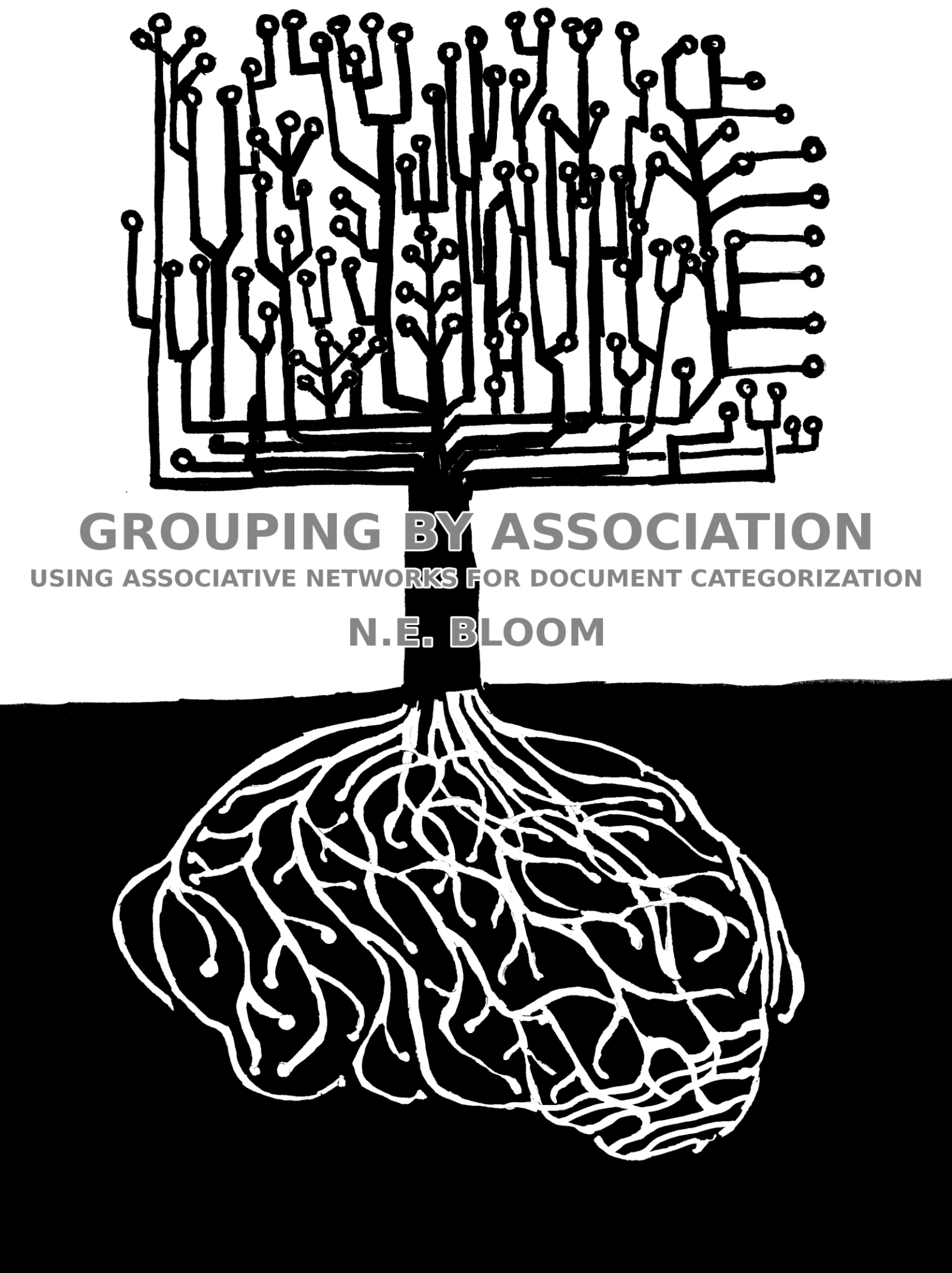

GROUPING BY ASSOCIATION

USING ASSOCIATIVE NETWORKS FOR DOCUMENT CATEGORIZATION

N.E. BLOOM

GROUPING BY ASSOCIATION:

USING ASSOCIATIVE NETWORKS FOR DOCUMENT CATEGORIZATION

N.E. BLOOM

PhD dissertation committee

Chairman

Prof. dr. P.M.G. Apers (University of Twente)

Secretary

Prof. dr. P.M.G. Apers (University of Twente)

Supervisor

Prof. Dr. F.M.G. de Jong (University of Twente)

Co-supervisor

Dr. M. Theune (University of Twente)

Members

Dr. Djoerd Hiemstra (University of Twente)

Prof. dr. Theo Huibers (University of Twente)

Prof. dr. Antal van den Bosch (Radboud University)

Prof. dr. Paul Buitelaar (National University of Ireland, Galway and University of South-Africa)

Referee

Dr. Dolf Trieschnigg (MyDataFactory, Meppel)

The research presented in this work was supported by:

pagelink COMMIT/

Cover Design by C.C.F. Bloom-Berendse

ISBN: 978-90-365-3878-7

GROUPING BY ASSOCIATION:

USING ASSOCIATIVE NETWORKS FOR DOCUMENT CATEGORIZATION

PROEFSCHRIFT

ter verkrijging van

de graad van doctor aan de Universiteit Twente,

op gezag van de rector magnificus,

prof. dr. H. Brinksma,

volgens besluit van het College voor Promoties

in het openbaar te verdedigen

op woensdag 10 juni 2015 om 12:45 uur

door

Nicolaas Emmanuel Bloom

geboren op 17 juli 1983

te Alphen aan den Rijn

This dissertation is approved by:

Prof. Dr. F.M.G. De Jong

Dr. M. Theune

"There is no room for '2' in the world of 1's and 0's, no place for 'mayhap' in a house of trues and falses, and no 'green with envy' in a black-and-white world." - Ravel Puzzlewell [1999]

Abstract

In this thesis we describe a method of using associative networks for automatic document grouping. Associative networks are networks of ideas or concepts in which each concept is linked to concepts that are semantically similar to it. By activating concepts in the network based on the text of a document and spreading this activation to related concepts, we can determine which concepts are related to the document, even if the document itself does not contain words linked directly to those concepts. Based on this information, we can group documents by the concepts they refer to.

In the first part of the thesis we describe the method itself, as well as the details of various algorithms used in the implementation. We additionally discuss the theory upon which the method is based and compare it to various related methods.

In the second part of the thesis we evaluate techniques to create associative networks from easily accessible knowledge sources, as well as different methods for the training of the associative network. Additionally, we evaluate techniques to improve the extraction of concepts from documents, we compare methods of spreading activation from concept to concept, and we present a novel technique by which the extracted concepts can be used to categorize documents. We also extend the method of associative networks to enable application to multilingual document libraries and compare the method to other state-of-the-art methods for document grouping.

Finally, we present a practical application of associative networks, as implemented in a corporate environment in the form of the Pagelink Knowledge Centre. We demonstrate the practical usability of our work, and discuss the various advantages and disadvantages that the method of associative networks offers.

Preface

My first serious step from following courses towards doing actual research was made when I worked with Joost Vromen on a project supporting Ivo Swartjes' research into a virtual storyteller [Swartjes et al., 2007]. In the project, we provided a model to generate creative solutions using case-based reasoning for situations in the story, heavily based on a work by Turner on computer creativity [Turner, 1994]

When the project had finished, I was forced to lay my focus elsewhere for a while to finish my studies, but the idea that computers could generate creative ideas was something that resonated with me, especially because it was so counter-intuitive with regards to what people expect of computers.

Between the positive experience I had with the project as well as the research I did for my Master's thesis and during my internship, I was certain I wanted to continue doing research in the field of Artificial Intelligence, and to try and get a PhD. Searching for opportunities to do just that while working as a programmer at Pagelink to pay the bills, I soon found that most job vacancies that offered this kind of chance were associated with specific projects for which funding had already been secured. As such, I wasn't just looking for someone to hire me, but also for a topic of research that would be interesting for me, not in the least because once started, I would be stuck with it for the next couple of years.

While I searched for an opportunity I spoke with Henk Kok, the head of our company, both about my plans to get a PhD and about research I had done earlier on case-based reasoning. After going back and forth on the topic some, we realised that not only was this a very interesting topic of research, but it was also something that could greatly benefit Pagelink, and he very generously offered me the option to do my research within the company.

Having found both a place and a topic, I now needed a supervisor and a promotor. Mariët Theune, who had supported my earlier efforts with Joost Vromen and Ivo Swartjes as well, was kind enough to take up this task once again with Anton Nijholt acting as promotor. Over the years, of course, certain things have changed. Work on case-based reasoning evolved into associative networks, Anton Nijholt retired and Franciska de Jong – my new promotor – got me involved in COMMIT, a public-private research community in the Netherlands.

But while I had been quite sure that the future would bring these types of changes, there was no way I could have known at the time that the topic which we had worked on all those years ago would be the first steps on the road of this far larger and longer project.

Acknowledgement

The research presented in this thesis has been funded by Pagelink. I am very grateful for their support and I count myself as extremely lucky to have been given a chance not just to work on this research, but to be able to do so by continuing work on the topic that drew me into research in the first place.

I would additionally like to thank Anton Nijholt, Franciska de Jong and Mariët Theune for making my work possible and for all the support they have given to it. Special thanks go to Rieks op den Akker, Lynn Packwood and the members of the dissertation committee, especially Djoerd Hiemstra and Antal van den Bosch, for their reviews of my work.

Finally, I would like to thank my colleagues, friends and family for their continuous support and interest.

Contents

List of Figures

List of Tables

Part I

Basics

Chapter 1

Introduction

> *"When we talk mathematics, we may be discussing a secondary language built on the primary language of the nervous system."* - John von Neumann, as quoted by Oxtoby et al. [1958]

John von Neumann, a major contributor to fields like mathematics, statistics and computer science [Halmos, 1973], observed that mathematics can be thought of as a different language from the language of the nervous system, that is, the system by which the human brain naturally interprets information.

Von Neumann effectively painted a picture of how mathematics is a coding system that we learn on top of the natural way we understand the world. In Figure 1.1, we show how the language of the nervous system (represented by a network) underlies the way people think and how the language of mathematics is built on top of it. Additionally, we show natural language, by which people can express the ideas that arise from thoughts in the language of the nervous system. The interplay between these three 'languages of the mind' has been a pillar for the design of the document categorization system that will be presented in this thesis.

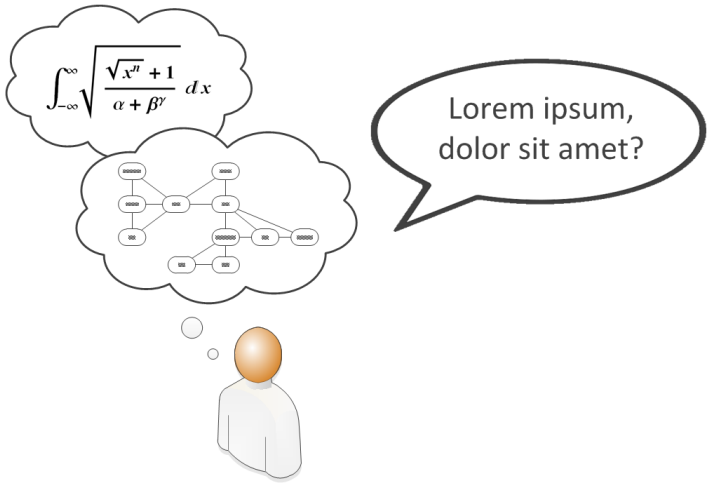

Figure 1.1: The Languages of the Mind

1.1 The Languages of the Mind

In this section we examine the languages of the mind, specifically the three languages displayed in Figure 1.1, that is, the two mentioned by Von Neumann, as well as natural language.

