidate synonyms, each of which gives rise to a different experiment:

1. Random selection (The candidate set consists of 9 words selected at random from the set of words represented in the word graph, plus the target.)

2. Selection based on contextual similarity to the problem word (The candidate set consists of the top 9 contextual neighbors of the problem word plus the target.)

Identifying the target in a set of randomly chosen words should be fairly easy. Candidates assembled according to selection criterion 2 all show a high degree of similarity to the problem word, and the task of tracking down the target becomes considerably harder.

We will denote the experiments associated with the two modes of choosing candidate words as $\mathit{Exp}_{\mathrm{random}}$ and $\mathit{Exp}_{\mathrm{LSA}}$ respectively.

Table 8.1 shows two of the test instances of both experiment $\mathit{Exp}_{\mathrm{random}}$ and experiment $\mathit{Exp}_{\mathrm{LSA}}$. The example test instances are all derived from the synonym pair (*consequence, outcome*). In experiment $\mathit{Exp}_{\mathrm{random}}$, the set of alternative words is an arbitrary sample of the words in the word graph which were chosen fully independently of the problem word. The set of alternatives is completed by adding the target, the last word in the list which is highlighted in italics. In experiment $\mathit{Exp}_{\mathrm{LSA}}$, the alternatives are contextual neighbors and thus semantically closely related to the problem word. Many of the choices are near-synonyms of the problem word which makes it much more difficult to spot the true synonym. In fact, besides the target at the end of the list, there are other valid synonyms among the alternatives (*effect* and *result* resp.). If our method picks out one of these alternative synonyms, we consider this just as good a choice as picking out the target.

$\boldsymbol{Exp}_{\textbf{random}}$	
Target	Candidate Synonyms
consequence	amphetamine postcard move invention ratio cart child fulfillment adult *outcome*
outcome	plumbing boats inspection hour pool frankincense loneliness fair honesty *consequence*

$\boldsymbol{Exp}_{\textbf{LSA}}$	
Target	Candidate Synonyms
consequence	impact *effect* change implication circumstance disruption uncertainty disaster behaviour *outcome*
outcome	*result* evaluation assessment effectiveness uncertainty conclusion participation trial circumstance *consequence*

Table 8.1: Two of the test instances used in experiments Exp_{random} (upper table) and Exp_{LSA} (lower table).

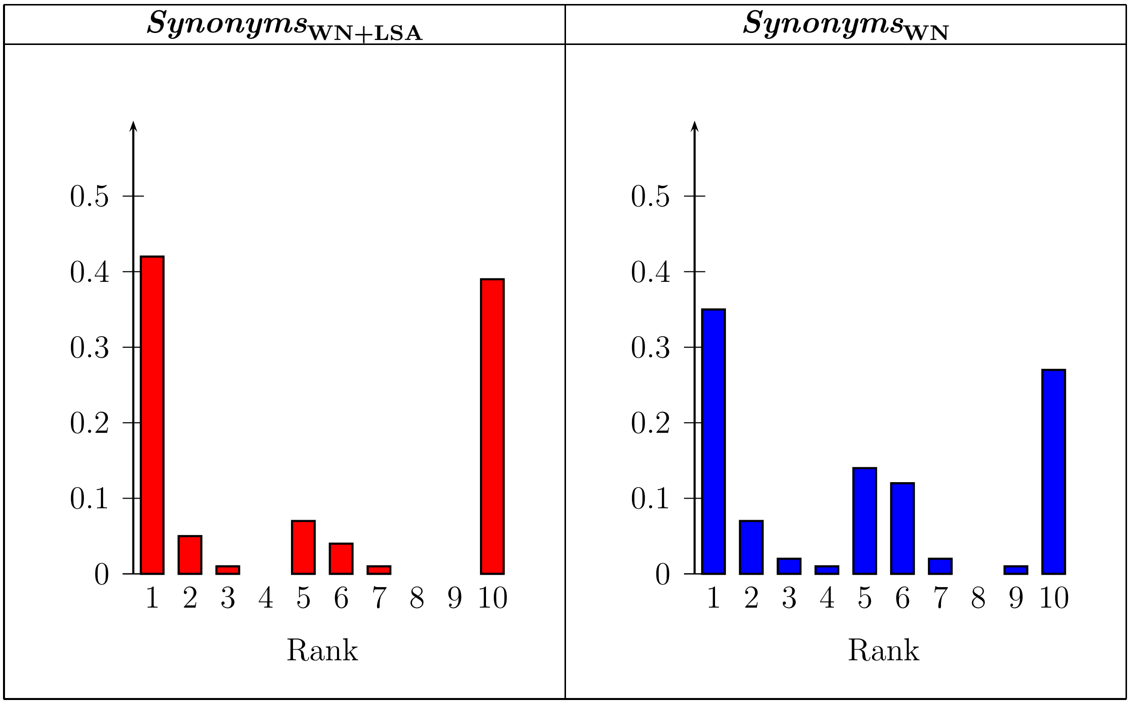
A word may be a problem word of more than one test instance. For example, *design* occurs in several of the WordNet-derived synonym pairs. Its synonyms are *invention*, *innovation*, *plan* and *pattern* each of which serves as the target of *design* in a test instance. Some of the test instances allude to different meanings of *design*. Since the LSA Space is a representation of words rather than of word senses, the set of alternative words in experiment Exp_{LSA} is the same for all test instances sharing the same problem word, *design* in this case. Synonymy is very sensitive to context. In this form, our approach to synonymy extraction doesn't take context into consideration. In Section 8.5, we will offer some ideas on how to make the synonymy measure context-sensitive.

We evaluated our approach on all possible combinations of test set and experiment type. Figure 8.4 shows the distribution of the target ranks. Performance is disappointingly poor.

Let us first inspect the evaluation results on test set $Synonyms_{\text{WN+LSA}}$ (first column). In the easier task of tracking down the true synonym from among *randomly* chosen candidates (setup Exp_{random}), the target was the highest scoring candidate in more than 40% of the test instances. However, there are almost as many cases in which the target received the *lowest* score of all candidates. Only very few target ranks fall in the middle of the range. We expected to see a rank distribution which is concentrated near the beginning, and which decreases more or less rapidly towards the end of the rank range. Instead, the bulk of the distribution agglomerates at both extremes of the range. While less pronounced, the distribution of target ranks in experiment Exp_{LSA} shows qualitatively the same behavior.

The test instances fall into three major categories, direct hit, complete failure and undecidable test problem, which correspond to the big peak at the lower end of the range, the big peak at the upper end of the range, and the small one in the middle. The category

$\mathbf{Exp}_{\mathbf{random}}$



$\mathbf{Exp}_{\mathbf{LSA}}$

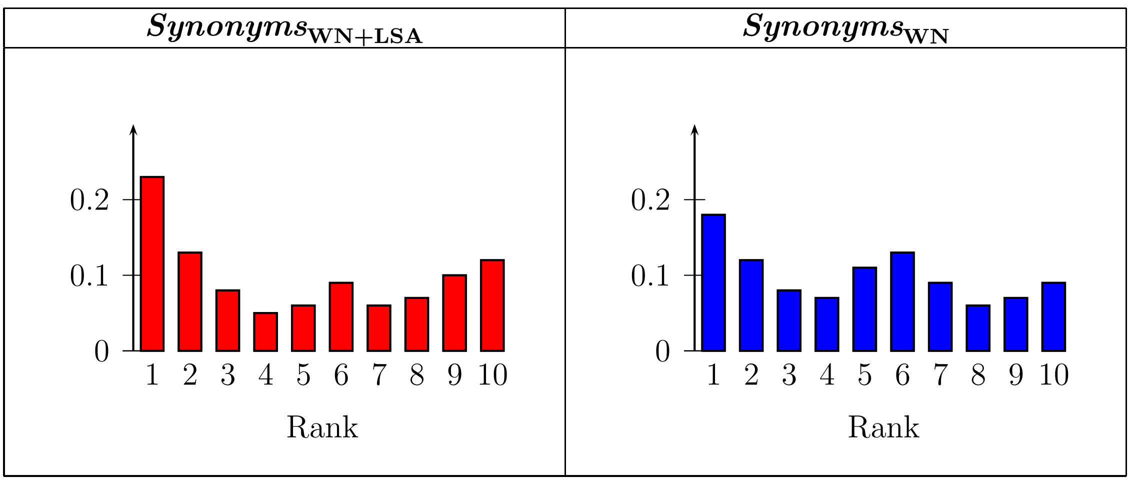


Figure 8.4: Histogram of the target ranks for each of the four experimental setups.

of undecidable test instances is composed of those test instances for which all candidates, including the target, received the same or similar scores. Candidates with the same score each receive the average rank. In particular, if all candidates score equally highly, they all receive the medium rank of 5.5.

We now compare these results with those obtained on the bigger test set $Synonyms_{\mathrm{WN}}$ (right column in Figure 8.4). The target rank distributions are very similar to the corresponding distributions in the left column, except that some of the distributional mass at the ends of the range has flown into the middle peak. This means there is a bigger number of test instances in which the target is indiscriminable from the other candidates. We suggest that the bigger proportion of undecidable test problems in test set $Synonyms_{\mathrm{WN}}$ can be explained by those problem-target pairs which, though they are synonyms in WordNet, yet, do not show this synonymy in the corpus, and therefore seem just as arbitrary as the other alternatives.

To understand the poor performance of our method, let us have a look at the test instances for which the target was the least scoring choice. In some of these cases, the problem word and the target, although being members of the same WordNet synset, are not entirely exchangeable, but rather near-synonyms, such as *bag* and *suitcase* or *picnic* and *outing*. Near-synonyms, naturally do occur in lists together, and our synonymy measure based on Hypothesis 1 rightly recognizes that these words are not genuine synonyms.

Another frequent cause of links between synonyms is ambiguity. Synonymy is a relationship between word senses rather than between words. Cooccurrences of synonymous words in which they both refer to the *shared* sense are rare, but synonyms may well cooccur if they refer to *different senses*. For example, *pond* and *pool* both refer to a small lake. But *pool* may also denote a swimming pool. And chances that *pool* "swimming pool" and *pond* "small lake" occur together in a list are quite high. Such cases are extremely frequent among words which share the same systematic polysemy, and are synonyms in one of their shared senses. The words *apricot* and *peach*, and *chocolate* and *coffee*, for example, are non-synonymous as fruits or food, but they are synonyms for the same color. Similarly, *rabbit* and *hare* are names of different animals, yet they are names for the same meal. In its current form, our approach is extremely vulnerable to such cases of regular polysemy coupled with synonymy. We expect that performance can be improved by taking context into account (cf. Section 8.5).

But even apart from the above listed cases, very often, the target word scored least although it seems to be a perfect synonym of the problem word. Going back to the corpus, we find that, contrary to our intuition, in fact, many synonyms cooccur in lists surprisingly often. These cooccurrences of synonyms comprise conventionalized expressions such as "(under the) terms and conditions" and "(The) aim and purpose of", but also many free coordinations:

- *Consequence in the form of <u>penalty and punishment</u> is the subject of the next chapter.*
- *. . . and <u>craftsmen and artisans</u> have reached as far south as Lisbon by the mid-twelfth century.*

- *Rob is full of stories and tales of fish landed and fish lost ...*
- *... the continuing process of adaptation and adjustment without which ...*
- *Mr Morraine spent most of his time dusting his collection of strange-looking rocks and stones ...*
- *Even if a non-verbal visual form of representation is not in the form of an implicit picture or image ...*

In particular, we observe a considerable number of "or"-coordinated synonyms. Coordinations, such as

- *The words Muslim or Moslem refer to the practitioners of Mohammed's religion ...*
- *Within the chest, or thorax, lie the lungs, which take in oxygen from the air and expel carbon dioxide.*

are sometimes even used to explicitly express synonymy between the constituent words.

Another reason for the often strong connections between synonyms in the word graph is modification. Synonyms, once modified, co-occur in lists quite a lot. The coordinations

- *death penalty and corporal punishment*
- *high-voltage power lines or supply cables*
- *college accommodation office or citizens advice bureau*,

for example, are far from being redundant. Though the head nouns of the coordinands are synonyms, the coordinands themselves aren't. The way we assemble the word graph, we (crudely) drop the modifying adjectives and nouns (cf. Section 3.3) and link the head nouns, which results in the following links for the examples given above: (*penalty, punishment*), (*line, cable*) and (*office, bureau*).

Most importantly, however, genuine synonyms hardly ever exist. Even the most similar words show some (if only subtle) difference in meaning or connotation, and it is almost always possible to come up with a context in which they are not fully interchangeable.

All in all, the assumption made in Section 8.1 that synonyms do not occur together in lists seems to be far too restrictive. Synonyms sometimes cooccur and sometimes don't, but they do tend to cooccur with the same words.

8.4 Revised Hypothesis

We therefore set up the following revised hypothesis on how synonymy is encoded in the topology of the word graph:

Hypothesis 2 *Synonyms correspond to pairs of nodes which may or may not be directly linked, but which are strongly connected via shared neighbors.*

Based on this more moderate hypothesis, we define a new measure of synonymy. Instead of the difference between second order and first order transition probabilities, we now simply use the second order transition probabilities between words to measure their degree of synonymy. Formally, the degree of synonymy of two words (w_i, w_j) is expressed as

$$Syn(w_i, w_j) = 0.5 * (t_{ij}^{(2)} + t_{ji}^{(2)}) \tag{8.2}$$

which is the (i, j) entry of the matrix $\frac{1}{2} * (T^2 + (T^2)^T)$ and represents the average two-step transition probability between the nodes n_i and n_j representing w_i and w_j in the graph.

We will now investigate whether we can improve synonymy recognition using the new synonymy measure. We test the new measure following the same evaluation process. Results are shown in Figure 8.5.

The new synonymy measure yields considerably better results. In experiment setting Exp_{random}, the previously large number of complete misses at the end of the rank range has been fully absorbed by the category of direct hits (rank=1) which accounts for nearly 80% of all test instances. We observe a similar redistribution of ranks for the more difficult task Exp_{LSA}. Although in this case, some of the mass at rank 10 remains, the rank distribution tails off towards the end of the range. 28% of direct hits may not seem like a lot, but considering the difficulty of the task, and the fact that we evaluate against WordNet which itself shows some inconsistency in the synonymy relationship, the results are encouraging.

The effect of dropping the assumption that synonyms are weak first-order neighbors is illustrated in Table 8.2. The upper table shows some example rankings of the candidate synonyms associated with a problem word. The target, together with all other synonyms in the candidate list are highlighted in bold font. Whereas the initial synonymy measure correctly and directly identifies the synonymy relationship between *scene* and *setting*, and *consequence* and *effect*, it does really bad on most of the other example problems. The picture changes, however, if we use the revised synonymy measure to rank the candidates. The targets, as well as the other synonyms among the candidates move towards the beginning of the list, occupying ranks between 1 and 3.

tabcolsep3pt

8.5 Conclusion

The initial idea of looking for words which are distant in terms of first-order but close in terms of second-order cooccurrence had to be revised as, contrary to our intuition, synonyms cooccur not only with the same words but with each other, too. The method based on this assumption works well in recognizing absolute synonymy relationships, such as between different spellings of the same word ((*moslem, muslim*), (*urbanization, urbanisation*)), derivational variants ((*cut, cutting*), (*advertisement, advertising*)), singular and plural forms ((*bookshop, bookshops*), (*lady, ladies*)), and full forms and their abbreviations ((*gym, gymnasium*), (*fridge, refrigerator*)). However, fully equivalent words are rare.