1.1.1 Natural Language and the Language of the Nervous System

A key feature of the language of the nervous system is that it is closely linked with spoken language, to the point where as we learn to speak, our own internal thoughts become verbalized as the 'voice inside our head' [Vygotsky et al., 2012]. By extension, this also allows the language of the nervous system to be used to learn written language, which, though different in some aspects, is fed by the same grammar, vocabulary and conceptual model of the world as spoken language [Halliday, 1989].

We posit that association – the mental connection or bond between sensations, ideas, and memories [Merriam-Webster, 2014] – underlies the way in which people understand the world, as well as the way they naturally use and understand language.

In growing up, children experience a multitude of stimuli. They see, hear and touch things they never encountered before. Soon, patterns become apparent in these experiences, such as the sight and scent of the child's parent, which often seem to be related to the

taste of food, or comfort [Stifter et al., 2011]. As the child matures, the associations it makes become more complex: it learns that the sound of a certain word, for example, *mama* or *papa,* is associated with the sight, sound and scent of a specific parent. Then more such patterns develop, which the child can use to make ever more complex choices and interactions with the world. The associations thereby become the basics of language [Bochner and Jones, 2008], a maturation one might argue based on the way in which it develops in humans, of the underlying language of the nervous system.

However, the associations go further than merely matching one object or sensation to another or a word to a concept. The child does not just learn that the word *mama* refers to a specific individual. The corresponding concept, that is more or less language-independent, but that for sake of simplicity and in accordance with scholarly conventions we will refer to as MAMA, in turn comes with a multitude of additional associations, such as food, comfort and protection for which the child may learn the words. Thus, associations carry from word to concept to other concepts and back to words. Of course it is clear that despite the link between the word *mama* and the concept FOOD, the word *mama* itself does not refer to food, yet there is still a clear association between the concept of MAMA and the concept of FOOD, even to the point of leading the child to go to their mother when they are hungry.

Different children may have different experiences and thus their associations will be different. For example, one child may hear the word *chocolate milk* and remember many wonderful times drinking hot cocoa and spending time with the family, evoking a positive sentiment, while another may have a much more negative connotation, having suffered burns from a spill for example or simply disliking the taste. Thus, through different experiences, associations between concepts may differ between individuals, even if both individuals agree on the basic object that the words represent. In effect, though the words in spoken language are the same between individuals, the representation of the concept in the language of the nervous system is slightly different for each individual.

1.1.2 The Language of Mathematics

When learning the language of mathematics, we have to adjust to a different vocabulary and syntax which differs from the natural language we speak. In the language of mathematics, variables and symbols represent concepts or operations that cannot always be easily captured in natural language, and it has a different grammar formed by equations and functions that may manipulate those variables according to specific rules. Just as English cannot be translated into Dutch by simply replacing individual words with their direct translation [Bassnett, 1980, Nes et al., 2010], so too mathematics requires more than merely learning the meaning of its symbols. Through learning mathematics, we learn another way of thinking and interpreting the world.

That different way of thinking, supported by symbols and equations, allows us to tackle problems which we could never have hoped to solve without mastering mathematics. With the aid of mathematics, we can describe things such as the positions of the planets, the way particles interact and even the way in which an apple falls from a tree with great accuracy [Newton, 1687].

However, it would go too far to say that mathematics is simply a superior system for describing the world in general. Many things which are easy to understand through the language of the nervous system are very difficult to capture with a mathematical definition. For example, it is nearly impossible to use mathematics to describe such things as the Amazon River, the personality of Nicola Tesla, or even something as fundamental and universal - in terms of our nervous systems in any case - as love (see Figure 1.2).

The saying that *"computer science is no more about machines than astronomy is about telescopes"* is attributed to Edsger Dijkstra [Haines, 1993]. This statement, though intended to describe computer science as a field of mathematics, also echoes the sentiment that computers themselves are machines which express mathematics. One might even say that for computers, mathematics is a primary language in the same way that humans have a primary language of the nervous system, both engrained in their respective hardware.

Realising the advantages that humanity has gained by mastering the language of mathematics on top of the language of the nervous system, one might wonder if it would be possible for computers to have something like a language of the nervous system built on

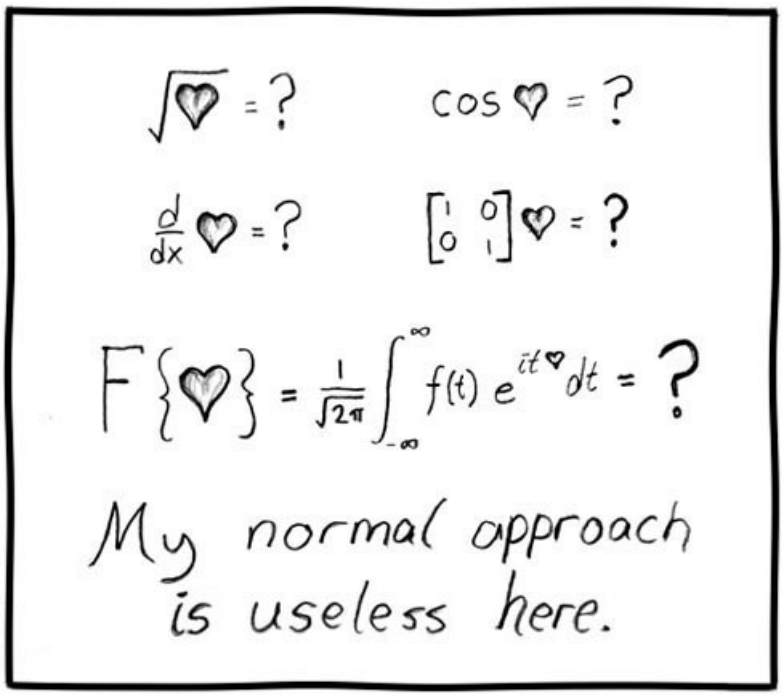

Figure 1.2: Useless - *Image from xkcd* [Munroe, 2006]

top of the language of mathematics as displayed in Figure 1.3, and what new advances and understanding this could bring to the world. In this thesis we will, as many others before us, try to take a step towards creating this language.

1.2 Contrast between Association and Mathematics

Natural language and the language of the nervous system are closely linked, the former being used to express ideas from the latter so naturally that it is even reflected in the words *natural language* themselves. A strong contrast can be seen, however, between the language of the nervous system and the language of mathematics. To examine this contrast, we will first look at how the so-called Problem of Universals [Klima, 2013] may highlight a difference between the language of the nervous system and the language of mathematics,

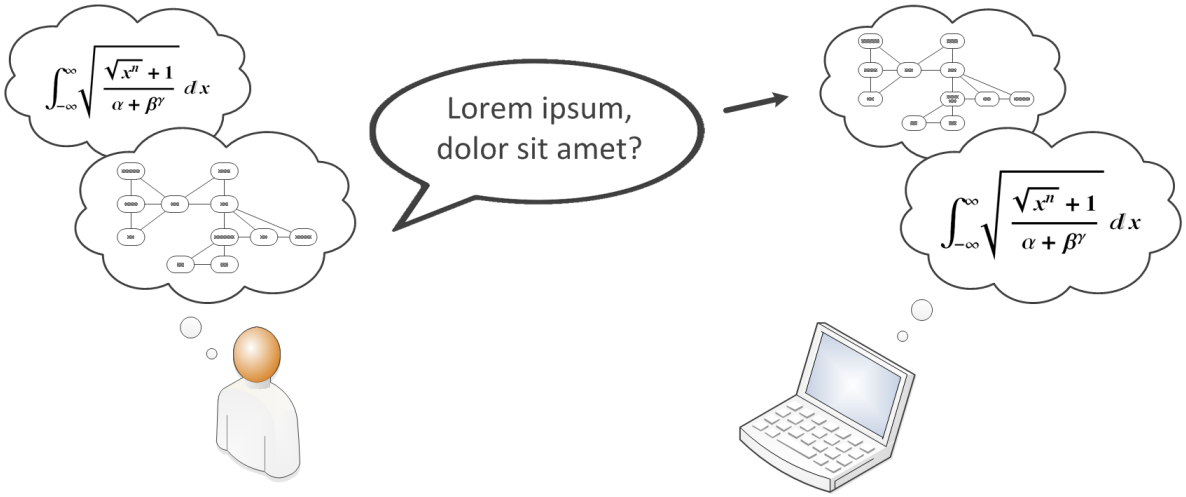

Figure 1.3: Languages of Computers

and then examine how various paradoxes may arise when the two languages interact. Finally, we will examine the consequences that these differences have for natural language processing by computers.

1.2.1 Problem of Universals

The idea expressed in the previous section that children may have different experiences with certain concepts such as CHOCOLATE MILK, touches on the philosophical Problem of Universals [Klima, 2013], which asks whether concepts such as WARM and BROWN actually exist, and whether it is even possible to speak universally about singular objects and their properties [Quine, 1964]. The Problem of Universals defines concepts like WARM as qualities that two or more entities have in common, and those various kinds of concepts or properties are referred to as universals. It asks, for example, how we can know that all possible chocolate milks are brown when we can observe only a limited number.

Plato and Aristotle, two Greek philosophers who pondered the problem, each gave their own interpretation of this problem. Aristotle interpreted concepts as consisting of the experiences individuals have with specific instances of those concepts. Thus, Aristotle thought that ideas like chocolate milk for one person were formed from all the chocolate milk they had ever seen or heard of, while the concept of chocolate milk for another person

was likewise formed by all the chocolate milk that the other person had ever seen or heard of [Scaltsas, 1994]. Plato in contrast believed that there is a single, perfect concept of things like chocolate milk [Churchland, 2012]. In his view, our personal experiences have no impact on what chocolate milk is. The abstract idea of chocolate milk exists, in Plato's view, independent of our experience with it, and real world examples are merely imperfect incarnations of this abstract idea.

Compared to the language of mathematics, the language of the nervous system is better equipped to deal with fuzzy relationships and generic patterns that may or may not hold, as it closely resembles Aristotle's ideas: a person's associations with CHOCOLATE MILK are based on that individual's experiences with it including the ones that are not universal. But even with Aristotle's model of different associations, the basic physical properties of concrete objects generally remain the same. Though everyone has different experiences with chocolate milk, people understand that it is a brown liquid you can drink, even if it evokes different feelings for them.

Where the Aristotelian view can be described as associative, Plato followed a view that more closely resembles a language of mathematics-based interpretation. He would say that there is a pure description of CHOCOLATE MILK to which various samples can be compared to determine if they are chocolate milk. Plato's approach mirrors the one used to program computers with the ability to reason about the world. In this approach researchers try to capture the basic properties of concepts, such as CHOCOLATE MILK being a brown liquid you can drink, in a computer model. As an example of this, Lenat et al. [1995] created a common sense database called Cyc. In this database, Lenat et al. describe in detail the properties of a large number of concepts, including CHOCOLATE MILK, thereby creating a model of the world similar to Plato's interpretation of universals. The Knowledge Vault [Dong et al., 2014, Hodson, 2014], a project by Google to automatically collect facts from the internet, might also be considered an example.

A limitation of the language of mathematics, especially as implemented on computers, is that everything is represented by binary values. Concepts are encoded in absolutes, in series of ones and zeros. This corresponds to the assumption that something either has a certain property or it does not have that property. That it belongs to a certain group or that it does not. Following that absolute logic, fringe associations have no representation

in the logical language of mathematics. While there may be some link between MAMA and FOOD for example, if we use that link as a basis for reasoning within the language of mathematics, it must always hold true. But the concept of MAMA does not necessarily involve FOOD. A more complicated model which describes when MAMA is linked to FOOD would require detailing every possible situation in which this link holds true, for example in the form of an exhaustive description of the specific times and places where the two are found together. With an exponential number of possible pairs of objects, such a model would quickly become impossible to express in a finite definition, even if it was possible to describe each individual relation in a complete and accurate manner. Thus, the absolute mathematical description runs into problems with fringe associations, edge cases and fuzzy borders. Those problems cause Lenat and others following the same philosophy to meet with limited success.