We found that for tracking down close synonyms as well, we need to relax our assumptions about first-order cooccurrence of synonyms. Mere second order cooccurrence

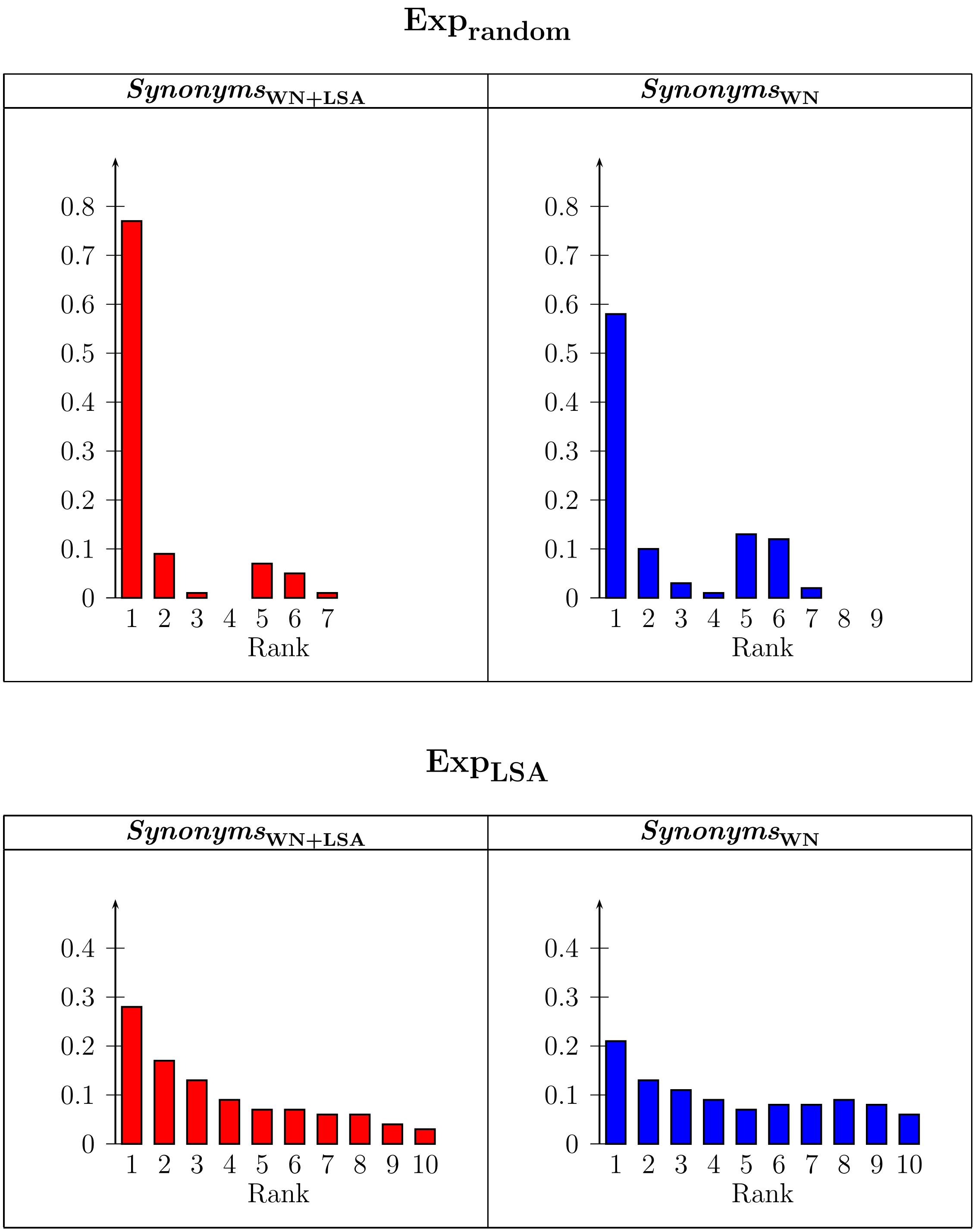


Figure 8.5: Histogram of the target ranks for each of the four experimental setups obtained using the revised synonymy measure.

Initial synonymy measure								
Rank	**Problem word**							
	bag	suitcase	scene	setting	consequence	outcome	disability	impairment
1	tray	basket	**setting**	**scene**	**effect**	participation	bereavement	disorder
2	jar	tray	image	objective	impact	**result**	mobility	depression
3	**handbag**	drawer	drama	emphasis	circumstance	circumstance	retirement	diarrhoea
4	tin	handbag	film	aim	uncertainty	assessment	carers	stress
5	envelope	desk	comedy	guideline	disaster	evaluation	pensioner	fatigue
6	drawer	armchair	cinema	layout	disruption	conclusion	people	schizophrenia
7	bottle	cardboard	movie	monitoring	change	uncertainty	age	personality
8	packet	sofa	offence	smoke	behaviour	trial	illness	illness
9	**suitcase**	raincoat	opera	style	**outcome**	effectiveness	**handicap**	**handicap**
10	box	**bag**	cameras	task	implication	**consequence**	**impairment**	**disability**

Revised synonymy measure								
Rank	**Problem Word**							
	bag	suitcase	scene	setting	consequence	outcome	disability	impairment
1	**suitcase**	**bag**	**setting**	style	**effect**	**consequence**	**handicap**	illness
2	box	basket	image	**scene**	impact	participation	**impairment**	**disability**
3	tray	tray	drama	objective	**outcome**	**result**	illness	**handicap**
4	jar	drawer	film	emphasis	implication	circumstance	age	disorder
5	**handbag**	handbag	comedy	task	circumstance	assessment	bereavement	depression
6	bottle	desk	opera	aim	change	effectiveness	mobility	diarrhoea
7	tin	armchair	cinema	guideline	behaviour	evaluation	retirement	stress
8	envelope	cardboard	movie	layout	uncertainty	conclusion	carers	fatigue
9	drawer	sofa	offence	monitoring	disaster	uncertainty	people	schizophrenia
10	packet	raincoat	cameras	smoke	disruption	trial	pensioner	personality

Table 8.2: Examples of synonym rankings using the initial (upper table) and the revised measure of synonymy (lower table).

strength (as measured by second-order transition probabilities) proved to be the more adequate measure of (near-)synonymy.

A problem with our approach to learning synonyms is its vulnerability to polysemy. A way out may be to look only for synonyms within clusters of similar words as obtained by the curvature and link graph clustering approaches (cf. Sections 5.6 and 6.1). This would mean that we apply our technique to each of the subgraphs which correspond to the members of a cluster and their links within the cluster.

We expect that better results could also be obtained by enriching the topological information of the word graph with contextual information as, for example, provided by Latent Semantic Analysis. More specifically, it is conceivable to compute a vector representation of each link in the word graph from the contexts it was observed in. This would allow us to consider only links related to a certain topic or context, and thus to extract context-sensitive synonyms.

Chapter 9

Related Work

9.1 Extraction of semantic relations

Several people have used lexical patterns to extract semantic relationships from text. Some of these approaches are described in this section.

Automatic acquisition of hyponyms from large text corpora: The first major study on the acquisition of hyponymy relations from text was that of Hearst (1992). She discovered that certain syntactic constructions, which she calls lexico-syntactic patterns, are reliable indicators of a hyponymy relationship between words in the text. Consider the following example

> *Make a final sowing of an early carrot, such as Amsterdam Forcing, Early Nantes, Almoro or Bertop, for pulling as baby carrots at around Christmas time .*

Even if we have never encountered the terms *Amsterdam Forcing* or *Early Nantes* before, this sentence makes us conclude that they are kinds of early carrots. Hearst identified several other patterns, for instance

- y, such as $x_1, \ldots, x_{n-1}$ and/or x_n
- y, including $x_1, \ldots, x_{n-1}$ and/or x_n
- $x_1, \ldots, x_n$ and other y

which all indicate that the x_i are kinds of y. In addition, Hearst suggests that new patterns can be learned automatically by the following bootstrapping algorithm:

1. Start with a set of word pairs which exhibit a clear hyponymy relationship.
2. Look in the text for places where hypernym and hyponym occur close to each other and remember the syntactic context.
3. Examine the set of contexts. Groups of similar syntactic patterns give rise to new patterns which can be used to extract more hyponymy relationships.
4. Add the newly discovered hyponymy relationships to the set of known hypernym-hyponym pairs and go to Step 2.

Automatic Acquisition and Expansion of Hypernym Links: Prométhée (Morin and Jacquemin, 2004) is a supervised system which extracts conceptual relations between terms. It is built on Hearst (1992)'s work on automatic extraction of hypernym links using lexico-syntactic patterns. The process of learning conceptual relations can be outlined as follows:

1. Select a conceptual relation, e.g., hypernymy.
2. Collect a list of term pairs linked through the selected relation, either manually or by using a thesaurus or another knowledge base.
3. Find sentences in which these term pairs occur and represent the sentences as lexico-syntactic expressions.
4. Produce candidate lexico-syntactic patterns by clustering similar lexico-syntactic expressions.
5. Let an expert validate the candidate lexico-syntactic patterns.
6. Use new patterns to extract more pairs of candidate terms.
7. Let the expert validate the pairs of candidate terms, and go to Step 3.

To reduce human involvement, the developers suggest feeding Prométhée with the output of an unsupervised relation extraction system instead of manually collecting pairs of semantically related terms in Step 2. Unsupervised relation extraction systems usually identify relations between terms based on co-occurences in sentences or documents using association measures such as mutual information and log-likelihood. Whereas they achieve high coverage in identifying semantically related words, precision is low. However, the pairs of semantically related words are well-suited as input to a supervised system such as Prométhée which is able to sort out noisy pairs of terms.

Finkelstein-Landau and Morin (1999) designed a technique for the expansion of links between single-word terms to links between multi-word terms. For example, given a link between *apple* and *fruit*, a similar link between *apple juice* and *fruit juice*, and between *apple growing* and *cultivation of fruit* is infered.

Two phrases $p_1(w_1, w_2)$ and $p_2(w'_1, w'_2)$ containing the content words w_1, w_2 and w'_1, w'_2 respectively are considered to be semantic variants of each other if

- w_1 and w'_1 are head words and w_2 and w'_2 are arguments with similar thematic roles.
- w_1 and w'_1 and/or w_2 and w'_2 are semantically related through a relation S (e.g., synonymy, hypernymy). The non-semantically related words are either identical or morphologically related.
- $p_1(w_1, w_2)$ and $p_2(w'_1, w'_2)$ are linked through the same semantic relation S.

Morin and Jacquemin use a set of predefined syntactic patterns describing the possible syntactic realizations of semantic variants, and morphological and semantic relatedness between the single-word terms w_i and w'_i to establish links between the multi-word terms $p_1(w_1, w_2)$ and $p_2(w'_1, w'_2)$. For example, the syntactic pattern $(w_2 w_1) \Rightarrow (w'_1 PrepDet?Adj* w'_2)$ describes the variation between *apple growing* and *cultivation of fruit*. Since *apple* and *fruit* are related via the hyponym relationship and *growing* and *cultivation* are synonyms, it can be infered that *apple growing* is a kind of *cultivation of fruit*.

Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction: Cederberg and Widdows (2003) follow Hearst (1992)'s approach of using lexico-syntactic patterns to automatically acquire hyponymy relations from text. Latent Semantic Analysis (LSA) is used to filter results which involves an increase in precision. Hyponymy relations non-attested in the corpus are deduced from noun coordination information, leading to improved recall.

The lexico-syntactic patterns identified in (Hearst, 1998) are used to extract hypernym-hyponym candidate relations. The list of possible relations is sorted according to a context-based similarity score between the candidate hypernym and hyponym. The similarity score used is based on a vector model, as commonly used in Information Retrieval, in which each word is represented by a vector of co-occurrence counts of the word in question with a set of content-bearing words. This high-dimensional vector space is then reduced using LSA and similarity between words is computed as the cosine of the angle between the corresponding vectors in the reduced space (Deerwester et al., 1990). The top ranking hypernym-hyponym pairs can be considered valid relations, since not only do the terms appear in a construction which frequently indicates a hyponymy relation, but the high LSA similarity, indicating high contextual similarity of the two terms, further enhances the plausibility of the candidate relation.

Only part of all correct hypernym-hyponym pairs detectable in the corpus actually occur in the constructions identified by Hearst. To deduce hypernym-hyponym relations not covered by these patterns, Cederberg and Widdows use noun coordination information. Nouns which co-occur in lists are often semantically similar and tend to originate not only from the same semantic field but even from the same ontological level, e.g., there are comparably many more instances of "fruit and vegetables" and "apples and pears" to be found in a corpus than instances of "apples and vegetables" or "carrots and fruit", etc.

From the hypernym-hyponym pairs (p, c) which passed the LSA filter, a clustering algorithm (Widdows and Dorow, 2002) based on co-occurrences of nouns in lists is used to find nouns $\tilde{c}$ which are similar to the hyponym c. These neighbors are used to infer new hypernym-hyponym relations $(p, \tilde{c})$, namely the pairings of the hypernym p with each of the neighbors $\tilde{c}$ of the hyponym c.

Because of ambiguity, the hyponym neighbors $\tilde{c}$ often do not correspond to the meaning of the hyponym c in the original hyponym-hypernym pair (p, c), e.g., *mass* co-occurs in lists with *length*, *weight*, etc., but inferring *length* IS-A *religious service* from the correct relation *mass* IS-A *religious service* results in an invalid relation. To avoid derivation of incorrect relations $(p, \tilde{c})$, once again, LSA is applied to filter out relations for which the LSA similarity between the hypernym p and the potential new hyponym $\tilde{c}$ falls below a certain threshold.

In cases where the hypernym p consists of several words, Cederberg and Widdows first determine the noun n_p in the compound whose meaning is most closely related to the meaning of the hyponym c, and use only n_p instead of the entire compound p for filtering. For example, in the relation *mass* IS-A *religious service*, *religious* is more predictive of the meaning of *mass* than is *service*. Predictivity is measured using LSA similarity. That is, filtering proceeds as follows:

- Compute LSA similarity between each noun in the multi-word hypernym p and the original hyponym c.
- Choose the noun n_p in the compound which scores highest.
- To filter the derived potential hypernym-hyponym pairs $(p, \tilde{c})$, compute the LSA similarity between each of the potential new hyponyms $\tilde{c}$ and n_p. Those pairs $(p, \tilde{c})$ for which this similarity exceeds a certain threshold are deemed valid relations.

Since *length* and *religious* are not contextually similar the invalid relation *length* IS-A *religious service* is eliminated.

9.2 Semantic similarity and ambiguity

9.2.1 Bootstrapping semantic categories

This section reviews several studies which aim at bootstrapping categories of similar words from a set of known "seed" category members.

A Corpus-Based Approach for Building Semantic Lexicons: Riloff and Shepherd (1997) propose a method for growing semantic categories given a small set of seed category members. The algorithm relies on the assumption that the members of a category are found in close vicinity of each other in text. Category members are assembled using the following bootstrapping algorithm:

1. Start with a set of seeds S of known category members.
2. Search the text for occurrences of the seed words, and collect narrow contexts around each occurrence. Pick out the nouns appearing in these contexts. They are candidates for new seed words.
3. Compute the affinity of each candidate n to the set of category seeds S using Equation 9.1.
4. Add the k top scoring seed candidates $n_1, \ldots, n_k$ to the seed list: $S = S \bigcup \{n_1, \ldots, n_k\}$, and go to Step 2.

Affinity of a candidate noun n to the category, which is represented by the current set of seeds S is computed in the following way:

$$\mathrm{Aff}(n) = \sum_{c \in C(S)} \frac{f(n,c)}{f(n)} \tag{9.1}$$

where

- $C(S)$ is the set of all contexts of seeds $s \in S$

- $f(n)$ is the frequency of the noun n in the corpus
- $f(n, c)$ is the frequency of n in context c.

Bootstrapping is stopped after a certain number of iterations, and the seed list S is used to produce a list of candidate category members, which is sorted by affinity to S. The list is then given to the user who decides which of the nouns in the list are true category members.

Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction: Roark and Charniak (1998) adopt Riloff and Shepherd (1997)'s approach to learning semantic categories. However, they obtain more coherent results by

- Using more structured contexts for collecting seed candidates
- Using more elaborate statistics for selecting new seed words and for ranking candidate category members.