Several attempts have been made to overcome the problems and limitations that these absolute, binary values bring. Fuzzy logic [Hájek, 1998, Turunen, 1999, Novák et al., 1999] maintains the binary values, but allows for the assigned truth values to be between zero and one. Many-valued logic [Cignoli et al., 2000, Gottwald, 2001] is another attempt which uses more than two truth values (but otherwise maintains these as absolute). Many other attempts have been made as well, with Schank and Abelson [1977] being an important attempt in relation to our work (see also Chapter 4). Many of these techniques took off in the 90's due to the increasing availability of processing power, memory and as a result digital data sources to support such work. To simplify the illustration of the difference between the associative language of the nervous system and the logical language of mathematics, we skip over these approaches for the moment.

As said, the languages of the nervous system and mathematics each have their characteristics and capabilities. Logic allows the establishment of absolute truths based on reasoning - it can be used to draw conclusions that hold true in all situations. It is limited in that it cannot easily incorporate properties that are 'sometimes' true or that are hard to define with absolute values, such as tall, rich or old. Russell [1923] argues that being intrinsically hard to define in fact holds for all terms with 'vague' definitions. We might even say that whether someone is tall, rich or old depends at least in part on the past experiences

of the observer. This would bring us back to the associative, Aristotelian perspective on universals.

1.2.2 Logical Paradoxes

We can also illustrate the contrast between the language of mathematics and the language of the nervous system using paradoxes such as Heraclitus' river [Graham, 2011], which occur when the descriptions of reality for both languages deviate.

Heraclitus famously stated that *"no man ever steps in the same river twice"* (attributed to him by Plato [350 BCE]), claiming that the river is always changing and never remains the same, therefore one cannot step in the same river again. Viewed through the language of mathematics, this holds true - since the river has changed, the values of its variables are different and even a very small difference mathematically implies a divide. Take a cube, for example, which has a very exact definition [Weisstein, 2015]. If one of the vertices is changed, even if only by a very small margin, then the object is no longer a cube, and all rules such as the way to calculate its volume or surface area cease to apply. This idea that even a minor change makes an entirely different object goes against the common way in which people understand and interact with things such as rivers and cubes as relatively constant, continuous objects.

Another paradox that can help to illustrate the divide between the two points of view is the Ship of Theseus paradox:

> *"The ship wherein Theseus and the youth of Athens returned [from Crete] had thirty oars, and was preserved by the Athenians down even to the time of Demetrius Phalereus, for they took away the old planks as they decayed, putting in new and stronger timber in their place, insomuch that this ship became a standing example among the philosophers, for the logical question of things that grow; one side holding that the ship remained the same, and the other contending that it was not the same."* - [Plutarch, 75]

Many variations of the riddle referenced by Plutarch exist, and the question raised – whether the ship remains the same if every piece of it is replaced – has kept philosophers busy for centuries, and various solutions have been proposed.

Following the earlier logic of Heraclitus' river, the ship would not be the same if any part of it had changed. Aristotle [350 BCE] argued that objects have different causes, and according to his arguments, the ship is the same because its design and purpose remain the same, even if it is no longer composed of the same parts. Another common argument is that the paradox occurs because there are different definitions of "the same": a distinction is made between qualitatively identical ("gelijk" in Dutch) where the ship continues to hold the same properties, and numerically identical ("zelfde" in Dutch), where the ship is identical only to itself. Sider [2003] argued that objects like the Ship of Theseus are four-dimensional, and that our perceptions at different moments are merely slices of the greater object. Thus, while the three-dimensional composition may be different at different times, the Ship of Theseus remains the same object in a four-dimensional view.

To explain the discrepancy between the two views highlighted by the Ship of Theseus and Heraclitus' River paradoxes, we posit however, that association underlies the way in which people use natural language, that is, association is at the core of the language of the nervous system.

Many paradoxes and riddles similar to these two exist and their variations are endless. A notable variation of the Ship of Theseus paradox is George Washington's axe [Browne, 1982], and a more modern example is the case of the Sugababes, a British band formed in 1998, which lost its founding members over the years, to be replaced by new ones one by one until in September 2009, none of the founding members remained, each having been replaced by new members; the three original members reunited in 2011, with the original Sugababes still in existence. The group of the original band members now goes under the name Mutya Keisha Siobhan (which is formed from the names of the original band members), and several lawsuits have been fought over the use of the name 'Sugababes' [Bray, 2012]. For our own work, the question whether the content of a text remains the same if individual words are replaced by words covering the same semantic concept is especially relevant.

It is important to note that these paradoxes only occur through the contrast between the associative language of the nervous system and the logical language of mathematics. Even though the associative understanding limits how accurately something like a ship can be defined, and even though concepts have different associations for different people, we can

still have a collective understanding of what a concept represents in the real world and what some of that concept's basic properties are. One example to demonstrate this is Heraclitus' river. Expressed logically as a specific formation of water, the river is never the same. Yet a fisherman sailing up and down the river, and a merchant crossing the river with a wagon on a bridge, both understand the associative idea that the river is a volume of water that flows through a bedding from the mountains to the sea. Though the merchant and the fisherman have different experiences of the river (a source of income versus an annoying obstacle along the way to the market perhaps), the associations do not change because the general configuration of the molecules of water in the river has been altered. From an associative perspective, the river today and the river tomorrow are basically the same thing. Over time, associations can change, but there is no clear boundary beyond which something turns into a different river as there would be when modelling the world using logic.

This is also where the Ship of Theseus paradox hails from. From a logical perspective, to describe the ship, one would need to establish what exactly the Ship of Theseus is. But from an associative view, this is not necessary at all. The ship remains the Ship of Theseus, even if every plank is replaced. In fact, even if the ship was fully burned down and rebuilt from scratch, associatively it would still be the Ship of Theseus. This kind of perspective may seem unimaginable from a logical point of view, but an example of this associative view was in fact described by Douglas Adams in Last Chance to See [1990]. When he pondered the paradox, he wrote:

"I remember once, in Japan, having been to see the Gold Pavilion Temple in Kyoto and being mildly surprised at quite how well it had weathered the passage of time since it was first built in the fourteenth century. I was told it hadn't weathered well at all, and had in fact been burnt to the ground twice in this century.

"So it isn't the original building?" I had asked my Japanese guide.
"But yes, of course it is," he insisted, rather surprised at my question.
"But it's burnt down?"
"Yes. Many times."
"And rebuilt with completely new materials."
"But of course. It was burnt down."
"So how can it be the same building?"
"It is always the same building."

I had to admit to myself that this was in fact a perfectly rational point of view, it merely started from an unexpected premise. The idea of the building, the intention of it, its design, are all immutable and are the essence of the building. The intention of the original builders is what survives. The wood of which the design is constructed decays and is replaced when necessary. To be overly concerned with the original materials, which are merely sentimental souvenirs of the past, is to fail to see the living building itself." - [Adams and Carwardine, 1990]

The associative understanding of the building that Adams explains is internally consistent. It just does not match with how we would understand that building through the logic-based language of mathematics.

1.2.3 Computers and Natural Language Processing

In ancient times the paradoxes described in the previous section were curiosa to occupy philosophers, never having any real impact on the world itself beyond the fields of philosophy and linguistics. However in our modern computer-driven era, the tension between the associative language of the nervous system and the logical language of mathematics from

which the paradoxes stem is often perceived as an obstacle, as it limits what computers can do, especially when it comes to dealing with the associative world of human language. As said, several efforts have been made throughout the past centuries to resolve the tension, for example by describing the definition of terms in natural language more precisely [Russell, 1923, Quine, 1981], by using fuzzy logic [Goguen, 1969] and by using many-valued logic [Weber and Colyvan, 2011], but the core differences between the perspectives were never bridged, and computers have not mastered the language of the nervous system.

Interestingly enough, some scholars have even gone as far as to literally try to find a shortcut to the nervous system through Brain-Computer Interfaces [Vallabhaneni et al., 2005]. Apparently this route may be the only one that is feasible for people with impaired physical conditions that block their speaking or motoric capabilities, and it would be very interesting to understand the role of association in such scenarios, but that goes well beyond the scope of our work here.

As said, one of the places where the divide is especially obvious is in situations where natural language is processed automatically. A primary reason why the divide shows up here is that most of our understanding of language is associative, rather than logical and as a result computers have trouble with it. That is not to say computers cannot produce useful results. Systems do exist to search texts, extracting topics and grouping documents, but these are usually data-driven or based on detailed information provided by human experts beforehand. Those methods give limited insight into the intended meaning of the text itself.

It is illustrative of the divide that conceptually simple tasks such as search, topic extraction and document grouping are the ones computers can handle, while more complex language tasks such as finding precedents for legal cases or translating jokes and poetry remain beyond the capacity of computers. Humans perform these tasks quite successfully, even if they do not process information at the same pace as computers. While computers are faster for tasks involving natural language, they have trouble in terms of accuracy. Most state-of-the-art solutions in the field of natural language processing rely on statistical approaches [Manning and Schütze, 1999], for example to predict the chances that a certain document belongs with a certain group. Rather than using knowledge regarding the text and language itself, the computer is basically making a ‘best guess’ effort to find which meaning is the most likely. These probabilistic techniques can be said to resemble the

probabilistic model of a coin toss. We understand very well that about 50% of the coins will land head and about 50% of the coins will land tails, but this is a far cry from claiming an understanding of how air resistance, gravity and the material and shape of the coin itself impact the way it will land. Likewise, while probabilistic models used in natural language processing may be effective for specific situations (and we cover them in more detail in Chapter 7), such models do not provide a deeper understanding of the underlying concepts expressed in the words they analyse and as such, these models do not represent a secondary language of the nervous system for computers.

It is our hypothesis that if we want to progress beyond the current ability to handle textual data and to improve the quality of text processing, we need to look for methods that provide a greater understanding of language using associative thinking. Such methods would eventually make natural interaction with computers as easy as it is between humans. These methods should be expressed in a logic-based manner so that they can be used by computers, effectively building the secondary language of the nervous system on top of the primary language of mathematics as used by computers, as we suggested at the start of this chapter. If computers are provided with a way to model associative thinking, they become able to mimic the way humans process language and information and thus should be better able to handle the complex language processing tasks mentioned above. Models of associative thinking can become the pillar of a bridge that crosses the divide between associative and logical thinking.

We propose to adopt the concept of associative networks – that is, networks of concepts in which each concept is linked to concepts that are semantically similar to it – and model language by using the way people understand the concepts expressed through language as a guideline, while still using the language of mathematics so the model can be applied by a computer. Our hypothesis is that if we want to progress beyond the current ability to handle textual data, we need to look for methods that provide a greater understanding of language using associative thinking. To investigate whether this hypothesis can be sustained, we created and tested a model based on associative thinking which should help computers to acquire some capacity of association and reach over the divide.

1.3 Automatic Document Grouping

With the rise of the Internet, more information has become accessible to individuals than at any other point of time in history, so much in fact that no human is able to process all that information. The success of companies like Google illustrates how strong the need to find specific information in the ocean of data has become [Basu, 2007]. To help us organize all this information, that ocean of data has to be structured and indexed in such a way that if we describe what we are looking for, we are able to find a link to the relevant information. The desired structure needs to be accurate and complete to make sure it is helpful in retrieving the information that was actually requested (and nothing else).

In this work, we will focus on automatic document grouping, which can be a significant aid in the endeavour to assign structure to the ocean of data. By grouping documents containing similar information together and correctly labelling these groups, it will be possible to access the indexed information more easily while we can omit information we do not need. Proper grouping allows users to find documents concerning a specific topic even if they are unsure of the common terminology, and makes it easier to browse all information related to a topic in its entirety.