Rather than considering all nouns in the vicinity of a seed for seed candidates, the authors focus on very specific noun constructions, namely conjunctions ("cars and trucks"), coordinations ("man, woman and child"), appositives ("The African elephant shrew, a highly-strung insect-eating mammal") and compound nouns ("panther cat"), which are often composed of nouns belonging to the same semantic category.

Roark and Charniak state that different statistics are needed for seed word selection and the final ranking of category members. New seed words are selected using a statistic which is very similar to the one used by Riloff and Shepherd:

$$\mathrm{Aff}(n) = \sum_{s \in S} \frac{f(n, s)}{f(n, *)} \tag{9.2}$$

where

- $f(n, s)$ is the number of times the candidate noun n and the seed s jointly appear in one of the earlier listed noun constructions
- $f(n, *)$ is the total number of cooccurrences of noun n with any word in such a noun construction.

Since this measure tends to favor low frequency words over more frequent ones, Riloff and Shepherd had to employ a minimum frequency cut-off, which does not only eliminate noise, but also disposes of a lot of valid category members. Rather than using a minimum frequency cut-off, Roark and Charniak propose to properly deal with these low frequency words in the final ranking.

Whereas Riloff and Shepherd use the same statistic for both tasks (seed word selection and final ranking), Roark and Charniak use the log-likelihood measure (Dunning, 1993) to decide which of the final candidates are likely to be true category members. The log-likelihood measure has been reported to accurately predict the unexpectedness of cooccurrence counts of low to medium frequency words. Therefore, those low frequency words which were erroneously added to the seed list during the bootstrapping procedure, are demoted in the final candidate list.

The authors conclude that each of the two statistics is well suited for one of the tasks but not the other. The empirical probability of Equation 9.2 accurately handles high frequency words. This is very important in the process of growing the list of seed words. Adding a highly frequent incorrect word to the seed list is particularly dangerous, as it will cause lots of other non-category members to become seeds and contaminate the category. Wrongly included low frequency words, on the other hand, don't do as much harm. However, for the final step of ranking category candidates, it is important to detect erroneous low frequency words which sneaked into the seed list because of the empirical probability's tendency to promote low frequent words.

A Graph Model for Unsupervised Lexical Acquisition and Automatic Word-Sense Disambiguation: The algorithm described in (Widdows and Dorow, 2002) has the same basic structure as the approaches of Riloff and Shepherd (1997) and Roark and Charniak (1998). Just as in the two preceding algorithms, a semantic category is grown around a set of seed words by incrementally selecting new seeds based on their affinity to the current set of seeds.

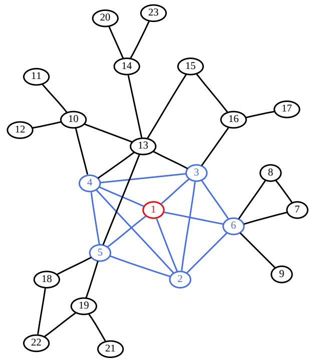
The novelty of the algorithm is the way new seed words are selected. To measure affinity of a word to a set of seeds, the authors rely on a word graph. Affinity is defined as the degree to which a word's neighborhood in the graph lies within the neighborhood of the current set of seeds.

The authors, too, focus on nouns cooccurring in lists, assuming that nouns in lists have similar properties. To build a word graph, lists of nouns are extracted from the corpus. Then, a graph is built by drawing a node for each noun and establishing a link between any pair of nouns which form part of a list. In each iteration of the incremental algorithm, each word in the neighborhood of the current set of seeds is considered for addition to the seed list[1]. The affinity of a candidate seed w to the current set of seeds S is defined to be the percentage of its links which lie within the neighborhood $N(S)$ of S:

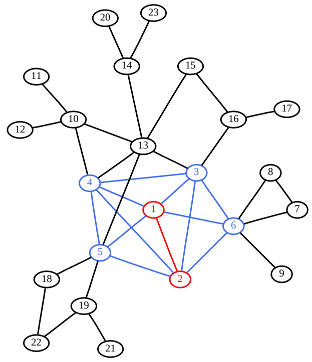
$$\mathrm{Aff}(w) = \frac{|N(w) \cap N(S)|}{|N(w)|} \tag{9.3}$$

The picture series in Figure 9.1 illustrates the algorithm. Bootstrapping is launched using node 1 as seed of the semantic category to be assembled. As can be seen, the more a candidate seed's links point away from S, the less likely it is added to the seed list, making the algorithm particularly robust to what Roark and Charniak name "infection" (the process of the seed list drifting away from the intended category due to the inclusion of a spurious or ambiguous word).

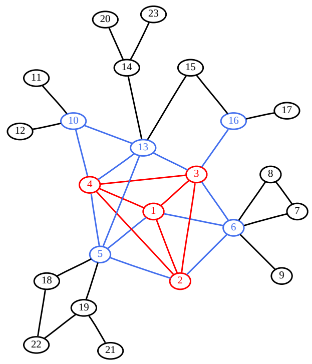
[1]The neighborhood $N(u)$ of a node u in the graph is the set of nodes which are directly connected to u. The neighborhood of a set of nodes is simply the union of the individual neighborhoods.



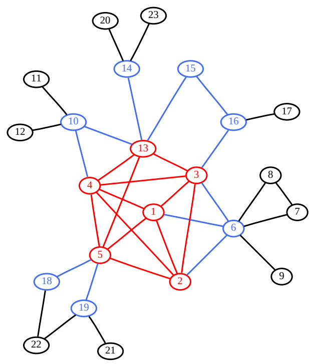
node	affinity
2	**1**
3	2/3
4	2/3
5	1/2
6	1/2



node	affinity
3	**2/3**
4	**2/3**
5	1/2
6	1/2



node	affinity
5	**2/3**
6	1/2
10	1/2
13	**2/3**
16	1/3



node	affinity
6	1/2
10	1/2
14	1/3
15	**1**
16	2/3
18	1/2
19	1/3

Figure 9.1: Bootstrapping a semantic category in the word graph. Red is used to indicate the current cluster, blue marks the cluster's neighborhood.

9.2.2 Hard clustering of words into classes of similar words

We distinguish two types of approaches for extracting semantic categories from text: Hard and soft clustering methods. Hard clustering approaches allow words to belong to a single cluster only. Soft clustering approaches, on the other hand, allow for multiple cluster membership. This section describes some of the hard clustering techniques.

Automatic construction of a hypernym-labeled noun hierarchy from text: Instead of just clustering nouns into classes of similar nouns, Caraballo's algorithm organizes the nouns in a text in the form of a hierarchy of hyponymy relationships. She adopts the working hypothesis of Roark and Charniak (1998) that nouns which occur together in coordinations, conjunctions or appositive structures are semantically related.

All cooccurrences of nouns within these constructions are collected in a cooccurrence matrix C. The row (resp. column) vectors of the matrix contain the cooccurrence counts of a noun with all other nouns, and act as semantic profiles.

Caraballo computes the similarity between any two of the nouns using the cosine similarity measure[2]. Replacing the cooccurrence counts in the cooccurrence matrix with the cosine similarities results in a matrix S which consists of all noun-noun similarities[3]. The following agglomerative clustering procedure is used to arrange the set of nouns in a hierarchy:

1. Initially, each noun forms a (leaf) node in the hierarchy.
2. Select the highest entry $S(i,j) = sim(n_i, n_j)$ of the similarity matrix S.
   Introduce a new node n_{ij} which subsumes nouns n_i and n_j.
3. If all nodes have been assembled under a common root node, stop.
   Otherwise, update S using Formula 9.4 and return to Step 2.

Let us assume that a newly introduced node n_{AB} resulted from merging two nodes n_A and n_B. The similarity between the new node n_{AB} and another node n_C (which is not subsumed by n_{AB}) is defined to be a weighted average of the similarities $sim(n_A, n_C)$ and $sim(n_B, n_C)$ of the merged nodes n_A and n_B, and n_C

$$sim(n_{AB}, n_C) = \frac{w(n_A) * sim(n_A, n_C) + w(n_B) * sim(n_B, n_C)}{w(n_A) + w(n_B)} \tag{9.4}$$

where $w(n)$ is the number of descendants of node n.

Figure 9.2 illustrates the agglomerative clustering process by means of a hypothetical similarity matrix. To get the idea of how the similarity matrix S is updated, let us use Formula 9.4 to recompute similarities after node n_{123} has been introduced. Node n_{123} is

[2]The cosine measure is given by the scalar product of the vectors after they have been normalized. It is the cosine of the angle enclosed by the two vectors.

[3]If the rows of the cooccurrence matrix C are normalized, then the similarity matrix S is simply the product of C with itself: $C^2 = C^T C = (c_i^T c_j)_{i,j} = (< ci, cj >)_{i,j} = (sim(n_i, n_j))_{i,j} = S$.

the result of merging nodes n_{12} and n_3. Hence, the similarity of the new node n_{123} and node n_4, for example, is

$$\begin{aligned} sim(n_{123}, n_4) &= \frac{w(n_{12}) * sim(n_{12}, n_4) + w(n_3) * sim(n_3, n_4)}{w(n_{12}) + w(n_3)} \\ &= \frac{2 * 0.15 + 1 * 0.3}{2 + 1} \\ &= 0.2 \end{aligned}$$

Similarity Matrix S **Noun Hierarchy**

	1	2	3	4	5
1	1.0	$\boxed{0.8}$	0.6	0.1	0.2
2	0.8	1.0	0.7	0.2	0.3
3	0.6	0.7	1.0	0.3	0.4
4	0.1	0.2	0.3	1.0	0.6
5	0.2	0.3	0.4	0.6	1.0

	12	3	4	5
12	1.0	$\boxed{0.65}$	0.15	0.25
3	0.65	1.0	0.3	0.4
4	0.15	0.3	1.0	0.6
5	0.25	0.4	0.6	1.0

	123	4	5
123	1.0	0.2	0.3
4	0.2	1.0	$\boxed{0.6}$
5	0.3	0.6	1.0

	123	45
123	1.0	$\boxed{0.225}$
45	0.225	1.0

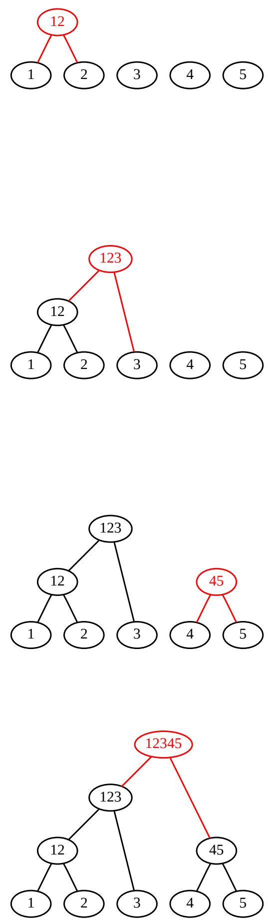


Figure 9.2: The Agglomerative Clustering approach.

The coordination patterns which Caraballo considers when counting cooccurrences include constructions such as $n_1, \ldots, n_k$ and other n from which she concludes that n is likely to be a hypernym (superordinate term) of the nouns n_i. She exploits this information for the purpose of labeling the nodes in the noun hierarchy.

Labeling of the tree is done in a bottom up fashion, starting with the leaves first. For each leaf node, a binary vector is compiled which specifies whether a noun has been seen as a hypernym of the leaf (1) or not (0). For labeling the internal nodes of the tree, the vectors associated with its children are summed up. The largest entry (entries) of a node are then chosen as node label(s). Due to the way labels are assigned, it may happen that the label of an ancestor is not a hypernym of the label of a descendant.

Caraballo states that the resulting noun hierarchy is highly redundant, which is mainly due to its binary structure. To collapse similar nodes, Caraballo proposes to summarize subtrees with uniform node labels under a single superordinate node.

9.2.3 Soft clustering of words into classes of similar words

An advantage of soft clustering approaches over hard clustering methods is their ability to assign words to several clusters if necessary. The clusters a word belongs to can be regarded as being representative of its different senses. This section describes some of the soft clustering approaches.

Distributional Clustering of English Words: Pereira et al. (1993) use soft distributional clustering to group words into similarity classes with the aim of overcoming data sparseness. The behavior of a rare word can then be predicted from the average behavior of the words similar to it.

The authors focus on clustering nouns according to the verbs with which they appear in verb-direct object relations. These relations are extracted from a parsed corpus as verb-noun pairs (v, n) so that n is direct object of v. Each noun n is considered to be a probability distribution P_n over the set of verbs V:

$$P_n(v) = P(v|n), v \in V \tag{9.5}$$

$P_n(v)$ can be interpreted as the probability that, given that we see noun n in a verb / direct object relationship $(\tilde{v}, n)$, $\tilde{v}$ coincides with v.

The P_n can be regarded as points in a similarity space which consists of all distributions over the set of verbs V. The authors use the Kullback-Leibler (KL) divergence $D(.,.)$, a commonly used measure to compare probability distributions, to measure similarity (or rather dissimilarity) in the similarity space.

The dissimilarity between two nouns n_1 and n_2 then corresponds to the KL divergence $D(P_{n_1}, P_{n_2})$ of the corresponding distributions P_{n_1} and P_{n_2} which equals

$$D(P_{n_1}, P_{n_2}) = \sum_{v \in V} P_{n_1}(v) log(\frac{P_{n_1}(v)}{P_{n_2}(v)}) \tag{9.6}$$

and which measures the loss of information in using the distribution P_{n_2} instead of P_{n_1} to describe n_1.

The goal is to find a set C of similarity clusters with centroids $P_c, c \in C$ (corresponding to points in the similarity space) and cluster membership probabilities $P(c|n)$ so that $P_n(v)$ can be decomposed as follows:

$$P_n(v) = P(v|n) = \sum_c P(c|n) P_c(v), \quad v \in V \tag{9.7}$$

That is, each noun n is a weighted average of the clusters it belongs to, the weights being the affinity $P(c|n) = P(n \in c)$ with which n belongs to the cluster. Conversion of Equation 9.7 (using the definition of conditional probability) yields

$$P_n(v) = P(v|n) = \sum_c P(n|c) P(v|c) P(c) \tag{9.8}$$

This amounts to the assumption that within a cluster c, the events v and n are conditionally independent, i.e., v and n are associated only through the clusters c.

The $P(c|n), c \in C, n \in N$ and the $P(v|c), v \in V, c \in C$ are the unknown model parameters which have to be estimated. We want to estimate the model parameters in such a way that the model best fits the observed data, that is the average distance of the $n \in N$ to the clusters $c \in C$, taking into account the membership probabilities $P(c|n)$, should be minimal:

$$D = \sum_n \sum_c P(c|n) \;\; D(P_n, P_c) \longrightarrow \min \tag{9.9}$$

Without any further restrictions, an optimal clustering is obtained by letting each noun constitute a cluster by itself (i.e., $C = N$) and by assigning each noun with probability 1 to itself (i.e., $P(c|n) = \begin{cases} 1, & \text{if } \; c = n \\ 0, & \text{otherwise} \end{cases}$)).

Therefore, as few distributional assumptions as possible should be made about cluster membership. This is achieved by adding the constraint that the average entropy H of the cluster membership distributions is maximal (that is cluster membership is maximally random):

$$H = -\sum_n \sum_c P(c|n) \log(P(c|n)) \longrightarrow \max \tag{9.10}$$

Constraints 9.9 and 9.10 can be combined into one single constraint, namely

$$F = D - H/\beta \longrightarrow \min \tag{9.11}$$

F is called free energy and β is a free parameter. Minimization of the free energy F corresponds to finding a trade-off between minimizing distortion and maximizing entropy. The distortion D and the entropy H depend on each other (the cluster distributions $P(v|c)$ obviously affect the cluster membership probabilities $P(c|n)$ and vice versa). It is therefore difficult to minimize distortion and maximize entropy at the same time. Pereira, Tishby and Lee follow a two-stage iterative approach to minimize the free energy F.

In the first step of the iterative process, the distortion D is minimized under fixed class membership probabilities $P(c|n)$ and thus fixed entropy H yielding estimates $P(v|c)$ for the cluster distributions. In the second iteration step, the entropy H of the membership distributions is maximized under the constraint that the distortion D and the cluster distributions $P(v|c)$ be fix at their current estimates. The cluster membership probabilities $P(c|n)$ maximizing H are found using Lagrange multipliers and are a function of the current cluster distributions $P(v|c)$.

This procedure leads to the following re-estimation formulas for the cluster membership distributions $P(c|n)$ and the cluster centroid distributions $P(v|c)$:

$$P(v|c) = \sum_n P(n|c) P_{\mathrm{MLE}}(v|n)$$

and

$$P(c|n) = Z_n * e^{-\beta D(P_n, P_c)}$$

Equation 9.12 tell us that each cluster is represented by a distribution which is a weighted average of the distributions of its constituents, the weights $P(n|c)$ indicating the probability of observing noun n in cluster c.

Formula 9.12 is intuitively clear as well: The probability of a noun n belonging to class c decreases as the distance between the noun's distribution and the cluster distribution increases. Z_n is a normalizing constant which ensures that $P(c|n)$ is a probability distribution, i.e., $\sum_c P(c|n) = 1$.

Clustering By Committee: Pantel and Lin (2002) developed an algorithm called *Clustering By Committee* for automatic discovery of word senses from corpus text. To begin with, the authors recursively identify a set of tight clusters with highly different features, called committees. Words are then picked by the committee with which they have the biggest overlap of features to form concept clusters. Once a word is assigned to a concept cluster, the overlapping features are removed from the word which enables the discovery of its less frequent senses.

A parser is used to extract grammatical relationships of the form (*head*, *relation*, *modifier*) from the corpus. Each word is then represented by a feature vector whose dimensions correspond to possible grammatical contexts (*relation*, *modifier*) of the word. For example, the grammatical contexts of the word "reptile" (i.e., the features corresponding to the positive entries in the feature vector) are:

$$
\begin{array}{rcl}
(\textit{subj-of} & , & \text{survive}) \\
(\textit{subj-of} & , & \text{move}) \\
 & \ldots & \\
(\textit{obj-of} & , & \text{catch}) \\
(\textit{obj-of} & , & \text{chase}) \\
 & \ldots & \\
(\textit{nmod-of} & , & \text{eggs}) \\
(\textit{nmod-of} & , & \text{skin}) \\
 & \ldots & \\
(\textit{nmod} & , & \text{lizard}) \\
(\textit{nmod} & , & \text{adult}) \\
 & \ldots & \\
(\textit{adjmod} & , & \text{slow}) \\
(\textit{adjmod} & , & \text{herbivorous}) \\
 & \ldots & \\
 & \ldots &
\end{array}
$$

Since some features are more descriptive of a word w than others, Lin defines the value of a feature f to be the pointwise mutual information $MI(w, f) = \frac{P(w,f)}{P(w)*P(f)}$ between w and f which can be approximated by

$$\widehat{MI}(w, f) = \frac{|w, f| * |*, *|}{|w, *| * |*, f|} \tag{9.12}$$

using the maximum likelihood estimates $\hat{P}(w) = \frac{|w,*|}{|*,*|}$, $\hat{P}(f) = \frac{|*,f|}{|*,*|}$ and $\hat{P}(w, f) = \frac{|w,f|}{|*,*|}$ to estimate the probabilities $P(w)$, $P(f)$ and $P(w, f)$. Since mutual information is known to favor low frequency words/features, the MI scores are adjusted via multiplication with an empirical discounting factor

$$a = \frac{\hat{P}(w, f)}{\hat{P}(w, f) + 1} \times \frac{min(\hat{P}(w), \hat{P}(f))}{min(\hat{P}(w), \hat{P}(f)) + 1} \tag{9.13}$$

The similarity between two words w_1 and w_2 is then defined to be the cosine of the angle between the corresponding MI vectors v_{w_1} and v_{w_2}:

$$sim(w_1, w_2) = \frac{\langle v_{w_1}, v_{w_2} \rangle}{\| v_{w_1} \| * \| v_{w_2} \|} \tag{9.14}$$

To cluster a set of words W, the algorithm performs the following three steps:

1. For each $w \in W$ compute its top n neighbors $N(w)$, where the neighbors are ranked according to the cosine similarity measure defined in Equation 9.14 earlier.

2. Recursively identify a set of tight and mutually far-off clusters (the so-called committees).

3. Assign each w to the cluster to whose committee it is most similar.

To determine committees, candidate committees are extracted from the set of local neighborhoods $N(w)$ in the following way. Each of the local neighborhoods $N(w)$ is divided into clusters (by means of average-link clustering). The tightest of the resulting clusters (the cluster whose average similarity between cluster members is biggest) is added to a list of candidate committees. The list of candidate committees is sorted in descending order of their tightness. Starting with the tightest cluster, the candidates are iteratively added to the final list of committees if their similarity to each of the previously extracted committees falls below a certain threshold.

Finally, in Step 3, concept clusters are computed each of which initially corresponds to a committee. This is done by assigning each w to the concept cluster to the committee (more precisely the centroid of the committee) to which it is most similar. To prevent ambiguous words from affecting cluster quality, words are only assigned to a *cluster* and not to the *committee* of the cluster.

To allow a word w to belong to several concept clusters, after addition of w to a cluster c, the features which w shares with c are removed from w and re-comparison of w's reduced feature vector with the committee centroids will reveal the less prevalent senses of w.

Automatic Word Class and Word Sense Identification: Gauch and Futrelle (1994) describe a method for automatic classification of words into classes of related words using a context-based similarity measure. In contrast to the bag-of-words approach commonly used in IR, Gauch and Futrelle also take into account the position of the context words relative to the target word to be classified.

Initially a set of content words is determined which are the 150 most frequent words in the corpus. Given a target word t, Gauch and Futrelle then record the words appearing within a context window of ± 2 words around t together with their position within the window. These co-occurrence counts are summarized in a 600-dimensional context vector, the first 150 indices of which correspond to the instantiations of the first context position with each of the 150 content words, the elements 151-300 of which correspond to the instantiations of the second context position with each of the content words, and so on.

Each entry of the context vector is filled with the mutual information between the context word c and the target word t given position j in the context window (the pair (c, j) coincides with the entry's index):

$$\mathrm{Mi}_j(c,t) \quad = \quad log\left(\frac{f_j(c,t)f_j(*,*)}{f_j(c,*)f_j(*,t)}+1\right)$$

where $f_j(c,t)$ denotes the corpus frequency of c occuring at position j in a context window around w and

$$\begin{aligned} f_j(c,*) &= \sum_{\tilde{t}} f_j(c,\tilde{t}) \\ f_j(*,t) &= \sum_{\tilde{c}} f_j(\tilde{c},t) \\ f_j(*,*) &= \sum_{\tilde{c},\tilde{t}} f_j(\tilde{c},\tilde{t}) \end{aligned}$$

That is the context vector of the target word t is given by

$$\vec{v}_t = (\ \mathrm{Mi}_{-2}(c_1,t),\ldots,\mathrm{Mi}_{-2}(c_{150},t),\ldots\ldots,\mathrm{Mi}_2(c_1,t),\ldots,\mathrm{Mi}_2(c_{150},t)\ )$$

Using the inner product of the mutual information vectors, Gauch and Fautrelle compute the similarity for all pairs of words. Words are then grouped into classes by application of hierarchical agglomerative clustering in the similarity space. Initially, each word constitutes a cluster. The two most similar clusters are successively joined, producing a binary word tree. Once a new cluster is formed it is represented by an updated MI-vector whose elements are the MI-scores resulting from the sum of the context frequencies of its constituent members.

Each node in the binary tree represents a word class which consists of all its descendant nodes (words) in the tree. Gauch and Fautrelle found out that nodes close to the root node consisted of syntactically close words, i.e., words with the same part-of-speech, but with little semantic commonness. Further down the hierarchy, however, syntactically *and* semantically close words were grouped together.

Just as the clustering method groups contextually similar words together, it can be used to detect the different senses of an ambiguous word by classifying all instances of the ambiguous word according to their context. The context vector of a word instance is built from only one realization of the target and its surrounding context. Therefore, in each of the 4 150-dimensional segments of the context vector (corresponding to the 4 positions in the context window) at most one entry is non-zero. This makes classification almost impossible, since there will hardly be any instances of the ambiguous word with overlapping context vectors. Based on the earlier computed word similarities, Gauch and Fautrelle expand the context vectors by adding non-zero entries for all those content words whose similarity with the actually observed context word exceeds a fixed threshold. Similarity of word instances then corresponds to similarity of the expanded context vectors, and classes of similar instances (i.e., sense clusters) are found using hierarchical agglomerative clustering.

Gauch and Fautrelle mention that this methodology can also be applied to words with syntactic (that is POS-) rather than semantic ambiguity, namely passive participles such as *deduced* and *cloned* which can also function as adjectives or past participles.

9.2.4 Word sense induction by clustering word contexts

Based on the idea that word contexts represent smaller (more specific) meaning units than words, several methods have been developed to cluster a target word's contexts into groups of similar contexts which can then be interpreted as senses. In this section, we review some of this work.

Automatic word sense discrimination: Upon examining the words surrounding a word in focus, we often know which sense of the word is being referred to in a particular instance. The following are occurrences of the word *rock* in the BNC together with a context of ten words to either side of *rock*.

> ... mile or so along the sand, she sat on a ***rock***, looking at the grey/indigo streak of Italian coastline. The ...
> ... the river valley. Along the crest of this ridge the ***rock*** had been exposed in several places, and eroded into a ...
> ... sell 15,000 jazz LPs throughout the UK. For pop and ***rock*** artists , success has to be measured by records selling in ...
> ... British musicians and bands currently working outside the mainstream of ***rock*** and pop. The bands which appeared were selected by the...

Each of these contexts addresses a particular sense of the word *rock*: The first two contexts are about *rock* as "stone", the other two about *rock* "music". The idea of Schütze's word sense discrimination algorithm is to use unsupervised clustering to automatically divide the set of all occurrences of a word into clusters of contextually similar occurrences which can then be interpreted as word senses.

Words and their occurrences are described as points in a highdimensional vector space, the so-called Word Space in which spatial proximity reflects semantic similarity.

The Word Space is built by recording cooccurrences of each word in the text with a set of content-bearing words in a cooccurrence matrix M. Each entry $M(i, j)$ of the matrix holds the number of times words w_i and w_j occur within a distance of less than k words in the text. Figure 9.3 shows a sketch of the cooccurrence matrix.

The columns of the matrix form the dimensions of the Word Space, and each row constitutes a coordinate vector defining the position of the corresponding word in Word Space. Since words with related meanings are surrounded by similar words, they receive similar coordinates and are therefore mapped to points near one another in Word Space. This is visualized in Figure 9.4 which shows a projection of the Word Space onto two dimensions by considering only the *music* and *ice* coordinates of each word.

Very frequent words have big coordinates and are mapped to points far away from the origin. Rare words, on the other hand, have small coordinates and are close to the origin and to each other, even if they share only few positive coordinates. This frequency effect can be overcome by normalizing the coordinate vectors (which are called word vectors from now on). All words then have the same distance to the origin. They are mapped to points on the unit circle, as is illustrated in Figure 9.5.

$$\begin{array}{ll}
 & \begin{array}{cccccccccc} \dots & \dots & \dots & \text{music} & \dots & \dots & \text{ice} & \dots & \dots & \dots \end{array} \\
\begin{array}{l} \dots \\ \dots \\ \text{sand} \\ \dots \\ \text{guitar} \\ \dots \\ \text{concert} \\ \dots \\ \dots \\ \text{lava} \\ \dots \\ \text{song} \\ \dots \\ \dots \end{array} &
\left(\begin{array}{cccccccccc}
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & 7 & \dots & \dots & 32 & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & 32 & \dots & \dots & 6 & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & 25 & \dots & \dots & 8 & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & 21 & \dots & \dots & 2 & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & 7 & \dots & \dots & 32 & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots
\end{array}\right)
\end{array}$$

Figure 9.3: A sketch of the word cooccurrence matrix M.

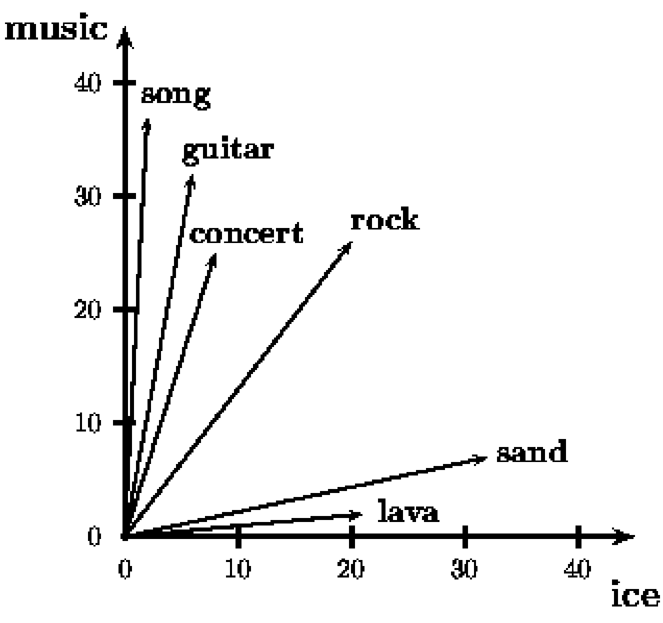


Figure 9.4: A projection of the Word Space onto two dimensions.

Similarity between words in the Word Space is measured as the cosine of the angle enclosed by the corresponding word vectors, which is equal to their scalar product and takes values between 0 and 1. The coordinate vectors of nearby points on the unit circle enclose a small angle and give rise to a high similarity score (close to 1).

Sets of words can now be represented in this same Word Space by their centroid which is obtained by summing their coordinate vectors and normalizing the result. In particular, Schütze takes an occurrence of a word to be the set of its surrounding words. A word occurrence is therefore mapped to the point in Word Space whose coordinate vector is the centroid of the coordinate vectors of its neighboring words. Occurrence vectors can be thought of as the average direction of their contributing word vectors. Figure 9.5 shows the vector representations of two occurrences of *rock*, one surrounded by the words *sand* and *lava*, and another one with neighboring words *guitar* and *concert*.

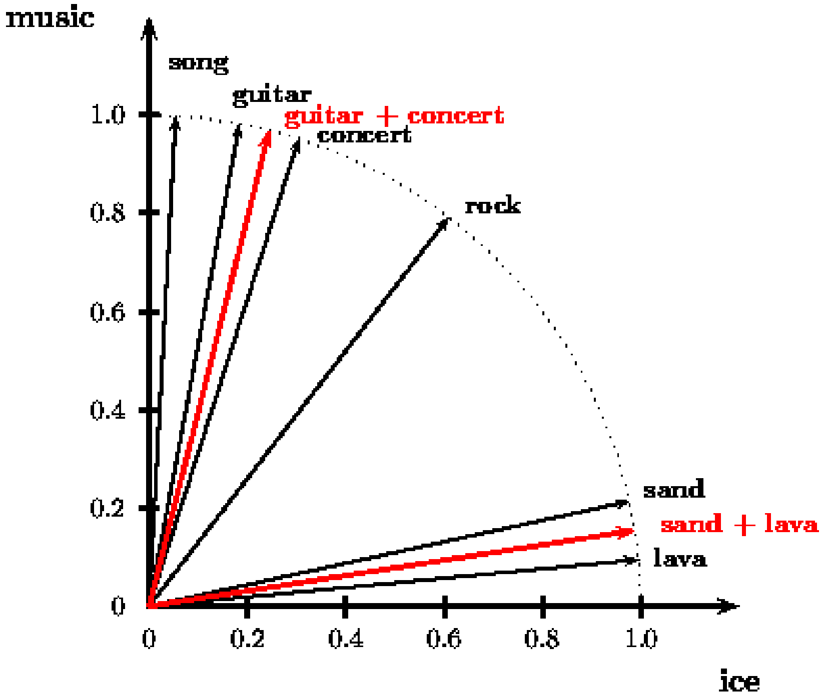


Figure 9.5: Normalized word vectors and context vectors.