1.3.1 Bridging the Divide between Association and Mathematics

One possible reason why document grouping is so difficult for computers is that people may describe the same concept in different ways and may use the same word to describe different concepts. These phenomena, known as ‘synonymy’ [Quine, 1951] and ‘ambiguity’ [Ravin and Leacock, 2000] respectively, are two of the key factors in the complexity of natural language interpretation that humans seem very well equipped for, in contrast to the systems designed for the automatic processing of natural language. As a result of ‘synonymy’ and ‘ambiguity’, the grouping we seek would ideally be based on the concepts discussed in the document, rather than the specific words used in the document. The criteria for groupings by topic cannot be easily caught in strict logical rules. This problem is sometimes referred to as the semantic gap [Ehrig, 2006]. An alternative way of capturing the essence of this limitation is by comparing it to the Ship of Theseus paradox, and asking

whether a document in which every word is replaced by a different but semantically similar word still covers the same topic.

A key advantage that association has over formal logic becomes clear in the following scenario: two individuals – despite speaking the same language – use very different words to describe the same object, with each using the idioms they are familiar with. For example, a racing aficionado may describe a vehicle as a *Formula 3 Porsche* while a layman may simply describe it as a *racing car*. When the layman says *racing car*, the racing aficionado may not be clear on what type of *racing car* they are talking about (which to them would not be a trivial difference). Likewise, the layman may have never heard of a *Formula 3 Porsche*. Despite this difference in vocabulary used, through association with other words in the context such as references to driving, winning or racing, humans are quite capable of coming to a shared understanding about something. This allows the racing aficionado and the layman to realize what the other is describing.

If we can understand and harness the human ability to recognize concepts despite different words being used to describe them, we will be able to use that understanding as a stepping stone for bridging the perspectives of logic-based computer processing and associative language. Harnessing this human ability granted by our primary language of the nervous system will in turn allow computers to use techniques based on the human understanding of the documents they are grouping (in a sense modelling the human understanding), thereby going beyond logic-based techniques.

1.3.2 Real World Application

As a real world scenario for which the task of document grouping is relevant, one could think of a business that produces large libraries of documents on the products they produce, including technical manuals, sales brochures, questions by clients and answers to those questions and much more. For obvious reasons, a brochure made with the goal of selling a car will describe the car in a different way and will use different words than the technical documentation intended for the mechanic who has to repair it, so even though they cover the same topic, the words in the document will be different.

Pagelink – the company sponsoring the research reported in this thesis – provides business automation solutions to large companies that often have such large libraries of documents. Helping employees of these companies to access the information that has been produced in the past can be an important functionality of the automation solution, and products using this research such as the Knowledge Centre (see Chapter 12) aid in gaining such access.

With companies producing ever more information[1] the document collections used by companies can be very large and it would require a lot of manpower to group these documents manually. Labelling can offer some relief but for large libraries, labelling too can demand an extensive effort. As the model of associative networks proposed here can be put to work automatically it could make such investments of time and money unnecessary.

However, the automation of document clustering is not enough to satisfy all of our requirements: the groupings created should have the quality of a manual grouping and allow knowledge to be shared widely within the company, even between different departments. Returning to our earlier example, by using a system based on the approach advocated here, a sales manager should be able to find technical documents for a customer, even if that sales manager is not familiar with the proper technical terms that specialist engineers use in this documentation. Based on the benefits described earlier, an associative model based on the language of the nervous system as we have proposed would provide all of these benefits.

Moreover, the libraries of documents used in large corporations are often dynamic and any real world application needs to be fast enough to be able to deal with frequent adding, removing and editing of documents in the library (see also Chapter 12).

Finally, many large companies operate internationally and therefore do not limit themselves to documents in a single language. Thus, it would be highly desirable for a system based on our research to be able to handle documents in multiple languages. All in all, automatic document grouping is an increasingly relevant and complex task.

[1] Up to 90% of all data in the world may have been generated in the past two years [Dragland, 2013]

1.3.3 Proposed Solution

Associative networks, as mentioned, are networks of concepts in which each concept is linked to concepts that are semantically similar to it. To create such a structure, a source that provides these semantic relationships is needed. We have developed a method to create associative networks based on commonly available sources such as WordNet and Wikipedia and a method to use the links between concepts within an associative network to automatically group documents based on the words those documents contain.

Our method extracts words from the documents in a library, translates them to concepts, finds concepts that are semantically similar to the concepts extracted from the document using the links in the associative network and then compares the expanded set of concepts related to the document to discover which other documents cover the same topics. We then group the documents based on these results.

By using this approach, we can accurately model the associations that humans have with the words in documents based on the language of the nervous system. We hypothesize that this in turn will provide us with better document grouping results than other models, as we allow computers to 'think' more like humans.

1.3.4 Research Questions

The issues mentioned above raise many questions, both in terms of the grander goal of bridging the gap between the associative language of the nervous system and the logical language of mathematics, and more practically in terms of the real world application of our work. The latter is our primary focus, discussed in parts I and II of this thesis, and especially in Chapter 12, while the manner in which this all fits into the grander goal is discussed in the rest of Part III. Our research questions are the following:

1. *How can an associative network be created such that it does not require a large amount of manual configuration?*

2. *How can the connections in an associative network be trained to accurately represent the associations between the concepts modelled in the network?*

3. *How can the input used by an associative network be improved by using Natural Language Processing?*

4. *How can a model of associative networks be used to automatically group documents?*

5. *Which methods can improve the groupings created by associative networks, and can those methods provide additional insights into the structure of associative networks?*

6. *How can an associative network be expanded to handle document collections with documents in multiple languages?*

7. *What lessons can be learned by applying associative networks in a real-life knowledge management platform?*

In this thesis we will describe a number of experiments, each focussed on answering one or more of the research questions posed in the previous sub-section. Each of the experiments covers parts of the process we developed for automatic document grouping. By focussing on these individual parts of the system, we can establish the usefulness of those parts and validate that they contribute to the full process. As each experiment focusses on a specific problem, we can evaluate specific aspects of our associative solution by comparing them to other methods, either to show that our method performs on a par with or better than the state-of-the-art techniques or to show the improvements on our original versions. By combining the results of each step, we can support our conclusions about the system as a whole.

After discussing the results of the individual experiments, we will give a more general analysis involving all of the experiments together, revisiting the integrated process we developed and linking it back to the language of the nervous system and the language of mathematics which we covered in this chapter.

Several of the experiments mentioned in this thesis were previously published; for these experiments the original publication is explicitly mentioned at the start of the chapter in which the publication text has been integrated.

1.4 Thesis Overview

This thesis is divided into three parts. In Part I we will cover the basics of our method, discuss some of the general theory and cover related work. Specifically, in Chapter 2 we will describe in general terms the overall method we have developed and implemented to automatically group documents using associative networks. The different steps which are involved in turning an unordered collection of documents into structured categories will be outlined, and we will go into more detail about these steps in Chapter 3. In Chapter 4 we will look at other solutions to the problem of document grouping as well as work related to the various parts of the methodology we have developed.

In Part II, we will present the various experiments performed. In each of the chapters we will describe some of the theory behind the step covered in more detail to give context to the experiment. We will then describe the experiment itself, present the results and draw some conclusions about the specific parts of the process – and how to improve upon them.

The experiments will be presented in order of their place within the entire document categorization process. Following the steps of our process (described in the next chapter), we will start with the creation (Chapter 5) and training (Chapter 6) of associative networks. In Chapter 7 we will describe our work on Natural Language Processing which we have used to refine the information from documents. In Chapter 8 we will describe how our method of association concentration uses associative networks to extrapolate additional information from texts being categorized. In Chapter 9 we will go into the topic of Power Graph Analysis, which can aid in the categorization of documents as well as provide insights into the quality of associative networks themselves. In Chapter 10 we will describe how we can make associative networks capable of handling documents in multiple languages. Finally, in Chapter 11 we will examine the performance of the entire process, comparing it to state-of-the-art methods.

In Part III we will describe our application of associative networks in a real-life system, the Pagelink Knowledge Centre (Chapter 12). In Chapter 13 we will discuss the insights acquired and describe the strengths and weaknesses of our method. Finally in Chapter 14, we will return to the questions raised in this chapter, draw our conclusions and discuss future work.

Chapter 2

Automated Document Grouping

"Step by step, it's all up to you. Then pretty soon you'll show the whole wide world, you made something new!" - LazyTown's Stephanie [2004]

In this chapter we provide a general overview of the pipeline we have designed and implemented to automatically group documents. We describe each of the steps in general terms before we examine them in more detail in future chapters. Figure 2.1 shows the 5 steps involved in this process. First we create an associative network based on a source from which we can extract relations between concepts (Step 1). Next, we train an associative network based on documents from a training set (Step 2). We then extract bags of words from the documents we wish to group (Step 3). The bag of words is fed into an associative network, created and trained in Step 1 and 2, using association concentration (Step 4) to create an activation pattern from the bag of words. This pattern can then be compared to other patterns of documents in the collection to generate an estimate of the distances between various documents. Based on the distances between documents, a grouping in the form of a categorization or classification can then be made, depending on the purpose of the application (Step 5).

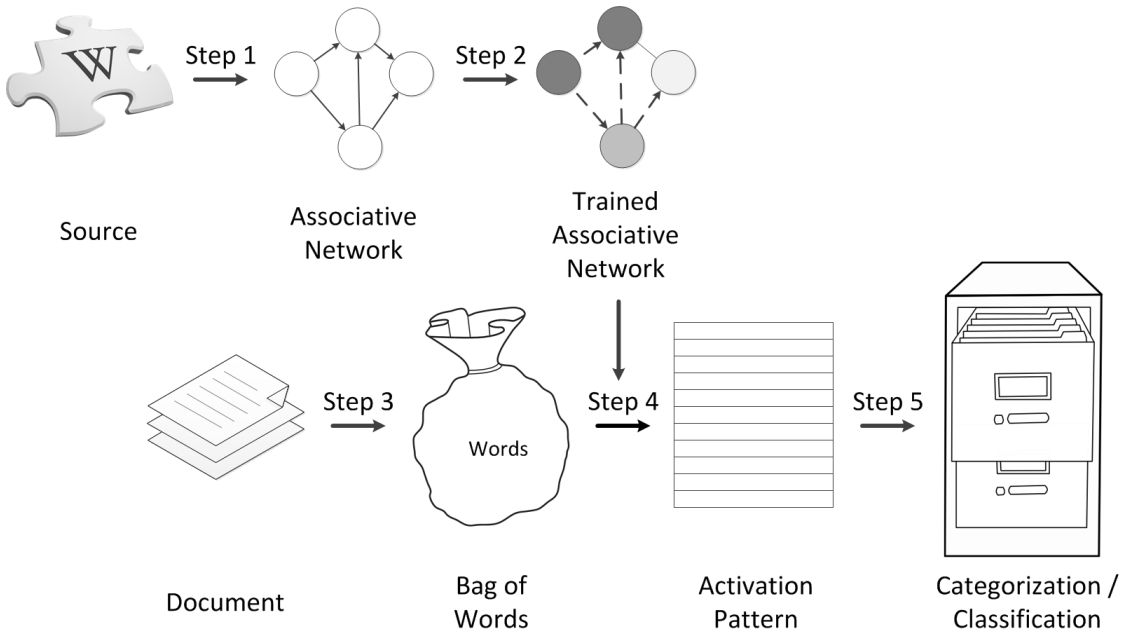

Figure 2.1: Process for categorizing or classifying documents

2.1 Step 1: Creating the Associative Network

In Step 1 a resource with semantic lexical relations is used to construct an associative network, into which the bag of words will be fed during Step 4 for the purpose of calculating associations.

Within the context of our approach towards document categorization, an associative network is a network of concepts which are connected with weights representing how similar in meaning those two concepts are. We can use an associative network to find concepts related to the text in a document even if the words describing those concepts are not in the document. This is done by means of association concentration, a technique described in the next section. In turn, those concepts can help us group documents correctly.

Structurally, an associative network can be thought of as a graph where each lemma in a language is a node and each relation between two lemmas is represented by a weighted edge. As there are many words in languages, associative networks generally have in the order of hundreds of thousands of nodes and millions of edges.

We do not construct associative networks by hand but rather rely on existing sources such as Princeton WordNet [Miller, 1995, Fellbaum, 1998] to provide us with a base structure, which can then be trained in Step 2.

In this chapter we only give a general overview of the use of associative networks in the context of automatic document grouping. In Chapter 3 we go into more detail on the model behind our associative networks and the algorithms we use. In Chapter 5 we describe how associative networks can be easily created. Multilingual associative networks connecting words in multiple languages are described in Chapter 10.

2.2 Step 2: Training the Associative Network

In Step 2, the associative network created in the previous step is trained, that is the weights between concepts are adjusted to more closely represent the distance between concepts in the network.

This training is done based on training documents which are related to the documents from the document grouping task. A primary technique we developed for training is based on back propagation, though we also developed Montessori training as an alternative technique. Both are described in detail in Chapter 6.

Once the associative network has been trained it is stored for later use in Step 4.

2.3 Step 3: Extracting the Bag of Words From a Document

The process of grouping documents starts properly with the documents themselves. We take the text from a document and make sure it meets the minimum qualification for use with our method, and we then extract a bag of words.

2.3.1 Document Pre-processing

To categorize or classify documents in a library, we start by scanning the text of each document, removing meta-data to acquire the raw text of the document, from which we wish to

extract a bag of words. Though meta-data can be useful in helping to make categorizations, we do not use it in our experiments. As an added advantage, this eliminates the quality of the meta-data as a factor in the success of our method.

We turn each document into raw text, not containing meta-data, annotations, lay-out information, etc. Since we do not use meta-data, for our purposes such data is garbage in the best case and pollution of the experimental data in the worst case. Our process is relatively robust, so we can make allowance for the occasional piece of meta-data or the like if the conversion into raw text from the data is not perfect, and such remnant meta-data is treated as part of the content of the document.

To give a real world example, if an HTML document is converted into raw text for the purposes of being processed with our method, we might extract all information not between angled brackets ('<' and '>'), which would eliminate a large amount of HTML styling. However, this might cause JavaScript code – which is not generally formatted between angled brackets – to be included to the raw text of the document. Associative networks can deal with some pollution, though of course the problem of extracting this type of noise can also be resolved by improving the quality of the raw text extraction, such as by using methods such as proposed by Gupta et al. [2005]. Like HTML, many other current data formats also contain a lot of meta-data, but allow for a relatively easy extraction of the actual text.

We presume that words in the documents are spelled correctly. That is, if we encounter the word *walk* we presume that this is the intended word, and that it is not intended to be *wall* with a small spelling error, and if we encounter the word *walp*, we presume that despite not knowing what it could possibly mean, the word is spelled as intended. If a document covers the topic of walking, the process will not break over a single misspelled word and in fact it will generally compensate for such errors without difficulty, as the error is compensated by the more frequent presence of the correctly spelled word as well as terms related to the general concept.

There are some other requirements to the input, such as the language used and the minimum number of documents necessary to make a grouping, but these are relevant to other steps of the system, so they are covered in the relevant sections below.

Bag of Words	
but	1
fast	1
faster	1
he	1
they	1
was	1
were	1

Bag of Lemmas	
but	1
fast	2
he	1
they	1
to be	2

Table 2.1: Bag of Words and Bag of Lemmas for the sentence *'He was fast but they were faster'*

2.3.2 Extracting the Bag of Words

After the document has been cleaned up to contain only the raw text, we extract a bag of words, a representation of the document as an unordered set of words (disregarding word order and grammar) which indicated how frequently each word occurs in the document. Using bags of words, we can determine which words are frequently used in the document and which are not.

As words can have different surface forms due to inflection, in most cases we use some type of translation from the surface form to the underlying lemma. Thus, though we use the term bag of words, in those cases a bag of lemmas would be a more accurate description.

As an example of how this works, consider the sentence '*He was fast but they were faster'*. In table 2.1 on the left side we depict the bag of words that would result from this sentence, while on the right we depict the bag of lemmas that can be extracted, which is a smaller list.

In the most basic case, the translation from words into lemmas is done by matching words by the surface forms of each lemma. If a word, for example *fast*, in the document matches the surface form of multiple lemmas (such as for going a period without eating and for travelling at high speed), all those lemmas are activated equally for that word. Since this can lead to inaccurate results (not all lemmas found in this way will be the one intended by the writer of the document), we examined the value of natural language processing techniques for improving the recognition of the correct lemma, a topic covered in Chapter 7. Of course the language of the original document also affects this step (see Chapter 10).

2.4 Step 4: Association Concentration

Once the associative network has been created we can then use the bag of words to activate the associative network and create an activation pattern. To do this we find links between words in the bag and nodes in the associative network and spread activation through the network based on the weights of the connections.

To start this process, we take the bag of words or lemmas generated in Step 3 and use it as input for the associative network created and trained in Step 1 and 2. Each word in the bag provides an activation value for the node in the associative network that represents the lemma, based on the word's frequency in the bag of words. As each lemma is linked to other lemmas, the node can then share this activation with its neighbouring lemmas based on the weight of the connection between the neighbours, activating them in turn. The neighbours then share their activation to their neighbours and so on, throughout the associative network.

Because of the association concentration algorithms (described in Chapter 3) the spread of activation will automatically concentrate in the nodes representing lemmas linked closely to the input words while leaving nodes that are not very closely linked with comparatively low activation, thus allowing us to identify lemmas that are closely related to the text despite not being present within it. This is why the method is called association concentration.

With this method, activation is spread through the associative network, first to neighbours and then along to their neighbours and so on. Activation flows strongly to words that are closely related to the document and only very weakly to words that are not closely related. Association concentration can thus be used to determine not just which words or lemmas are related to the document, but it can provide a numerical value for each word that describes how closely the word is connected to the original document.

In Chapter 3 we will explain the theory behind association concentration in more detail, while Chapter 6 describes how association concentration can be used to determine how closely two words are actually related to one another. Chapter 8 describes different algorithms behind association concentration, while Chapter 10 covers the influence that using more than one language has on the way association concentration works.

Word	*Activation Spread*
car	1.000
finish	1.000
racing	0.686
victory	0.456
vehicle	0.321
wheel	0.272
speed	0.160
line	0.105

Table 2.2: Simplified Activation Pattern

The spread created by association concentration is called an activation pattern. An activation pattern lists how much activation has spread to different words once association concentration has finished. Because of the way association concentration works, we expect the activation pattern to list high values for words that are closely related to the content of the document to be categorized while having low values for words that are only very distantly related to the content of the document.

A simplified activation pattern is shown in Table 2.2 as an illustration. As shown, an activation pattern somewhat resembles a bag of words: like the bag of words it consists of an unordered set of words which each have a value, but rather than being an enumeration of the number of times a word or lemma is present in a document, an activation pattern stores how much activation has spread to each word through association concentration. The higher the value for a word in the activation pattern, the more closely it is related to the document. This also means activation patterns reveal something about the topic covered by documents. In this, the method somewhat resembles Latent Dirichlet Allocation [Blei et al., 2003], which uses the relations between words and a generic topic to see if a document covers that topic based on how many such related words are in the document.

Activation patterns are discussed as part of association concentration in Chapter 3.

Besides producing an activation pattern, association concentration also results in association sub-graphs. Association sub-graphs are similar to activation patterns in that they contain the result of the spread made by association concentration, but they go one step

further and capture not just how much activation spread each concept related to a document has received, but also how that activation has spread through the associative network. The path of the activation spread is used to adjust the weights in associative networks to train them, and the details of this process are described in Chapter 6.

2.5 Step 5: Document Grouping

Though a single activation pattern can give us an indication of which words are characteristic of a document, activation patterns also allow us to compare two or more documents to determine how similar they are in terms of semantics, that is, how much distance there is between them. Moreover, we can do this not just for two individual documents but also for one document compared to a set of documents, which is important for the clustering task.

A numerical value for the distance between two documents can be used to determine the common coverage of a specific topic, providing us with a value that can tell us whether one document covers the topic more extensively than another or whether they give the same amount of attention to the topic. This too can be applied between two documents or between a set of documents, which can in turn be used to determine which topics a group of document shares.

In Chapter 3 we describe the exact details and mathematical formulas by which these values can be determined.

Once we have a comparison between documents we can use this to create a categorization or a classification. In both cases documents are grouped together based on their content, but using different methods as pictured in Figures 2.2 and 2.3. In the case of classification, we group documents into predetermined classes, placing each document in the class it is most closely related to. In the case of categorization, there are no predetermined groups, but rather we have a pile of documents and we group it into some sort of structure, grouping documents that cover similar topics together.

To classify documents we must have some knowledge of the predetermined classes, for example in the form of a set of sample documents for each class. Using the comparison described earlier, we can then calculate the average distance between the document we wish to classify and the documents in each of those classes. The class with the lowest average

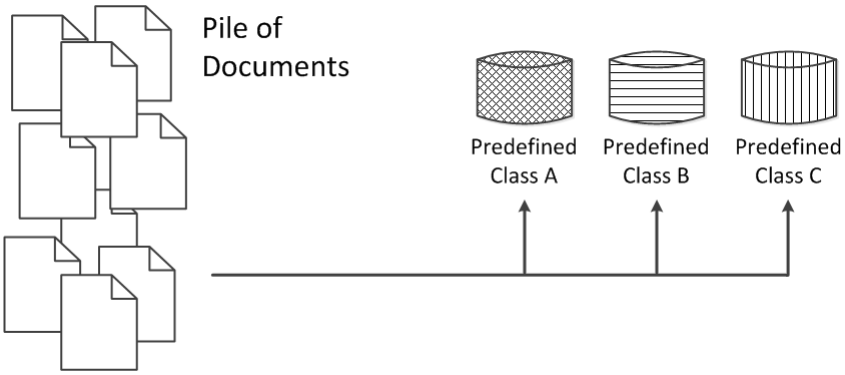

Figure 2.2: Classification

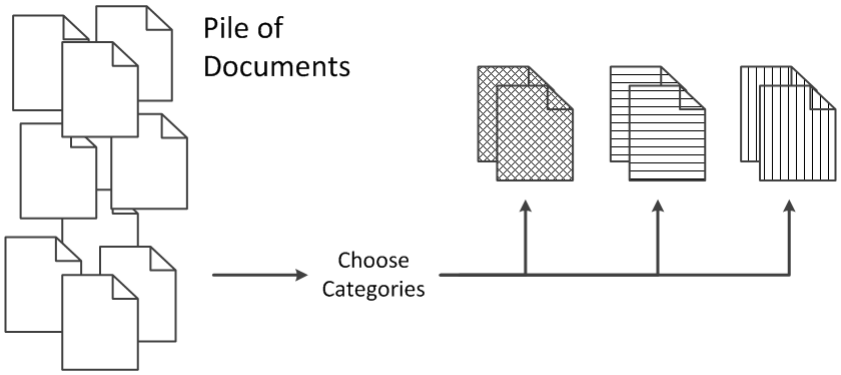

Figure 2.3: Categorization

distance between all the documents in that class and the document we wish to classify is likely to be the correct class. Alternatively, we can look at the n documents closest to the source document and see which class they belong to.

To categorize documents we instead analyse all documents we are trying to categorize, finding the exact distances between each possible pair of documents, thus for n documents we calculate $n(n-1)$ distances in total. A clustering algorithm can then be used to identify subsets of documents that are closely related to one another based on the distance between them. We use both multi-level graph partitioning (see Chapter 8) and power graph analysis (see Chapter 9) for this purpose. Each cluster of documents becomes a category, together forming a hierarchy of closely related documents.

However, the categories created using this method do not have a name yet and naming the categories is important to create a comprehensible structure. Based on the activation patterns of the documents, we are able to discover what topics a group of documents shares. To determine the name of the category we can simply find the word that is most covered and shared between all documents in the cluster. That word represents the content that each document in the category shares. We discuss the details and algorithm used for this in Chapter 3.