The cooccurrence matrix tends to be very sparse, which is a problem when computing similarities between the row vectors. The scalar product will be close to 0 even for semantically similar words because of the many, not always matching, zero entries. This problem is circumvented by applying Singular Value Decomposition (SVD) (Deerwester et al., 1990) to the original cooccurrence matrix. SVD identifies the main directions in Word Space which are then used as new dimensions of a lower dimensional feature space. Related content-bearing words which constituted different dimensions in the original Word Space are collapsed to form a single dimension in the reduced Word Space.

For ease of understanding, however, Figures 9.4 and 9.5 show vectors in the original Word Space, but to picture their representations in the reduced Word Space, simply replace *music* and *ice* with two of the new dimensions.

To learn the senses of a target word, Schütze collects all instances of the word in the text and computes their representations in Word Space. Instances of the same sense of the word will have similar surroundings and will therefore be much closer to each other in Word Space than instances of different senses of the target word.

An EM based clustering algorithm is applied to the set of all occurrence vectors to identify groups of similar occurrences. The desired number of clusters needs to be specified in advance and allows the user to decide how fine the distinctions between different senses should be. The centroids of the resulting occurrence clusters (the "sense vectors") serve as Word Space representations of the different senses of the word.

To disambiguate an unseen occurrence of an ambiguous word, its vector is compared to all of the sense vectors, and the occurrence is categorized as an instance of the sense to whose representation it is most similar.

Discovering the Meanings of an Ambiguous Word by Searching for Sense Descriptors with Complementary Context Patterns: Rapp (2003) pursues two different approaches to automatic induction of word senses. The intuitions underlying his approaches are:

- *Complementarity of features*: The lexical environment of an ambiguous word is the union of the lexical environments of each of its meanings.
- *Variance of co-occurrence frequencies*: The words the ambiguous word w co-occurs with and the frequencies with which they co-occur with w, vary widely across different text types and corpora. For example, while, in a text on nutrition, "apple" will co-occur often with "eat", "vitamins", "food" etc., it will rarely co-oocur with words such as "computer", "toshiba", "monitor", etc. In a technical journal, on the other hand, we expect much more occurrences of "apple" with computer-related terms, but hardly any occurrences with food-related terms.

The basic assumption underlying this approach is that the different meanings of an ambiguous word are described by complementary features. Therefore, the feature vectors of a set of words, each representative of one of the meanings, should add up to the feature vector of the ambiguous word.

A feature vector is computed for each word by recording its co-occurrences with all other words in the corpus within a window of ± 1 words (after elimination of function words). Semantic similarity of words is then measured as the similarity of the corresponding co-occurrence vectors, using a standard vector similarity measure [4].

More formally, the assumption is that if the k senses of an ambiguous word w are well-described by the words $s_1 \dots s_k$, the co-occurrence vector v_w of w should be approximately

[4] Rapp uses the city-block metric. The city-block distance between a vector x and a vector y is defined as $d(x, y) = \sum_{i=1}^{N} |x_i - y_i|$ where $x = (x1, \dots, x_N)$ and $y = (y_1, \dots, y_N)$.

equal to the average $\frac{1}{k} * \sum_{i=1}^{k} v_{s_i}$ of the co-occurrence vectors v_{s_i} of the sense descriptors s_i. In his experiments, Rapp only considers the case of $k = 2$, but his procedure can be easily translated to $k \geq 2$.

To induce the senses of a given ambiguous word w, the author first identifies a set of possible sense descriptors consisting of the top n words most closely related to w. Relatedness to w is measured using the log-likelihood association score with a window of two words. This method allows first order (a word which occurs close to the ambiguous word in the corpus, e.g., "eat" as sense descriptor for apple-"fruit") as well as second order associates (a word with a similar lexical neighborhood, e.g., "strawberry" as sense descriptor for apple-"fruit") to function as sense descriptors.

For each possible combination of candidate sense descriptors, he then computes the average of the corresponding co-occurrence vectors and compares the resulting vector with the co-occurrence vector v_w of w using the city-block vector similarity. The pair of possible descriptors for which the average co-occurrence vector is most similar to v_w is considered to be the most characteristic out of the two main senses of w.

Rapp states that a pair of words which perfectly distinguish the meanings of an ambiguous word usually does not exist. Instead of using single words as sense descriptors, Rapp suggests using linear combinations of words which are better suited to capture word meaning and can be computed using Independent Component Analysis (ICA). ICA is a formalism usually applied in signal processing to recover k independent signals from at least k different linear mixtures of these signals. For a correct application of ICA, independence of the original signals is crucial.

In the case of sense induction, ICA can be applied to co-occurrence vectors of w derived from corpora of different domains. The co-occurrence vectors constitute different mixtures of w's senses and ICA can be used to identify the underlying independent components which correspond to w's senses.

Rapp takes Yarowsky's list of ambiguous words and computes co-occurrence vectors for each of the words in each of three corpora of different domains. He then applies ICA to each vector-triple to identify two independent components, i.e., the descriptors of the two main senses. To arrive at an interpretation of these independent vectors, that is to label the senses, Rapp compares the independent components with the co-occurrence vector of each word in the corpus which consists of the union of the three different corpora. The word whose co-occurrence vector in the joint corpus is closest (in terms of vector similarity) to the independent vector is chosen as sense label.

Cartographie lexicale pour la recherche d'information: *HyperLex* (Véronis, 2003) is an algorithm for automatic discrimination of word-uses in a given text. The approach is based on the detection of high density components in the graph consisting of a target word w and its cooccurrences which are collected using a bag-of words approach. The basic hypothesis underlying HyperLex is that the different uses of a word form groups of highly interlinked nodes in the graph. Detecting the different uses of a word therefore amounts to isolating regions of high density in the cooccurrence graph.

The algorithm iteratively determines the most characteristic (*root*) cooccurrent of each of the uses of the target word w. It starts from a list of w's neighbors in the graph, which is sorted by decreasing frequency of occurrence. For each node in the list, the algorithm successively checks whether it qualifies as root of a (previously unseen) usage of w. A word v constitutes the root of a usage

1. If at least k of its neighbors are not neighbors of a previously discovered sense
2. The average link strength between v and these k neighbors is higher than a predefined threshold

The strength of the links in the graph is determined according to

$$w(v_1, v_2) \;=\; \max(\, P(v_1|v_2),\; P(v_2|v_1)) \tag{9.15}$$

where

$$P(v_1|v_2) = \frac{\mathrm{freq}(v_1, v_2)}{\mathrm{freq}(v_2)} \quad \text{and} \quad P(v_2|v_1) = \frac{\mathrm{freq}(v_1, v_2)}{\mathrm{freq}(v_1)} \tag{9.16}$$

For words v_1 and v_2 which are representative of the *same* usage of w, the probability of observing one word given the other is significantly higher than the marginal probabilities. For words v_1 and v_2 which are representative of *different* senses, the conditional probabilities are expected to be lower than the the marginal probabilities. In this case, the presence of one of the words predicts the absence of the other. The algorithm underlying HyperLex works as follows:

$L = \mathrm{Neighbors}(w)$ (Sorted by decreasing frequency)
$\bar{L} = \{\}$ (List of nodes already considered)
$R = \{\}$ (List of roots)

NEIGHBOR: For each $n \in L$ {
  If $n \in \bar{L}$ { next NEIGHBOR }
  If n satisfies Conditions 1 and 2 {
    $R = R \cup \{n\}$
    $\bar{L} = \bar{L} \cup \{n\} \cup \mathrm{Neighbors}(n)$
  }
}

The set R contains the discovered root words which together with their proper neighbors (words which have a strong link to the root and which are not linked to any other root word) represent the different uses of the target word in the corpus.

Discovering word senses from a network of lexical cooccurrences: Ferret (2004) developed a method for differentiating the senses of a word from a network of lexical cooccurrences. Nodes in the network correspond to words, and links represent significant cooccurrences of words in the corpus which are collected using a bag-of-words approach (Ferret experimented with both, first-order and second-order cooccurrences).

Word senses are discovered word by word by investigating the subgraph around the target word's cooccurrents. The idea is that cooccurrents describing the same sense of the target word are much more strongly interlinked than cooccurrents describing different senses.

To identify groups of related cooccurrents in the cooccurrence graph, Shared Nearest Neighbors (SNN) Clustering is used which is illustrated in Figure 9.6 and which can be summarized as consisting of the following steps:

1. For each link in the graph, count how many neighbors the incident nodes have in common.
2. Build the SNN graph by keeping only links with a minimum of n_0 shared neighbors.
3. In the SNN graph, mark all "strong" links. These are links with a weight of at least $n_1 > n_0$ common neighbors.
4. Assign a weight to each node which is the number of strong links the node is involved in. Nodes which participate in at least m strong links give rise to core coccurrents of a sense.
5. Core nodes connected by a strong link are grouped into the same sense cluster. Other non-core nodes are then added to a core cluster if they are attached to the cluster via a strong link.

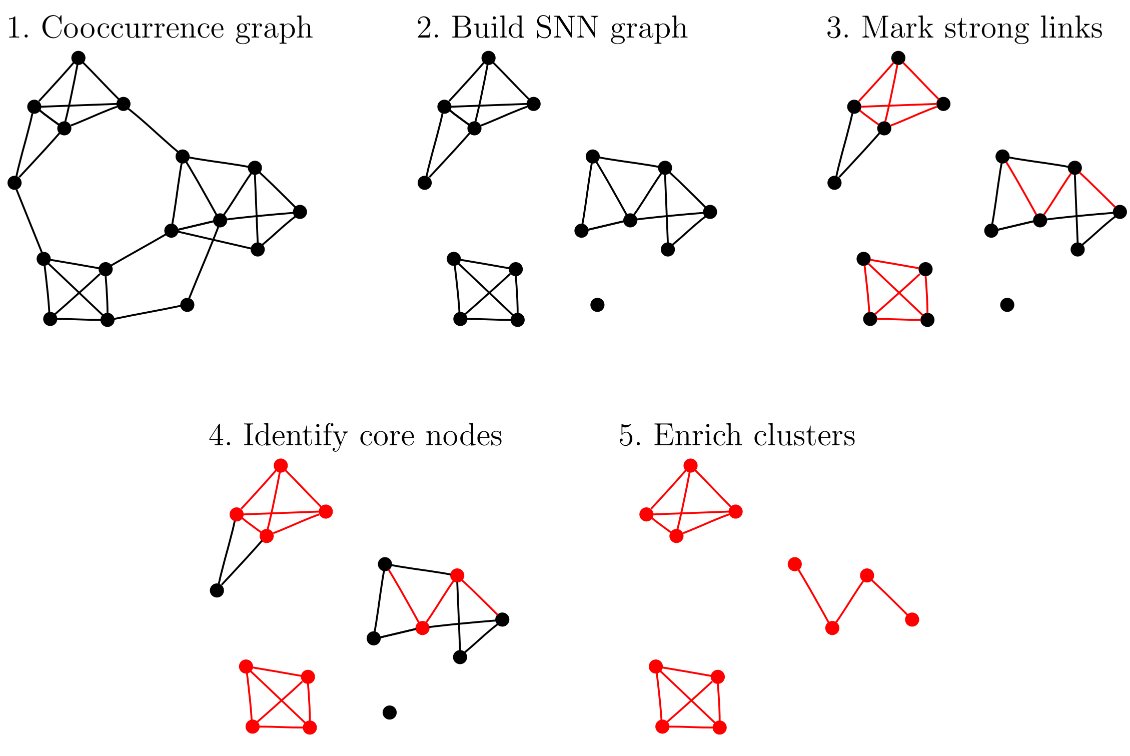


Figure 9.6: SNN Clustering.

9.2.5 Identifying and assessing ambiguity

Ambiguity is not a binary decision of *false* versus *true*, but rather varies along a continuous scale ranging from non-ambiguous to highly ambiguous.

Little work exists on the topic of automatic ambiguity assessment. In fact, to our knowledge, the only piece of work in this direction is the ambiguity detection algorithm of Sproat and van Santen (1998). The phenomenon of (non-)ambiguity is however closely related to the concept of generality (specificity), some of which we review in this section.

Automatic ambiguity detection: Sproat and van Santen (1998)'s approach to quantifying ambiguity rests on the idea that, whereas the neighborhood of an *unambiguous* word is semantically coherent, dissimilarity among an *ambiguous* word's neighbors is high.

A word's neighbors are represented as points in a two-dimensional space in such a way that Euclidean distance best preserves semantic distance between neighbors. Semantic distance between two words is measured as the relative frequency of co-occurrence of the words within a fixed-sized text window.

The authors observed that the neighbors of an ambiguous word w are distributed along rays diverging from the origin which corresponds to the target word w, and that different rays correspond to different meanings of w.

In terms of polar coordinates (r, θ), this means that the distribution of the angle θ shows a peak for each of w's meanings. Therefore, for unambiguous words, only one (broad) peak will be observed. With increasing ambiguity of w, the distribution of θ deviates more and more from a single-peak distribution.

To quantify the degree of deviation from single-peakness, the authors compute a best single-peaked fit to the distribution of θ, and use the inverse of the goodness of fit as a measure for a word's degree of ambiguity.

Measuring Semantic Entropy: Melamed (1997) developed a measure for semantic ambiguity/non-specificity which is based on the concept of *entropy*. The idea underlying Melamed's approach is that very unambiguous/specific words, which he calls semantically heavy, are consistently translated into other languages. The translational counterparts of ambiguous/general words, on the other hand, vary a lot. The more random the distribution of a word's translations, the more ambiguous/uninformative the word. The randomness of the distribution, and hence ambiguity, is measured using the information-theoretic concept of entropy.

Determining the specificity of nouns from text: Caraballo and Charniak (1999) propose and evaluate a method for determining the relative specificity of nouns. The idea underlying their method is that very general words tend to be modified frequently while more specific words are rarely modified. Caraballo and Charniak proceed by checking not only how often a noun is modified, but also how much the modifiers vary. This is captured by the distributional entropy of a noun's modifiers which, in information theoretic terms, measures the indeterminateness of the modifier distribution. Caraballo and Charniak showed that the measure works well for assessing the relative specificity of nouns. Evaluation was done by looking at three small WordNet sub-hierarchies and counting how many of the hypernym-hyponym relationships in the tree could be recovered.

9.3 Idiomaticity

The following approach by Degand and Bestgen (2003) to the acquisition of idioms is based on very similar ideas as our approach to idiom recognition described in Chapter 7. Furthermore, the results reported are similar to our findings.

Towards automatic retrieval of idioms in French newspaper corpora: Degand and Bestgen (2003) propose a procedure for automatically retrieving idiomatic expressions from large text corpora. Idioms are viewed as a kind of multi-word expressions, and the core idea is to use multiword extraction techniques to obtain a list of candidate idioms, and then apply several filters to remove all non-idiomatic terms from the list. The proposed filters are based on the following three hypotheses:

1. Idioms are lexically and syntactically rigid, i.e., they do not allow for synonym substitution and their order is fix.

2. The individual words composing the idiom show little semantic similarity.

3. Idioms show little similarity with their surrounding context.

To assess to which degree a candidate idiom fulfills these three assumptions, the authors propose the following. Property 1 is measured by counting an expressions number of "lexical neighbors". A "lexical neighbor" of an expression is a word sequence which is obtained from the expression by making only a small change, such as substitution of a single word, insertion of a word, or swapping of two words. Expressions which have many lexical neighbors are syntactically and lexically flexible. Instead, expressions for which only few neighbors are observed are good candidates for idioms.

Properties 2 and 3 are both measured using Latent Semantic Analysis. Words, expressions and contexts all have a vector representation in the same vector space and similarity between words, and similarity between an expression and its context, is measured using cosine similarity.