In our experiments in Part II, we follow the process described in this chapter as closely as possible. The first five experiments address Step 1 to 5 in our process, examining the creation and training of an associative network, the extraction of the bag of words, the association concentration that brings them together and finally the method of grouping documents. The sixth experiment extends our process with the ability to handle multilingual document collections. In our final experiment, we examine the entire process as a whole. Through these experiments, we show how associative networks can bring order to a document collection.

Chapter 3

The Association Concentration Method

> *"Whether to concentrate or to divide your troops must be decided by circumstances."* - Sun Tzu [500 BCE]

In Chapter 2 we provided a general overview of our approach to grouping documents using associative networks and association concentration. In this chapter we describe how associative networks are modelled and detail the algorithms used for association concentration, the method by which associative networks infer information about the semantic structure in a document. Finally, we describe some practical issues related to associative networks such as how to calculate the distance between documents and how to deal with out-of-vocabulary words and dynamic libraries.

3.1 Associative Networks and Lexical Semantics

As explained in Chapter 1, two individuals may use different words to describe the same object while still being able to understand what the other is speaking about. For example, where a racing aficionado might call a certain object a *Formula 3 Porsche*, a layman may simply describe the vehicle as a *race-car*. Despite using different words, both will be able to figure out what the other is speaking about from the other terms used in the conversation which they both understand, even if a word (such as *Formula 3 Porsche* in our example) is unknown to them.

Unlike computers, people are very good at recognizing patterns from minimal data. For example, they can determine that a *Formula 3 Porsche* is a race-car from a very short sentence such as *'The driver of the Formula 3 Porsche was faster and won'*. The sentence does not contain enough information to draw such a conclusion with certainty, but in most contexts, people can figure this out regardless.

Drawing this kind of conclusion is only possible if people are aware not just of the word itself, but of the wider context surrounding the concept that word represents. Parallel to the way in which people who have seen it previously are still able to recognise the Ship of Theseus described in Chapter 1 based on its generic appearance (even if they do not recognise some of the planks or rigging that was replaced), a layman would be familiar with what a race-car is conceptually, even if a specific race-car does not match any they have seen earlier. They would also know that it is associated with winning (a race), with going fast, and that the race-car would require a driver, for example.

Going back to the earlier example of *'The driver of the Formula 3 Porsche was faster and won'*, certain knowledge about race-cars can be linked to the sentence. For example, the *Formula 3 Porsche* has a driver, it goes fast – faster than its competition anyway – and as a result it wins a race. Thus the layman might rightfully conclude that *Formula 3 Porsche* is likely to be linked to race-cars, based on their knowledge of the latter.

Rather than merely knowing what words means, people know both the word and the properties of the concept that word represents. They are generally aware of its semantic field [Trier et al., 1973, Faber and Usón, 1999], its lexical relations [Gruber, 1965], and more generally how words in the sentence relate to the other concepts expressed within the sentence. As a result they are able to match the properties they know from the context surrounding the unknown term (i.e. the words used in the same utterance) to lead them to understand the meaning of that unknown term. This extra step, connecting concepts through their properties and concepts associated with them, lies at the core of associative networks and association concentration (see below).

3.1.1 Graph Model

As illustrated by what is stated above, human ideas and concepts do not exist in a vacuum. Rather they are defined by their connection to other ideas and concepts, and each of those is in turn connected to even more ideas and concepts, together forming a large network. This network of connections is called an associative network.

Associative networks are networks of concepts that are related to one another and these relationships can be stronger or weaker depending on how closely the concepts are related.

Based on this idea, an associative network can be modelled as an undirected weighted graph. In this graph each node represents one of the concepts, and each edge represents the relationship between two concepts. Edges are undirected, as a relationship between two concepts is always two-way. If the concept of FLY is related to the concept of MOSQUITO, then the concept of MOSQUITO is likewise related to the concept of FLY. Additionally edges are weighted, with higher values representing closer relationships between two concepts. For example, the concepts of FLY and MOSQUITO are more closely related than FLY and INSECT, the latter covering a far wider range of creatures with far less similarity between them, thus the edge between the nodes for FLY and MOSQUITO has a higher weight than the edge between the nodes for FLY and INSECT.

Concepts in the associative network must be rooted in the real world in some way. As we do not have access to Plato's Universals, we instead have to follow a more Aristotelian approach and use observations. Words are candidates for observation, and we use them as such in our research. Other types of empirical data are also possible, including visual patterns or sound bites, but these lie outside of the scope of our work and thus we have not explored them.

In terms of scale, lexical associative networks based on natural language vocabularies have between several hundred thousand and a bit over a million nodes, with usually a bit over a dozen, though sometimes up to a hundred edges per node.

3.1.2 Example Associative Network

Figure 3.1 depicts a simplified version of an associative network. Note that since associative networks generally cover most of the words in a language, they are much bigger than the

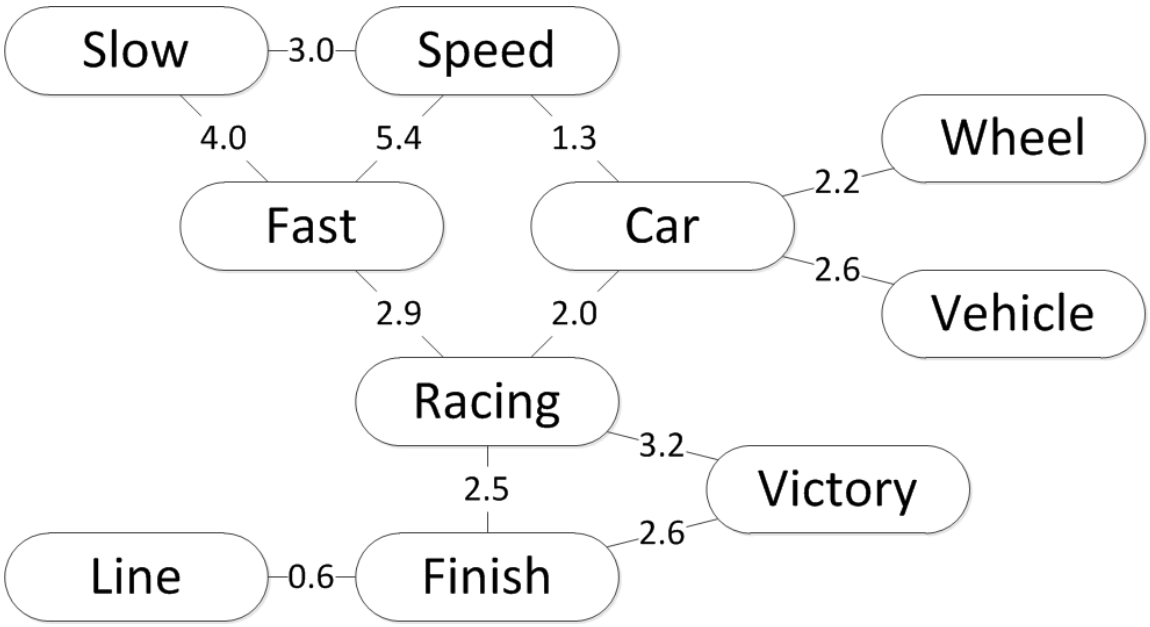

Figure 3.1: Simplified Associative Network

one depicted. As mentioned, associative networks consist of hundreds of thousands of nodes with words and millions of edges connecting them together, but since this would be impractical in print, we present a strongly simplified example here. The number of connections between nodes, by comparison, is relatively small (up to a thousand edges per node), meaning the network forms a sparse graph.

Edges between nodes represent lexical semantic relations [Gruber, 1965, Jurafsky and Martin, 2009] between the words or concepts represented by the nodes and may include both *is-a* relationships such as between CAR and VEHICLE and *has-a* relationships such as between CAR and WHEEL in addition to more associative relationships such as between CAR and RACING. Even antonym relationships may be present such as between FAST and SLOW. It should be noted that though these different types of relationships can be represented by edges in the network, the associative network does not differentiate between them – all types of relationships are treated the same and in fact the information on what type of relationship exists between two nodes is not stored.

In the figure, we can also see that closely related concepts such as SPEED and FAST are connected by edges with a high weight, while concepts that are less closely related such as SPEED and CAR are connected by edges with a lower weight.

3.1.3 Creation and Training

Making an associative network of all words in a given language would be an enormous undertaking. Fortunately, others have already worked on similar problems, creating networks of related words. We can use these networks to our advantage and save us from having to build an associative network from scratch.

Two excellent sources that we used to create associative networks are Princeton WordNet [Miller, 1995, Fellbaum, 1998] and Wikipedia. Both were used to create networks where each node represents a concept. The links between related concepts they provided were used to create edges in the associative network. The process of creating such a network is described in Chapter 5.

Once it has been created, an associative network can be trained. Training adjusts the weights in an associative network. This should help improve the accuracy of the weight of the edges between various concepts, as most methods of creating an associative network do not immediately provide perfect values for these weights (some indeed do not provide any values). The process of training an associative network is described in Chapter 6.

3.2 Implementing Association Concentration

In our earlier example, we stated that a very short sentence such as *'The driver of the Formula 3 Porsche was faster and won'* provides enough context for most people to figure out that *Formula 3 Porsche* refers to a race-car, even if they had never heard the name before. We explained that people are able to use the properties typically attributed to a race-car – being fast, being driven by someone and participating in a competition – to figure this out.

However, while an associative network would contain the links from these concepts to the concept of a race-car, we have not explained how it can be used to draw a formal

conclusion about what a Formula 3 Porsche is. After all, the associative network does not just contain links between terms like race-car and fast, but it contains countless other links as well. For example, many terms are related to the concepts of driving, going fast and participating in a competition. Driving might relate to travel, being faster might be related to reaction time and participating in a competition can be linked to the lottery. How can we identify the race-car amongst all these many possibilities?

3.2.1 Spreading Activation

A key step in solving this conundrum is to find concepts that connect as closely as possible to the information we have, while ignoring all connections that are superfluous given the context.

In its basis, association concentration makes use of the fact that while each word in a language is related to many other words, there will be only a limited set of words in a document that is closely related to each (or rather most) of the words in the document. For example, suppose we have a document containing the word *car* and the word *finish*. Using the associative network in Figure 3.1, the word *car* can be related to the words *speed*, *racing*, *vehicle* and *wheel*, while the word *finish* can be related to the words *line*, *racing* and *victory*. Since the word *racing* is a shared neighbour between the two, it makes sense that this word is relevant for a document containing both the word *car* and *finish*.

Association concentration takes this one step further. Merely looking at shared neighbours would not work for most documents as they would contain too many different words with no way to distinguish between accidental and actual matches. Instead, association concentration spreads activation out from the words in the document like an oil stain, traversing along the network in all directions, not just to neighbours of each word, but to the neighbours of those neighbours and so on. We call this oil stain the activation pattern. At first this may seem even worse than merely looking at shared neighbours, but the activation pattern is affected by the weights in the associative network, and activation spreads more easily to words that are connected through a higher-valued edge. If a word is related to

more than one word in the document, that word will receive activation spread from multiple sources, resulting in the concentration of activation in that word and thus a high value in the activation pattern.

Thus, using association concentration, we may find that *race-car* is very closely related to *Formula 3 Porsche*, and *racehorse* a bit less so. Like *race-car*, *racehorse* is related to going faster and winning. With a bit of a stretch, *racehorse* can also be related to the word driving, as one can drive and race a horse-drawn carriage, but it is more closely linked with riding than with driving, thus the term *race-car* is a more likely candidate to explain what *Formula 3 Porsche* is.

Clearly the quality of the associative network has an impact here. In order for association concentration to work in all cases, a complete associative network (or at least as complete as possible) is necessary, with the most accurate possible weights for its edges. Without this, we cannot link the various properties which we recognised surrounding the *Formula 3 Porsche* to link it to the term *race-car*, nor would we be able to distinguish that it, rather than *racehorse* is most closely related.

3.2.2 Algorithms

Activation can spread through the associative network in different ways to create an activation pattern. We describe two methods by which such activation may be spread, and others may also be possible. These methods, which we named 'Spreading Activation' and 'Flow Activation' respectively, offer similar, but not identical activation patterns for a given input, the difference being the termination condition.

In either case, the associative network is activated by a certain input, typically a document represented as a set of one or more words with their frequency in the document. As a simplified example, in Figure 3.2, the associative network is activated with an input value of 5 for FAST, as the word *fast* was present five times in the document, and 4 for FINISH as the word *finish* was present four times in the document. The input values are indicated in the figure by the incoming arrows and the marking of the nodes. The activation is spread from this input, activating neighbouring nodes in the network, which may in turn activate even more nodes.

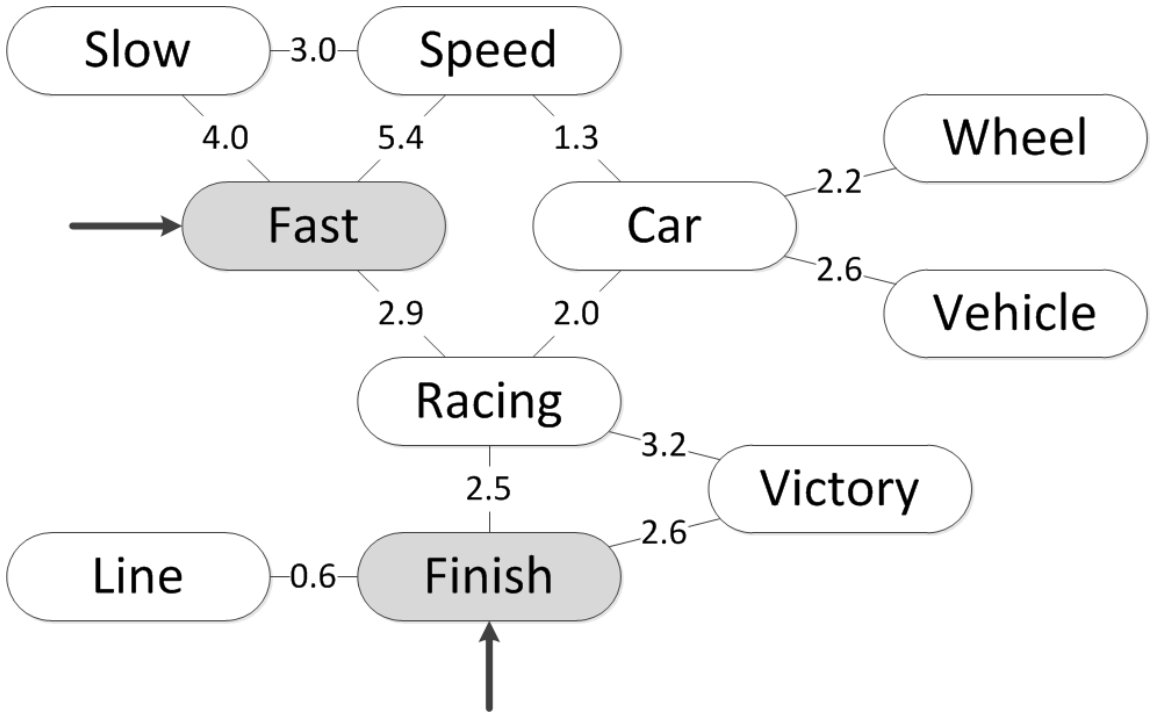

Figure 3.2: Primary Activation

Once the primary input has been provided, neighbouring nodes are then activated with a value proportional to the weight of the edge connecting to them. For example in the same figure, activation spreads from the node for FINISH to LINE, RACING and VICTORY. The input value of 4 will then be divided over those three nodes proportional to the weight of the edges between them and FINISH. Thus, the node LINE would receive $0.6/(0.6+2.5+2.6)$ $\approx 10.5\%$ of the input value. For the input value of 4, this would be roughly 0.42.

This method ensures that the closer two concepts are related, based on the weight of the edge between them, the more one activates the other. Thus activation will easily spread to closely related concepts while distant concepts activate one another only minimally.

Figure 3.3 shows an example of association concentration: both the concept FAST and the concept FINISH link directly to RACING. This means that the concept RACING receives activation from multiple sources and will thus have a higher value than it would have gotten from either source independently.

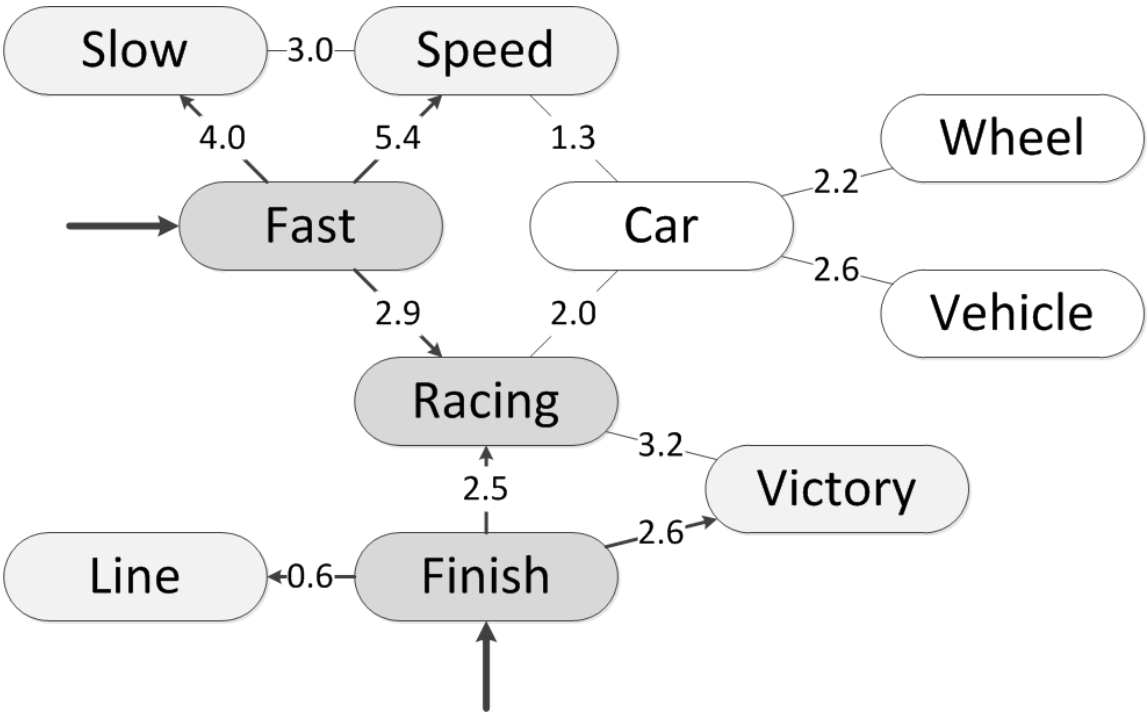

Figure 3.3: Secondary Activation

Up until this point, both spreading activation and flow activation operate identically. In both models, each node can be activated only once. This means nodes cannot activate their neighbour and then be activated by their neighbour in turn, for example. If allowed to spread unchecked, each activation pattern would encompass the entire associative network. As mentioned, the difference between the methods lies in the termination conditions.

We created the Spreading Activation algorithm (see Algorithm 3.1), based on the idea that nodes with a very low level of activation are not very relevant for the activation pattern. Using this, if the activation falls below a threshold value, that node is not activated at all and thus will not spread activation. Eventually, all nodes not yet activated will have a level of activation that falls below the threshold value and the activation stops spreading. This means all terms that have not been activated are too distant in the associative network from the input nodes to be considered relevant.

The Flow Activation algorithm (see Algorithm 3.2) is based on flow networks [Ford and Fulkerson, 1962], which are directed graphs where each edge has a limited capacity

Algorithm 3.1 Spreading Activation

```
# declare basic functions
function WEIGHT(a, b)
    weight of edge between node a and b
end function

function TWE(a)
    sum of weights of all edges connected to a
end function

s = list of all nodes

a_i = the activation value of the input for node i

THRESHOLD = the global threshold value

while s is not empty do
    remove node r from s with the highest a_r and a_r > THRESHOLD
    if no r is found then stop
    end if
    for all neighbour v of r do
        # increase activation
        a_v ← a_v + a_r * (WEIGHT(r, v)/TWE(r))
    end for
end while
```

Algorithm 3.2 Flow Activation

```
# declare basic functions
function WEIGHT(a, b)
    weight of edge between node a and b
end function

function TWE(a)
    sum of weights of all edges connected to a
end function

s = priority queue of all nodes sorted by activation value from large to small

a_i = the activation value of the input for node i

SINK = the global sink value

while s is not empty do
    remove node r from the front of s
    for all neighbour v of r do
        # only add activation above the sink value
        a_v ← a_v + MAX(a_r * (WEIGHT(r, v)/TWE(r)) − SINK, 0)
    end for
end while
```

and each node receives input from incoming edges called ‘flow’. This ‘flow’ originates from nodes called ‘sources’ and is absorbed by nodes called ‘sinks’. The amount of ‘flow’ going into a node must equal the amount of ‘flow’ going out of the node.

In the context of associative networks, the Flow Activation algorithm acts in a similar manner, with one important difference: in regular flow networks there is a predefined sink, but in an associative flow network every node is a partial sink. This means every node absorbs a small amount of ‘flow’, thus reducing the total amount of activation being spread. During an operation, once a node has absorbed this amount, it is saturated, and from then on acts as a regular node in a flow network, passing all further incoming ‘flow’ on to its connected nodes.

In both cases, once the spread of activation stops, the total activation for each node is collected. The eventual activation pattern thus consists of a set of nodes and their activation values. As described in Chapter 2, this activation pattern can then be used to compare the associations made for different documents.

3.2.3 Performance

One of the advantages of using association concentration is the performance in terms of computational complexity and therefore speed. Regardless of whether flow activation or spreading activation is used, any vertex in the network is activated at most once during the process of association concentration. Even if the whole network is activated, the number of activations is thus limited to O(V) where V is the number of vertices. The number of activations depends largely on the configuration of the flow activation or spreading activation algorithm, but the number of activations is usually far less than V; the associative network is sparse and the nodes activated in a given input are likely to be clustered – in fact, this clustering is why the associative network technique works.

The fast performance means the system can respond to changes in the libraries in real time, allowing it to keep a correct categorization at all times.

3.3 Practical Issues

In this section we describe some practical issues related to the method of association concentration such as how to determine the relationships between documents based on their activation patterns, how to deal with out-of-vocabulary words and how to handle dynamic data. While not strictly part of the association concentration method, these issues are closely related to the general concept and are encountered in practice when using associative networks for document grouping.

3.3.1 Distance between Documents

By comparing the activation patterns for two documents, we are able to tell how closely they are related compared to other documents. Thus we can tell if a document is more closely related to one document than to another. Moreover we are able to tell by which concepts two documents relate to one another. For example, we are able to tell that two documents are closely related because they both describe SAILING. We do this by comparing the activation patterns of the two documents with regards to the concept SAILING: if

both documents have high values for this concept within the activation pattern, this indicates that both documents are related to this topic. The relationship between two documents in a more general sense can then be calculated by looking at all possible topics.

To calculate how similar two documents A and B are in relation to a concept c, we calculate the difference in activation between the activation patterns of the two documents in relation to that concept c, expressed in the following formula:

$$D(AB \mid c) = \left| V_{A(c)} - V_{B(c)} \right| \tag{3.1}$$

where $V_{A(c)}$ is the activation spread of the concept c in document A and $V_{B(c)}$ is the activation spread of the concept c in document B.