The authors applied their algorithm to French newspaper text and find that Filter 1 retrieves most idioms, but a lot of non-idiomatic expressions, too. Filter 2, on the other hand, turns out to miss out on a lot of idiomatic expressions, whose constituents are, contrary to the assumption, semantically similar. Most of the non-idiomatic expressions are correctly recognized as such. Similarly, Filter 3 failed to extract some of the idiomatic expressions, which did show some similarity to the surrounding text segments.

9.4 Synonymy

Many techniques have been developed for the automatic extraction of similar words. However, most of these techniques do not distinguish between contextually similar words and true synonyms. This section reviews some work on the extraction of true synonyms.

Identifying synonyms among distributionally similar words: Lin et al. (2003) developed a filter for eliminating semantically conflicting words (in particular antonyms whose distributional profiles are often very close to the one of the target word) from a list of distributional neighbors. Their method is based on a set of patterns (*from ... to ...* and *either ... or ...*) indicative of antonymy or more generally semantic incompatibility. If a distributional neighbor and the target word often occur in such a pattern (this is tested by querying a Web search engine), they are likely to be semantically conflicting.

Lin et al. propose another filter which makes use of a bilingual dictionary. By translating the French translations of an English target word back into English, the authors obtain a set of potential synonyms of the target word which they compare against the set of its distributional neighbors. True synonyms are those words which occur in both sets. One drawback of this method is its dependence on the availability and quality of bilingual dictionaries.

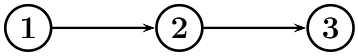
Automatic extraction of synonyms from a dictionary: Blondel and Senellart (2002), use graph analysis to extract synonyms from a monolingual dictionary. The relationship "appears in gloss" induces a directed graph where each node denotes a dictionary entry. More precisely, there is a link from word u to word v if v appears in the definition of u.

Given a target word, a neighborhood graph is constructed which consists of all the words the target word points to (the words in its gloss) or is pointed to (the words associated with the glosses the target word appears in). The nodes in the neighborhood graph are then ranked according to their *centrality score*. A node showing a high degree of centrality is one which is a good *hub* (points to many nodes) as well as a good *authority* (is pointed to by many nodes) [5].

The centrality scores are computed by comparing the neighborhood graph with the line graph. A node's centrality is its similarity to node 2 in the line graph. Nodes in the neighborhood graph which resemble node 2 have many ingoing and outgoing links. Thus, centrality is a measure of the amount of information passing through a node. The authors' hypothesis is that these busy central nodes are good candidates for synonymy.

[5] The notions of *hub* and *authority* have been coined by Kleinberg (1999) to describe the quality of web pages. A good authority is a web page which is highly relevant to a certain topic. Such a web page is characterized by many incoming hyperlinks from other people who think it is interesting. A good hub is a web page which points to many good authorities related to the same topic. As Kleinberg puts it

> Hubs and authorities exhibit what could be called a *mutually reinforcing relationship*. A good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs.



We will illustrate Blondel et al.'s technique with the help of an example. Let us consider the target word "chocolate" which may have the following entry in a dictionary:

> **Chocolate:** A preparation made from *cocoa beans* that have been roasted, husked, and *ground*. Chocolate today is usually *sweetened* with *sugar*.

This definition has some overlap with the (imaginary) dictionary entry of "cocoa":

> **Cocoa:** Refers to the fruit of the cocoa plant; or to the dry *powder* made by *grinding* the *beans* and removing the cocoa butter. Heated with milk and *sweetened* with *sugar*, cocoa makes a tasty beverage.

Besides their definitions sharing words, "chocolate" and "cocoa" appear jointly in the glosses of other words, for example *cacao* and *truffle*:

> **Cacao:** A small (4-8 m tall) evergreen tree in the family Sterculiaceae, native to tropical South America, but now cultivated throughout the tropics. Its seeds are used to make *cocoa* and *chocolate*.
>
> **Truffle:** A bite-sized petit four, made from *chocolate* and ganache to which flavourings have been added, such as liqueurs or essences. Truffle mixtures can be piped in balls or long strands, rolled in *cocoa powder*, icing sugar or dipped in couverture.

Words in italics are shared by glosses. Figure 9.7 shows the neighborhood graph of the target *chocolate*. Apart from the *chocolate* node, the node with highest centrality is *cocoa* as it has both lots of incoming as well as outgoing links. It is therefore the most appropriate synonym for *chocolate* among the nodes.

Mining the Web for Synonyms: PMI–IR versus LSA on TOEFL: Turney (2001) uses simple pointwise mutual information on web frequency data obtained by querying a search engine (AltaVista) to pick the best synonym of a target word from among a set of given alternatives. Mutual information measures the discrepancy between the observed probability of cooccurrence of a candidate synonym c and the target t with the probability expected under the assumption that c and t occur independently of each other:

$$\mathrm{MI}(c,t) = log\left(\frac{P(c,t)}{P(c)\ \ P(t)}\right) \tag{9.17}$$

Different ways of defining cooccurrence of target and candidate synonym are explored, each of which corresponds to a different web query. He finds that simple queries such as "c AND t" or "c NEAR t" score antonyms as highly as synonyms. He observes better performance

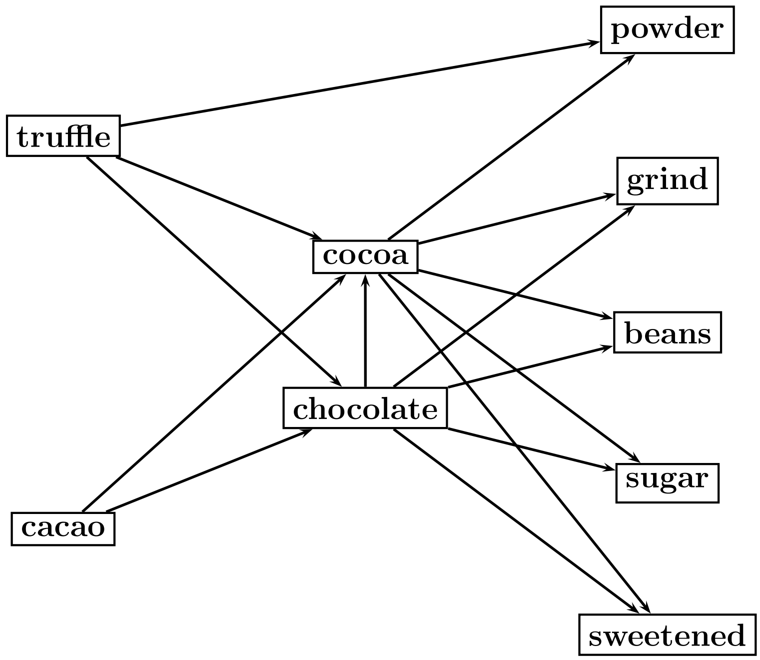


Figure 9.7: The dictionary neighborhood graph of *chocolate*. Links $u \to v$ represent the relationship "u appears in the gloss of v".

when requiring in addition that neither the candidate synonym nor the target occur near the word "not". If context is available, synonymy recognition can be further improved by requiring that c and t occur not only near each other but also together in a document with (some of) the given context words.

Chapter 10

Summary, Conclusion and Future Research

10.1 Summary

We have studied a graph model for nouns and their relationships in a text. The model has been built by collecting lists from the British National Corpus, and by connecting any two nouns which occurred in a list together. Our aim was to make explicit the semantic information hidden in the connectivity structure of the resulting word graph. We have found that a lot of interesting information about the meanings of words and their interactions can be extracted from this simple graph model.

In Chapter 4, we have proposed an algorithm which allows to iteratively extract the different senses of an ambiguous word from the subgraph induced by its neighboring words. By discriminating against previously extracted senses, the discovery of duplicate senses is avoided.

We presented results on a manually compiled test set consisting of words of different degree of ambiguity, and demonstrated that the algorithm is able to extract the different senses of a word even if they are closely related.

In Chapter 5, we have studied the graph-theoretic measure of graph curvature from a semantic point of view. Curvature is a measure of the interconnectedness of a node's neighbors. The neighbors of highly specific unambiguous words tend to be densely interlinked, which is reflected in a high curvature score. The neighborhoods of ambiguous words are instead more heterogeneous. The neighbors of an ambiguous word form different word communities which correspond to its different meanings. While neighbors pertaining to the same sense are linked to one another, neighbors related to different meanings are unconnected, which is reflected in a low curvature score.

Curvature has turned out to be not only useful for detecting ambiguous words. In addition, it provides an effective clustering tool. Ambiguous words behave like semantic "hubs", which connect otherwise unrelated word communities. By disabling these hubs, i.e., by removing nodes with low curvature scores, the word graph decomposes into separate groups of words which correspond closely to cohesive semantic categories.

We have shown that curvature clustering is particularly useful for predicting the meaning of unknown words. Whenever an unknown word is observed in a cluster together with other known words, its meaning can be inferred from the meanings of its known fellow cluster members. Our evaluation experiment was based on two different releases of the WordNet lexicon. We took the nouns covered by the earlier release to be the known words, based on which we tried to infer the meanings of the nouns which were added in the later release. Curvature succeeded in mapping many of the test words to the appropriate place in the WordNet noun hierarchy.

The nodes in the word graph represent words many of which are ambiguous between different meanings. The links between them, on the other hand, have a much more concise meaning, as their constituent words often disambiguate one another, and can be interpreted as word contexts.

Based on this observation, we investigated an alternative clustering approach in Chapter 6, which treats the links in the graph as the semantic atoms to be clustered. This approach yields a "soft" clustering of words into semantic categories, meaning that words may belong to several clusters at the same time. This is a great advantage over most conventional clustering approaches which, by forcing each word to remain in a single cluster, do not allow for words to have several senses.

The tool which allows us to shift our attention from the nodes in the word graph to the links, is the link graph transformation. The links in the original graph become the new nodes of a new transformed graph (the link graph). New nodes are linked in the transformed graph if, like links in the original graph, they occur together in a triangle. Since triangles embody a transitivity relationship, there is strong evidence that the links within a triangle all address the same meaning of the words involved.

The new nodes of the transformed graph are no longer (at least far less) ambiguous, and can thus be divided into non-overlapping clusters. We have used Markov Clustering to do this. By replacing links with their constituent words, the (mutually exclusive) clusters of links can be translated into (overlapping) clusters of words.

We have evaluated the performance of link graph clustering using WordNet as a gold standard. Link graph clustering yields considerably better results than the baseline, which we took to be Markov Clustering on the original untransformed graph.

The link graph has suffered some losses in density during the course of the graph transformation. Markov Clustering thus divided the link graph into many small clusters in such a way that several of the clusters a word appears in correspond in fact to the same sense. We have developed a measure of contrast which allows for selecting such pairs of clusters from among the many clusters a word was assigned to, which pertain to different senses of the word. Given two clusters a word appears in, contrast measures the trade-off between the two clusters being both good representatives of the word, and the clusters being dissimilar.

When building the word graph, we neglected word order by not giving the links in the graph any direction. In Chapter 7, we have extracted those word pairs (links), which have a bias for one of the two possible orderings. We found that asymmetry can be the result of semantic constraints which determine the ordering, or can be due to idiomaticity. The meaning of links whose asymmetry arises from semantic constraints is the composition of the meanings of the words it is composed of. The meaning of idiomatic links, on the other hand, cannot be inferred from the constituent words. Such word pairs pose a great challenge to computers trying to understand human language.

We have proposed different approaches for distinguishing between compositional and non-compositional links which include graph-theoretic as well as statistical techniques. Several of these methods have yet to be implemented and tested.

Many techniques have been developed for extracting semantically similar words. They do not however distinguish between semantic coordinates (such as *biker* and *climber*) and true synonyms (such as *biker* and *cyclist*).

In Chapter 8, we have attempted to extract synonyms from the word graph, our initial hypothesis being that synonyms correspond to nodes which live in the same densely connected word community, yet are not connected to one another. We have found this hypothesis to apply only to a small number of synonyms, in particular to spelling variants (*moslem* and *muslim*) or pairs of singular and plural form, which we included in our working definition of synonymy. We revised our initial hypothesis, such as to simply assume a strong bond between synonyms via shared neighbors, which caused a considerable improvement in performance. Again, we used WordNet as the gold standard to evaluate against.

10.2 Conclusion and Future Research

We have presented a collection of graph-theoretic tools for extracting semantic information from graphs built from free text. Both the model and the methods are easy to understand and implement and are easily adaptable to other kinds of linguistic (and even non-linguistic) data. Our techniques require minimal linguistic preprocessing, do not rely on any external knowledge sources, and require only a few parameters to be set. The results are comparable to those reported in the literature for previous approaches.

Our focus was on the graph-theoretic methodology rather than on sophisticated linguistic data processing. However, we expect our techniques to yield more accurate results if applied to data which is assembled in a more advanced way, such as by using a chunker. Furthermore, using more complex grammatical relationships, in particular syntagmatic relationships, will allow us to study other semantic phenomena, including regular polysemy and the disambiguation of nominalizations, for which the word graph in its current form is not applicable. The use of syntagmatic relationships will also allow us to extend the study of idiomaticity to expressions other than lists, such as adjective-noun or verb-object pairs (e.g., *French leave*; *wet blanket*; *kick the bucket*; *spill the beans*).

Another interesting direction to pursue is to investigate how enhancing the semantic information contained in (local) grammatical relationships with (global) topical information as provided by Latent Semantic Analysis, allows us to make predictions about the meanings of words even if they are not observed in the grammatical relationships used for building the graph. For example, we can associate a vector with each link in the graph, which is a representation of the contexts it occurs in. Each subgraph of the word graph is then associated with a context vector which is the average of the vector representations of the links it contains. This provides us with the ability to disambiguate an occurrence of a word by comparing its surrounding context with the context representations of its clusters which correspond to subgraphs.

Chapter 11

Zusammenfassung

Die ständig wachsende Flut von Daten und Information, insbesondere im medizinischen und technologischen Bereich, kann nur noch mit Hilfe automatischer Sprachverarbeitungssysteme bewältigt werden. Die meisten dieser Verfahren benötigen Kenntnis über Wörter und deren Bedeutungen, und greifen dazu auf existierende elektronische Lexika zurück, wie zum Beispiel das englische Online-Wörterbuch WordNet oder die deutsche Variante GermaNet.

Ein universelles Lexikon kann jedoch nie allen Bedeutungsunterscheidungen, die im Text enthalten sind, Rechnung tragen. Gerade Fachliteratur wimmelt von Fachbegriffen, die in universellen Lexika nicht aufgeführt sind. Fachlexika sind selten und können aufgrund des ständigen Auftretens neuer Wörter nicht von Hand aktualisiert werden. Es werden automatische Verfahren benötigt, die Information über Wörter und deren Bedeutungen automatisch aus einem repräsentativen Text extrahieren können.

Diese Arbeit stellt einen Wortgraphen vor, der Nomina und deren Beziehungen in einem Text modelliert. Ein Graph ist ein Modell zur Beschreibung von Objekten, die untereinander in Beziehung stehen. Die Objekte werden als Knoten (Punkte), die Beziehungen als Kanten (Linien) dargestellt.

Aufzählungen, wie zum Beispiel die folgende, enthalten oft Wörter mit ähnlichen semantischen Eigenschaften:

> *Würzen Sie Ihre Speisen also bevorzugt mit Ingwer, Meerrettich, Pfeffer, Senf, Oregano, Zimt, Nelken und Koriander.*

Wir sammeln alle Aufzählungen eines Textes[1], und bilden daraus einen Graphen, indem wir für jedes Wort einen Knoten einführen und zwei Knoten durch eine Kante verbinden, wenn die Wörter zusammen in einer Aufzählung auftreten.

Das entstehende Wortnetz enthält viel interessante und nützliche Informationen über Wörter, deren Bedeutungen und Beziehungen. Ziel dieser Arbeit ist es, diese Information, die in der Kantenstruktur versteckt ist, zu extrahieren. Wir wenden uns verschiedenen semantische Phänomenen zu, Ambiguität (Mehrdeutigkeit), Ähnlichkeit, Idiomatizität und Synonymie, denen jeweils einzelne Abschnitte gewidmet sind.

[1] Wir verwenden als Beispielfall das englischsprachige British National Corpus.

11.1 Semantische Ambiguität und Ähnlichkeit

Ein und dieselbe Wortform kann, abhängig vom Kontext, sehr unterschiedliche Bedeutungen annehmen. Mehrdeutige Wörter stellen eine besondere Schwierigkeit für viele Textverarbeitungswerkzeuge dar. Ein maschinelles Übersetzungssystem muss zum Beispiel erkennen können, auf welche Bedeutung des Wortes *Gericht* in dem folgenden Satz verwiesen wird:

> *Das Gericht vertagte das Verfahren, in dem mehr als 15.000 Aktionäre klagen, auf den 21. Juni 2005.*

Während es uns keine Schwierigkeiten bereitet, die beabsichtigte Bedeutung zu erkennen (meistens wird uns die Mehrdeutigkeit nicht einmal auffallen), stellt es für einen Computer eine schwierige Aufgabe dar, zu entscheiden ob *Gericht* als *court* oder *dish* übersetzt werden muß.

Das Forschungsgebiet der Automatischen Disambiguierung (Word Sense Disambiguation) beschäftigt sich mit der Aufgabe, die Bedeutungen von Wörtern aus einem gegebenen Kontext automatisch zu erschließen. Disambiguierungsverfahren benötigen vorab ein Lexikon, in dem alle möglichen Wortbedeutungen aufgelistet sind, die dann verwendet werden können, um die mehrdeutigen Wörter im Text mit einem Bedeutungs-"Etikett" zu versehen.

Wie wir schon erwähnt haben, können die Bedeutungen, die im Text vorkommen, und die Wortbedeutungen, die im Lexikon aufgelistet sind, nicht eins-zu-eins aufeinander abgebildet werden. Viele für den vorliegenden Text wichtige Wortbedeutungen fehlen im Lexikon. Dagegen sind im Lexikon viele Wortbedeutungen enthalten, die für den gegebenen Text oder Bereich völlig uninteressant sind. Alternativ könnte man das Repertoire der möglichen Wortbedeutungen direkt vom Text lernen.

Der Wortgraph bietet uns dafür die Möglichkeit. Schauen wir uns Abbildung 11.1 an. Die Nachbarn des *Gericht*-Knotens fallen in zwei Bedeutungsgruppen, die mit den zwei verschiedenen Bedeutungen "Speise" und "Gerichtshof" zusammenhängen. Wörter innerhalb einer Bedeutungsgruppe sind fest untereinander verbunden. Wörter, die unterschiedlichen Bedeutungsgruppen angehören, sind dagegen voneinander getrennt. Die zwei Wortgruppen werden ausschließlich durch den *Gericht*-Knoten zusammengehalten. Wird dieser vom Graphen entfernt, so zerfällt der Graph in zwei Teilgraphen, die genau der Bedeutungsunterscheidung "Speise" und "Gerichtshof" entsprechen.

Basierend auf dieser Idee haben wir einen Algorithmus entwickelt, der nach und nach aus dem Nachbarschaftsgraphen eines mehrdeutigen Wortes Bedeutungscluster extrahiert. Wir verlangen, dass jedes neue Cluster keine Ähnlichkeit mit den bereits gefundenen Clustern zeigt. Dadurch wird verhindert, dass dieselbe Bedeutung eines Wortes mehrfach extrahiert wird. Außerdem werden so die selteneren Bedeutungen eines Wortes deutlicher sichtbar und können ebenfalls gefunden werden.

Wir haben diesen Ansatz mit verschiedenen Wörtern unterschiedlichen Grades von Mehrdeutigkeit getestet, und festgestellt, dass, sogar wenn die verschiedenen Bedeutungen eines Wortes miteinander verwandt sind, diese oft gefunden werden können.

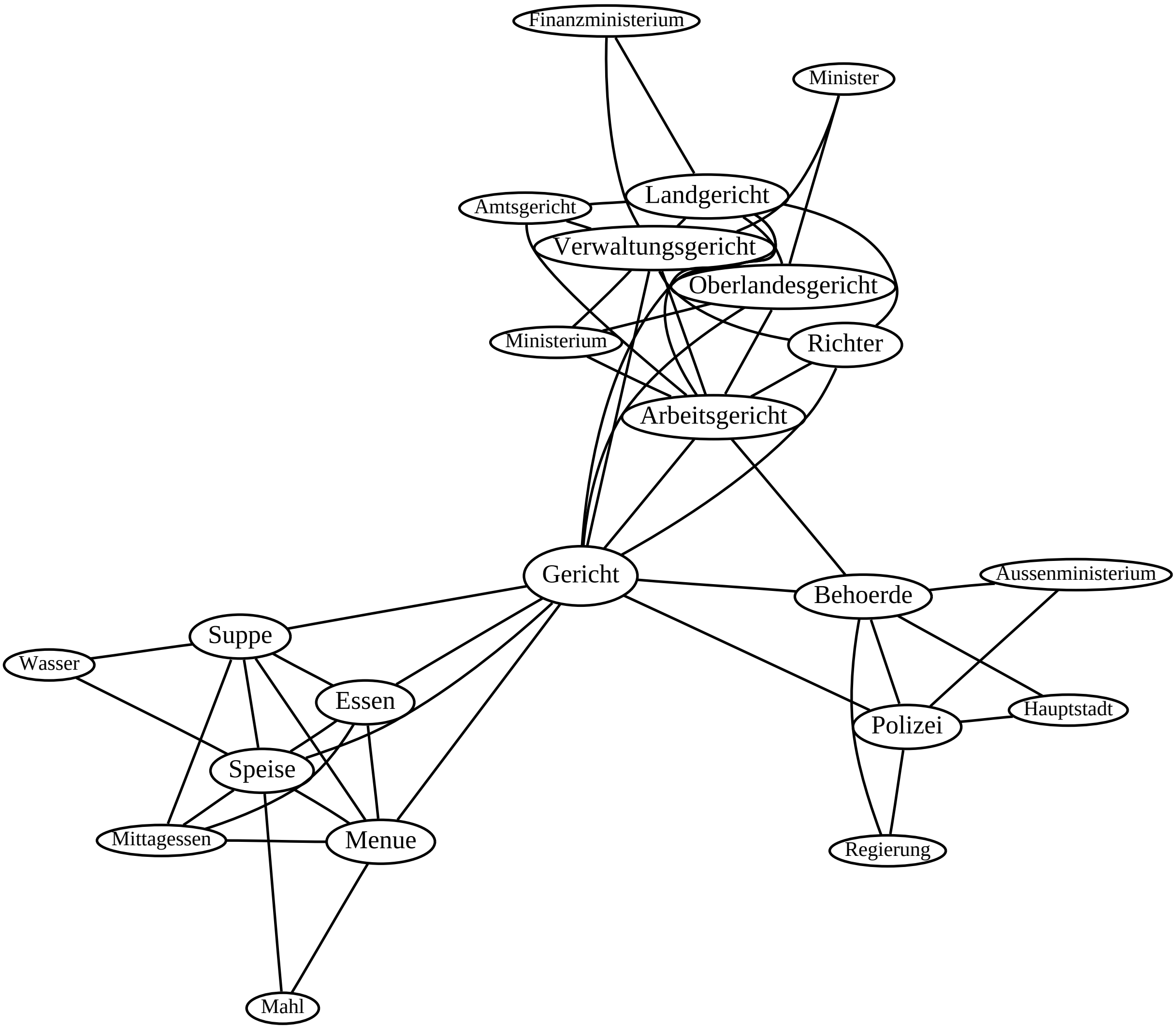
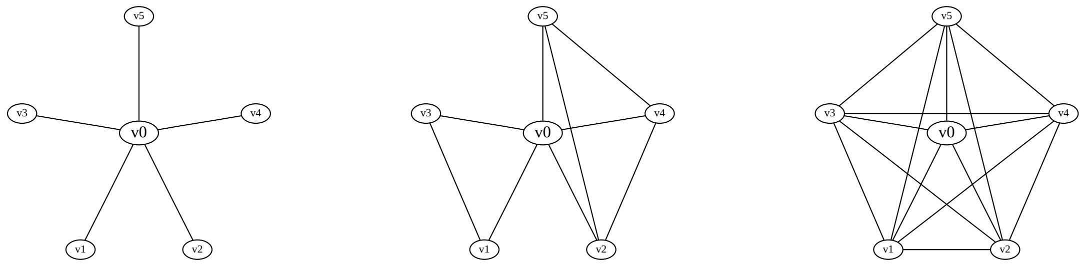


Figure 11.1: Die Nachbarschaft des Wortes *Gericht* im Wortgraphen teilt sich in zwei Gruppen auf, die den unterschiedlichen Bedeutungen "Speise" und "Gerichtshof" entsprechen.

Zunächst müssen jedoch die mehrdeutigen Wörter aufgespürt werden können. Ein graph-theoretisches Maß, das dies ermöglicht, ist die sogenannte *Krümmung* (curvature) eines Knotens im Graphen. Die Krümmung misst den Zusammenhalt innerhalb der Nachbarschaft eines Knotens. Die Nachbarn von sehr speziellen, eindeutigen Wörtern sind oft auch gegenseitige Nachbarn. Im Gegensatz dazu, ist, wie wir am Beispielwort *Gericht* gesehen haben, der Zusammenhalt zwischen den Nachbarn eines allgemeinen und/oder mehrdeutigen Wortes sehr viel geringer. Die Krümmung eines Knotens wird berechnet als der Anteil aller existierenden Verbindungen zwischen den Nachbarn eines Knotens an der Anzahl aller möglichen Verbindungen zwischen Nachbarn. Abbildung 11.2 zeigt drei Knoten unterschiedlicher Krümmung.



(a) $\mathrm{Kr}(v_0) = 0$.   (b) $\mathrm{Kr}(v_0) = 0.4$.   (c) $\mathrm{Kr}(v_0) = 1$.

Figure 11.2: Knoten geringer, mittlerer und starker Krümmung.

Wir konnten zeigen, dass die Krümmung eines Wortes im Graphen stark korreliert mit der Anzahl der Bedeutungen des Wortes im WordNet Lexikon, was bedeutet, dass die Krümmung tatsächlich ein gutes Maß darstellt, um den Grad der Mehrdeutigkeit eines Wortes vorherzusagen. Knoten hoher Krümmung entsprechen eindeutigen, Knoten geringer Krümmung hingegen mehrdeutigen Wörtern. Eine Gruppe von Wörtern, die sich durch sehr starke Krümmung im Wortgraphen auszeichnen, sind Ländernamen, die tatsächlich, mit wenigen Ausnahmen, eindeutig sind. Familiennamen hingegen stellen die Gruppe der Wörter mit niedrigster Krümmung dar. Familiennamen sind nicht nur oft "geliehen" von, zum Beispiel, Berufsbezeichnungen (*Schmied*, *Schneider*), Farben (*braun*, *schwarz*) oder aus der Natur (*Stein*, *Wald*), sondern auch mehrdeutig zwischen den vielen verschiedenen Personen, die den gleichen Namen tragen.

Die Krümmung ist nicht nur nützlich, um Ambiguität zu erkennen. Sie eignet sich auch hervorragend als Clustering-Werkzeug. Die mehrdeutigen Wörter, die durch geringe Krümmung charakterisiert sind, sind die Dreh- und Angelpunkte des Graphen. Wenn diese aus dem Graph herausgenommen werden, so geht der einzige Zusammenhalt zwischen den verschiedenen Wortgruppen verloren, und der Graph zerfällt auf natürliche Weise in kohärente semantische Klassen. Die erhaltenen Klassen sind besonders dafür geeignet, die Bedeutungen bisher unbekannter Wörter anhand der anderen bekannten Wörter in der Klasse vorherzusagen. Wir haben dies in einem Experiment folgendermaßen getestet. Wir haben diejenigen Nomina in unserem Graphen als bekannt angenommen, die in einer frühen Version von WordNet bereits enthalten sind. Alle Wörter, die erst in einer späteren Version ihren Eingang in WordNet gefunden haben, wurden stattdessen als neu betrachtet. Unsere Aufgabe war es, die Bedeutung der neuen Wörter anhand der bekannten zu identifizieren und an die richtige Stelle in WordNet zu setzen. Für viele der neuen Wörter ist uns das ziemlich präzise gelungen.

Die Knoten des Wortgraphen stellen Wörter dar. Viele Wörter haben mehrere Bedeutungen. Die Kanten des Graphen, die Wortpaare darstellen, haben hingegen beinahe immer nur eine Bedeutung. Zum Beispiel die Kante (*Gericht*, *Essen*) verweist eindeutig auf eine Speise, und die Kante (*Gericht, Arbeitsgericht*) auf den Gerichtshof. Es scheint daher sinnvoller, die Kanten anstatt der Knoten zu clustern. Die Kanten eines mehrdeutigen Wortes können dann auf verschiedene Cluster verteilt werden, die den verschiedenen Bedeutungen des Wortes entsprechen.

Um dies zu ermöglichen, führen wir eine Graphtransformation durch. Wir konstruieren aus dem bisherigen Wortgraphen einen neuen konjugierten Graphen, indem wir aus jeder Kante einen neuen Knoten machen. Innerhalb eines Dreiecks nehmen Wörter beinahe immer nur eine Bedeutung an. Wir verbinden daher zwei neue Knoten in dem neuen konjugierten Graphen miteinander, wenn sie, als Kanten im ursprünglichen Graphen, gemeinsam in einem Dreieck auftreten. Die Graphtransformation ist in Abbildung 11.3 illustriert.

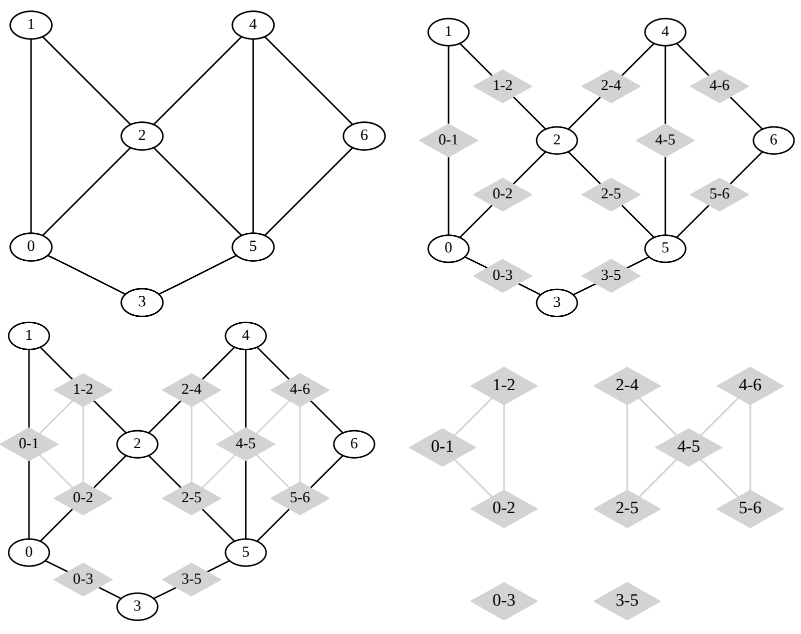


Figure 11.3: Vom Wortgraphen zum konjugierten Graphen. Der ursprüngliche Graph, neue Knoten, neue Kanten, der konjugierte Graph.

Der neu entstandene konjugierte Graph kann nun geclustert werden. Dafür verwenden wir das frei verfügbare Markov Clustering Tool. Die entstehenden Cluster bestehen aus

Knoten des konjugierten, das heißt aus Kanten des ursprünglichen Graphen. Diese (nicht-überlappenden) Cluster von Kanten, können in (überlappende) Cluster bestehend aus den einzelnen Wörtern verwandelt werden. Mehrdeutige Wörter können auf natürliche Weise über ihre verschiedenen Kanten mehreren Clustern angehören. Dies ist ein großer Vorteil gegenüber den meisten anderen Clustering-Verfahren, die es Wörtern , auch mehrdeutigen, erlauben, nur in einem einzigen Cluster zu erscheinen.

Wir haben festgestellt, dass das resultierende Clustering zwar sehr gut die Bedeutungen mehrdeutiger Wörter erkennt. Das Clustering ist aber etwas zu feinkörnig, so dass oft mehrere der Cluster, in denen ein Wort vorkommt, auf die gleiche Bedeutung des Wortes verweisen. Um dieses Problem zu umgehen, haben wir ein Verfahren entwickelt, das uns erlaubt aus den Clustern, in denen ein Wort vorkommt, solche Paare zu extrahieren, die *unterschiedlichen* Bedeutungen entsprechen. Wir nennen ein solches Clusterpaar ein kontrastierendes Paar.

Wir haben unser Clustering evaluiert, indem wir für jedes Wort und jede seiner Bedeutungen überprüft haben, wie nahe sie den Bedeutungen der anderen Clustermitgliedern in WordNet sind. Die Ergebnisse der Evaluierung haben wir mit einem naiven Ansatz verglichen, der die Wörter in disjunkte Cluster aufteilt. Das Clustering des konjugierten Graphen liefert erheblich bessere Ergebnisse als der naive Ansatz. Unsere Ergebnisse sind vergleichbar mit den Ergebnissen, die für andere Verfahren veröffentlicht wurden.

11.2 Idiome

Als wir den Wortgraphen konstruiert haben, haben wir die Reihenfolge der Wörter in einer Aufzählung einfach ignoriert. Oft spielt die Reihenfolge der Wörter aber eine Rolle. Wir haben die Kanten im Wortgraphen extrahiert, die Wortpaaren entsprechen, welche signifikant öfter in einer als in der anderen Reihenfolge auftreten. Diese Kanten konnten wir in zwei Klassen einteilen: Kanten, bei denen die feste Reihenfolge das Ergebnis semantischer Beschränkungen ist. Beispiele solcher Beschränkungen sind Starkes vor Schwachem (*Lehrer und Schüler*), Positives vor Negativem (*Leben und Tod*) oder Früheres vor Späterem (*Frühling und Sommer*). Die andere Klasse umfasst idiomatische Ausdrücke, wie zum Beispiel *Kraut und Rüben* oder *(an allen) Ecken und Enden*. Während die Bedeutung der Wortpaare der ersten Klasse von den Einzelwörtern abgeleitet werden kann, ist das für idiomatische Wortpaare nicht möglich. Diese stellen für Sprachverarbeitungssysteme daher ein Problem dar. Mit Hilfe verschiedener graph-theoretischer und statistischer Verfahren, versuchen wir automatisch die idiomatischen von den nicht-idiomatischen Wortpaaren zu trennen.

11.3 Synonyme

Während viele Verfahren für die Extraktion ählicher Wörter existieren, gibt es nur wenige Ansätze zur Extraktion echter Synonyme. Wir haben zunächst angenommen, dass Synonyme zwar mit den gleichen Wörtern in Aufzählungen vorkommen, aber nie gemeinsam auftreten (*Badeanzug und Bademütze* und nicht *Badeanzug* und *Schwimmanzug*). Im Wortgraphen bedeutet das, dass Synonyme Knoten entsprechen, die zwar nicht untereinander verbunden sind, aber viele Nachbarn teilen. Diese Annahme hat sich aber als zu restriktiv erwiesen: Entgegen unserer Annahme, treten Synonyme erstaunlich oft zusammen in Aufzählungen auf. Ein häufiger Grund dafür ist Ambiguität. Es kann sehr wohl passieren, dass ein Wort nur Synonym einer von mehreren Bedeutungen eines anderen Wortes ist. Dies trifft zum Beispiel für das Wort *Kakao* zu. *Kakao* und *(Trink)Schokolade* sind Synonyme, *Kakao* und *Schokolade(ntafel* hingegen nicht. Eine Aufzählung, die sowohl *Kakao* als auch *Schokolade(ntafel)* enthält, ist also sehr wohl vorstellbar.

Aber auch sonst treten Synonyme, insbesondere in *oder*-verknüpften Auzählungen, häufig gemeinsam auf (z.B., *Schwierigkeiten oder Probleme*). Eine abgeschwächte Version unserer ursprünglichen Hypothese hat sich als effektiver für die Extraktion von Synonymen erwiesen: Synonyme kommen möglicherweise zusammen, aber auf jeden Fall mit den glei-chen Wörtern in Aufzählungen vor, oder graph-theoretisch ausgedrückt, Synonyme entsprechen Knoten, die über extrem viele 2-Kanten-Wege miteinander verbunden sind. Mit dieser Methode konnten wir viele der Wortpaare (Kanten) als Synonyme identifizieren, die in WordNet als solche aufgelistet sind [2].

11.4 Einordnung und zukünftige Forschungsarbeit

Wir haben in dieser Arbeit verschiedene graph-theoretische Werkzeuge entwickelt und verwendet, um wertvolle semantische Information aus Texten zu extrahieren. Unsere Methoden benötigen minimale Vorverarbeitung, sind einfach zu verstehen und zu implementieren. Aufgrund der Allgemeingültigkeit sowohl des Modells als auch der vorgestellten Methoden, können diese direkt auf andere Arten linguistischen (und auch nicht-linguistischen) Materials angewendet werden. Gegenstand zukünftiger Arbeit wird es sein, das Graphenmodell auf andere Wortklassen (Verben, Adjektive) und komplexere grammatische, insbesondere syntagmatische Beziehungen zwischen Wörtern auszuweiten. Davon erhoffen wir uns, semantische Phänomene, wie zum Beispiel reguläre Polysemie oder die Mehrdeutigkeit von Nominalisierungen, für die der Wortgraph in seiner aktuellen Form nicht geeignet ist, untersuchen zu können.

[2] In WordNet ist die Synonym-Relation indirekt kodiert: Wörter sind in sogenannte Synsets gruppiert, die Konzepte darstellen. Wörter, die in dem selben Synset vorkommen, sind (Beinah-)Synonyme.

References

R. Albert, H. Jeong, and A.-L. Barabási. 2000. The Internet's Achilles' heel: error and attack tolerance of complex networks. *Nature*, 406(378).

J. Bagga. 2004. Old and new generalizations of line graphs. *International Journal of Mathematics and Mathematical Sciences*, 29:1509–1521.

T. Baldwin, C. Bannard, T. Tanaka, and D. Widdows. 2003. An empirical model of multiword expression decomposability. In *ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 89–96, Sapporo, Japan.

A.-L. Barabási and R. Albert. 1999. Emergence of scaling in random networks. *Science*, 286:509–512.

A.-L. Barabási and Z. Oltvai. 2004. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–111.

S. Benor and R. Levy. 2006. The Chicken or the Egg? A Probabilistic Analysis of English Binomials. *Language*, To appear.

V.D. Blondel and P.P. Senellart. 2002. Automatic extraction of synonyms in a dictionary. In *SIAM Workshop on Text Mining*, Arlington, TX.

B. Bollobás. 1998. *Modern Graph Theory.* Number 184 in Graduate Texts in Mathematics. Springer-Verlag.

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

H. Burger. 1998. *Phraseologie - Eine Einführung am Beispiel des Deutschen.* Erich Schmidt, Berlin.

S. Caraballo and E. Charniak. 1999. Determining the specificity of nouns from text. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing (EMNLP) and very large corpora (VLC).*

S. Cederberg and D. Widdows. 2003. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the Conference on Natural Language Learning (CoNNL)*, Edmonton, Canada.

G. Chartrand. 1985. *Introductory Graph Theory.* Dover.

M. Ciaramita and M. Johnson. 2003. Supersense tagging of unknown nouns in WordNet. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003).*

G. de Chalendar and B. Grau. 2000. SVETLAN' or how to classify words using their context. In *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW)*, pages 203–216.

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

L. Degand and Y. Bestgen. 2003. Towards automatic retrieval of idioms in french newspaper corpora. *Literary and Linguistic Computing*, 18(3):249–259.

S.N. Dorogovtsev and J.F.F. Mendes. 2001. Language as an evolving word web. *Proceed-*

ings of The Royal Society of London. Series B, Biological Sciences, 268(1485):2603–2606, December.

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

J.-P. Eckmann and E. Moses. 2002. Curvature of co-links uncovers hidden thematic layers in the world-wide web. *Proceedings of the National Academy of Sciences*, 99:5825–5829.

J.-P. Eckmann, E. Moses, and D. Sergi. 2004. Entropy of dialogues creates coherent structure in e-mail traffic. *Proceedings of the National Academy of Science*, 101:14333–14337.

P. Erdös and A. Rényi. 1960. On the evolution of random graphs. *Mat. Kuttató. Int. Közl.*, 5:17–60.

S. Evert. 2004. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, University of Stuttgart.

A. Fazly, R. North, and S. Stevenson. 2005. Automatically distinguishing literal and figurative usages of highly polysemous verbs. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 38–47, Ann Arbor, Michigan.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge MA.

R. Ferrer-i-Cancho and R.V. Solé. 2001. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1482):2261–2265, November.

O. Ferret and B. Grau. 1998. A thematic segmentation procedure for extracting semantic domains from texts. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, Brighton, UK.

O. Ferret. 2004. Discovering word senses from a network of lexical cooccurrences. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland.

M. Finkelstein-Landau and E. Morin. 1999. Extracting semantic relationships between terms: Supervised vs. unsupervised methods. In *Proceedings of the International Workshop on Ontologial Engineering on the Global Information Infrastructure*.

N. I. Fisher, T. Lewis, and B. J. J. Embleton. 1987. *Statistical Analysis of Spherical Data*. Cambridge University Press.

W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *DARPA Speech and Natural Language Workshop*, Harriman, NY.

S. Gauch and R. Futrelle. 1994. Experiments in automatic word class and word sense identification for information retrieval. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 425–434, Las Vegas, NV.

M. Girvan and M.E.J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826.

R. Gläser. 1998. The stylistic potential of phraseological units in the light of genre analysis. In A. P. Cowie, editor, *Phraseology - Theory, Analysis, and Applications*, chapter 3, pages 125–143. Clarendon Press Oxford.

M.A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Pro-*

ceedings of the 14th International Conference on Computational Linguistics, Nantes, France.

M.A. Hearst, 1998. *Wordnet: An Electronic Lexical Database*, chapter 5, Automated discovery of WordNet relations, pages 131–152. In Fellbaum (Fellbaum, 1998).

H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. 2001. The large-scale organization of metabolic networks. *Nature*, 407:651–654.

J.-U. Kietz, A. Maedche, and R. Volz. 2000. A method for semi-automatic ontology acquisition from a corporate intranet. In *Proceedings of EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, October 2000*, number 1937 in Springer Lecture Notes in Artificial Intelligence (LNAI).

A. Kilgarriff. 1992. *Polysemy*. Ph.D. thesis, University of Sussex, CSRP 261, School of Cognitive and Computing Sciences.

J. M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

D. Lin, S. Zhao, L. Qin, and M. Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1492–1493, Acapulco, Mexico.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI.

A. Maedche and S. Staab. 2000. Discovering conceptual relations from text. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 321–325.

C.D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

I.D. Melamed. 1997. Measuring semantic entropy. In *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics*, Washington, DC.

E. Morin and C. Jacquemin. 2004. Automatic acquisition and expansion of hypernym links. *Computers and the Humanities*, 38:343–362.

G. Müller. 1998. Beschränkungen für Binomialbildungen im Deutschen. *Zeitschrift für Sprachwissenschaft*, 16:5–51.

J.C. Nacher, N. Ueda, T. Yamada, M. Kanehisa, and T. Akutsu. 2004. Clustering under the line graph transformation:application to reaction network. *BMC Bioinformatics*, 5:207.

M.E.J. Newman. 2002. The spread of epidemic disease on networks. *Physical Review E*, 66.

G. Nunberg, I.A. Sag, and T. Wasow. 1994. Idioms. *Language*, 70:491–538.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.*

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - measuring the relatedness of concepts. In *Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, Boston, MA.

F. Pereira, N. Tishby, and L. Lee. 1993. Distributional clustering of English words. In

30th Annual Meeting of the Association for Computational Linguistics, pages 183–190, Columbus, Ohio.

D. Prescher. 2002. *EM-basierte maschinelle Lernverfahren für natürliche Sprachen*. Ph.D. thesis.

E. Prisner. 1995. *Graph dynamics*. Pitman Research Notes in Mathematics Series.

H. Pruscha. 1996. *Angewandte Methoden der Mathematischen Statistik*. B.G. Teubner, Stuttgart, 2nd edition.

R. Rapp. 2003. Discovering the meanings of an ambiguous word by searching for sense descriptors with complementary context patterns. In *Actes des cinquièmes rencontres terminologie et intelligence artificielle*, pages 145–155, Strasbourg, France.

P. Resnik. 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*, 11:93–130.

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124. Association for Computational Linguistics, Somerset, New Jersey.

B. Roark and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1110–1116.

M. Rooth, S. Riezler, D. Prescher, G. Carroll, and F. Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland.

I. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*, pages 1–15, Mexico City, Mexico.

S. Schulte im Walde. 2003. *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart. Published as AIMS Report 9(2).

H. Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

C. Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, (27):379–423, 623–656.

R. Sproat and J. van Santen. 1998. Automatic ambiguity detection. In *Proceedings of International Conference on Spoken Language Processing*, Sydney, Australia.

P. D. Turney. 2001. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167.

S. van Dongen. 2000. *Graph Clustering by Flow Simulation*. Ph.D. thesis, University of Utrecht, May.

J. Véronis. 2003. Hyperlex : cartographie lexicale pour la recherche d'informations. In

Actes de la Conference Traitement Automatique des Langues (TALN), pages 265–274, Batz-sur-mer, France.

D.J. Watts and S. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6):440–442, June.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan, August.

D. Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT-NAACL*. HLT-NAACL, Edmonton, Canada.

D. Yarowsky. 1993. One sense per collocation. In *ARPA Human Language Technology Workshop*, pages 266–271, Princeton, NJ.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Acknowledgements

Many people have helped me greatly with this thesis both scientifically and personally.

I am especially grateful to Dominic Widdows for his tremendous support and constant encouragement. It was wonderful to work with Dominic not only for his extensive knowledge and cool ideas but particularly for his kindness and contagious enthusiasm.

I would like to thank my thesis advisors Ulrich Heid, Hinrich Schütze and Christian Rohrer for their cheerfulness, many helpful comments and their genuine interest in my thesis research.

I am grateful to Stanley Peters for his kindness and for having made another stay at CSLI possible.

Many thanks to Jean-Pierre Eckmann, Elisha Moses and Danilo Sergi who were important collaborators on this thesis project and contributed many brilliant ideas. It was them who introduced Dominic and me to the concept of graph curvature which has become central to this thesis, and who originated the curvature clustering scheme. Thank you for the kind hospitality in Geneva and Rehovot. I very much enjoyed both the scientific and the social part of my visits.

The friendly and cheerful atmosphere at the IMS made me feel welcome right from the beginning. In particular, I would like to thank Tina Säuberlich, Sabine Dieterle, Inga Frese, Kristina Spranger, Giusy Rota, Sabine Schulte im Walde, Arne Fitschen, Christian Hying, Arndt Riester and André Blessing for their friendship and distractions in the form of coffee breaks, movies, parties, swims, runs and climbs.

Most thanks go to my husband Bernd for supporting me with his unlimited and gentle patience, and for cheering me up by turning mountains into molehills. It is wonderful to be working on a new "project" together: our now four and a half month old daughter Anna Noemi who is incomparably much more exciting and exhilarating than any scientific endeavor.