The higher the value of $D(AB|c)$, the bigger the distance between the two documents in terms of their relationship to concept c. If the distance approaches zero, this means either both documents cover the topic extensively or neither does. If it is relevant to know which of these is true, it is enough to simply look at the value of $V_{A(c)}$ or $V_{B(c)}$. The height of the value indicates how relevant the topic is to both of the documents.

Since the overall distance between two documents is the sum of the distances between two documents over all topics, we can then determine the distance between two documents expressed as $D(AB)$. Specifically, it is calculated as:

$$D(AB) = \sum_{i=1}^{n} \left| V_{A(i)} - V_{B(i)} \right| \tag{3.2}$$

where i is the index over all n concepts in the either of the two documents, $V_{A(i)}$ is the activation spread of concept i in document A and $V_{B(i)}$ is the activation spread of concept i in document B. In other words, the distance between document A and B is calculated as the sum of the absolute differences between the activation value V of each term in documents A and B. Knowing the distance between individual documents allows us to compare which of a set of documents is more closely related to a specific document. Other metrics, such as the Euclidian (L2) distance, might also be used.

3.3.2 Out-of-Vocabulary Words

Matching words to the concepts they represent in an associative network is something we examine in great detail (see Chapter 7), but it is not always possible to make a connection between a word and the concept. One problem may be that the word lies outside of the 'vocabulary' of the associative network, that is, the concept represented by the word does not exist in the associative network at all. It may be because the word is a specific name not known by the system or it could be that the associative network is incomplete and simply lacks knowledge of a certain concept.

Whatever the cause, out-of-vocabulary words may still be relevant in finding relationships between texts. One way to still represent the relevant information represented by the word in a documents activation pattern is to add the word as a virtual entity to the associative network, that is, expand the associative network by adding a node (unconnected to any other node) for that particular word. Since it is not connected to the rest of the associative network, the out-of-vocabulary word cannot spread activation across the network, but still provides an activation value for itself in the activation pattern generated, equal to the input from the bag of words. This allows matches to be found with other texts that use the exact same word. This option was used in our experiments (see Part II).

Another option is to use a self-learning associative network. In such a network, the word would have a node added to the associative network as described earlier, but it would be connected to the other terms in the document that mentions it. Training would then be used to set the weights of the connections between this new node and the rest of the associative network. We have not done experiments with this type of network, but its use is described in a little more detail as potential future work in Chapter 14. This kind of learning is most appropriate for a continuous use scenario, where the associative network operates in a dynamic environment.

3.3.3 Handling Dynamic Data

Associative networks are not always applied to static libraries and many modern document libraries, such as wikis, are designed to be easily be modified. Users with access rights can add, remove or edit documents. Any addition, removal or edit can influence the proper way

to categorise the documents, which can make maintaining groupings for the documents in the library tedious.

The simplest solution to the problem of dynamic grouping is to rerun the entire categorization process whenever a document is added, removed or edited. Especially for large libraries with frequent changes, such a solution is undesirable as it will require a large amount of resources to keep up with the edit frequency.

A better solution lies in dealing with the individual edits separately. There are several situations that can occur as a result of individual edits. When a document is removed from the library, the category that it was sorted into may become too small to remain useful. In such a case, the remaining content can be spread out over existing categories. If the categories are hierarchical, the articles can be incorporated directly into the parent category or distributed amongst its sibling categories. Likewise, by calculating the average distance between a new or edited article and the articles in each category, the category that most closely matches the new article (or that now matches the edited article most closely) can be found. Finally, when a category becomes too large, it can be split up into multiple subcategories by clustering the documents within into different groups. Associative networks work particularly well for such clustering as we examine in Chapter 8.

If a document is removed, no adaptation to the knowledge stored in the associative network needs to be made – this is because the document itself is not stored in the associative network, merely the relationships between concepts derived from that article. The assumption here is that even if the document itself is removed, the relationships between concepts extracted from the document are still valid.

Likewise, when a document is edited, only the relations between the document and the associative network change – the internal structure of the associative network remains the same and thus knowledge about relationships between concepts is unaffected.

An additional feature that makes our method suited for handling dynamic data is the fact that the links between two articles are independent of the rest of the library. This allows relations to be analysed in parallel and even allows one part of the library to be reordered without affecting the other parts.

Chapter 4

Background and Related Work

> *"A wise man ought always to follow the paths beaten by great men, and to imitate those who have been supreme."* - Machiavelli [1532]

In this chapter we discuss work that resembles ours, that tries to solve similar problems or that has inspired our own models. We do not cover work that is related to the experiments described in Part II here. Instead, such related work is discussed in the specific chapters devoted to those experiments.

To get a closer understanding of associative networks, we first take a look at their history (in Section 4.1), and then look at cognitive models that resemble associative networks. We examine work based on those cognitive models and compare it with our work including association concentration. In Section 4.2 look at case-based reasoning, which inspired our original ideas on this topic, and case retrieval nets, a technique from case-based reasoning that somewhat resembles association concentration. In Section 4.3 we look at connectionist models, which is a wider group of approaches that associative networks and association concentration are part of. Finally in Section 4.4, we look at other text classification and text categorization techniques (some of which are based on the previously discussed models), comparing them to our own work.

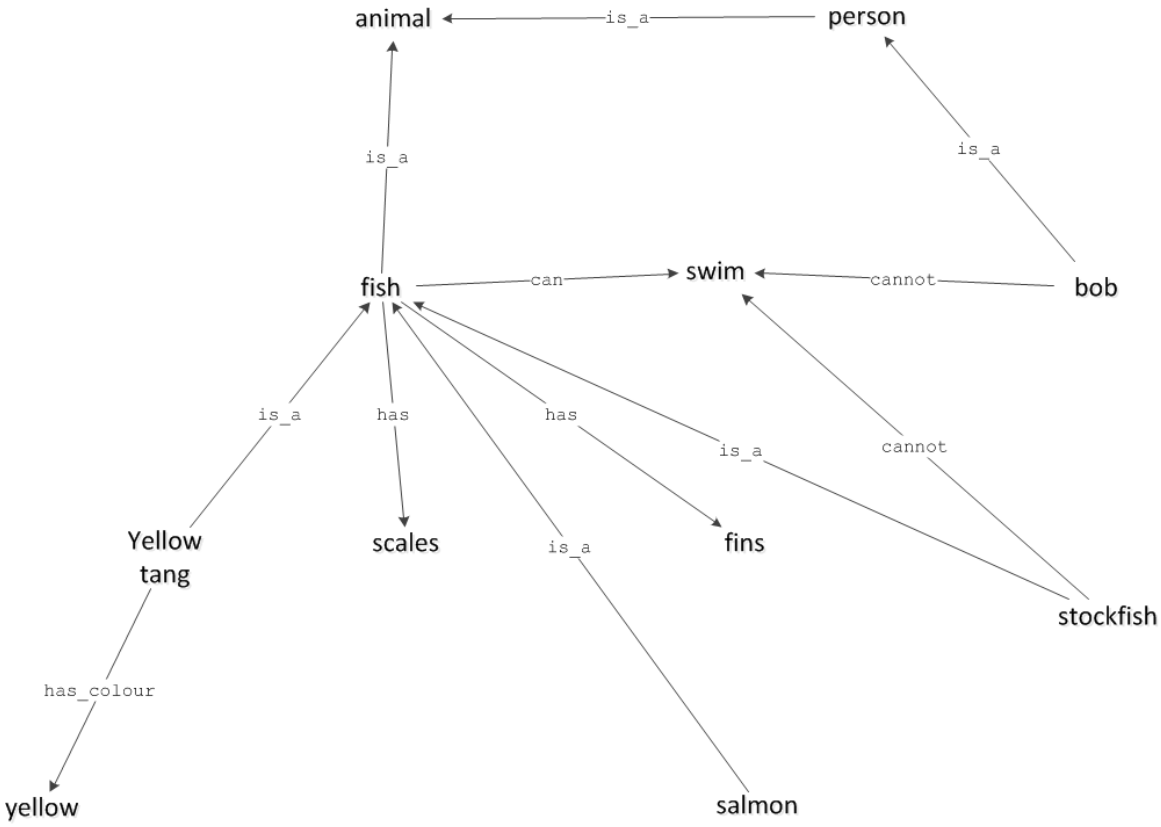

Figure 4.1: Example Semantic Network

4.1 History of Associative Networks

Associative networks evolved from Semantic Networks, a technique from the field of knowledge representation. Semantic Networks were originally introduced by Quillian [1968] and they form an early foundation of associative networks. Quillian's semantic networks express knowledge in terms of concepts and their properties, as well as the hierarchy of sub- and superclass relations such as '*is-a'* and '*instance-of'* relationships. In this hierarchy the lowest-level nodes denote individuals, while higher-level nodes denote classes that become more abstract higher up in the hierarchy. Properties are nodes in the network as well and connect to nodes which have those properties. Typically, a property is connected to the highest-level node where it would apply and it is presumed to apply to all descendants of that node as well. Figure 4.1 depicts an example of a Semantic Network.

In the field of knowledge representation, more generic versions of the Semantic Network evolved without the limits of specific relation classes, and these can be more accurately referred to as associative networks [Jackoway, 1984]. Findler [1979] also describes associative networks as a generalization of semantic networks, but keeps in their representation some of the meaning of the relationship between concepts. In this it contrasts with our own definition of an associative network, which only keeps the connection between concepts, but does not store what type of relationship it represents, merely that it is a connection. In this more generic type of network information items are also represented by nodes, and links express undefined and unlabelled associative relations between those nodes.

Crestani [1997] describes the evolution of networks that use weights to index terms and which provide information about the level of strength of the associations. One example of the developments observed by Crestani is the work of Kimoto and Iwadera [1990], who use the term dynamic thesaurus for what is essentially an associative network.

Schvaneveldt [1990] uses the term pathfinder associative networks, describing them as networks which are created based on proximity between terms, with certain links being eliminated if shorter (indirect) paths are available. This results in a similar style network as our own associative networks, though our construction and training method, especially when considering the entire domain of a language, is different.

Besides being used for knowledge representation, the term associative network has also been used in psychology to describe the associations, especially emotionally, which an individual may have with certain events or concepts, such as described by Bower [1981] and Schubert [1996]. In these, the associative network itself features additional connections between various concepts and emotional centres in the brain. Thus for example, a certain pattern may activate not just related concepts, but also the feeling of joy. Such associative networks are of course a literal representation of the language of the nervous system.

4.2 Case-based Reasoning

In this section we discuss case-based reasoning and case retrieval nets, a technique from the case-based reasoning field. We compare these approaches to associative networks and association concentration and examine the similarities and differences.

4.2.1 Cognitive Model

Case-based reasoning, first described in the works of Schank in the late seventies and early eighties [Schank and Abelson, 1977, Schank, 1982] is based on a cognitive model that describes the working of human memory and specifically how it deals with everyday situations that occur often in different configurations.

Like associative networks, this model presumes that human ideas do not exist in a vacuum, but rather that they are connected to many other ideas, providing context to one another. In case-based reasoning, a memory or case is formed from a set of properties that describe it. Together, these properties define the case and they can be used to compare different cases to one another. The presumption is that even though cases may differ in details, the general trend for similar cases would be similar. Memory-Based Learning Daelemans et al. [2003], Daelemans and van den Bosch [2005], a similar technique, is based on the same model of the working of human memory as case-based reasoning. As in case-based reasoning, cases encountered are stored as is, and future cases are compared by their features, and different features are weighed differently based on domain knowledge. Our own method of association concentration similarly to these two methods presumes that the properties of the data (such as words in a text) describe an underlying concept and that different samples - though they may differ in the details - have the same underlying trends. Case-based reasoning, memory-based learning and association concentration all presume that sets of related data – be they cases or documents – are descriptive of an underlying concept.

As an example of how the cognitive model describes human memory, Schank mentions a visit to a restaurant, where customers find a seat, place their order with the waiter and then wait for food to arrive. This scenario plays out the same, regardless of what restaurant is visited, who the waiter is and what food is ordered, and the fact that all parties involved

know this allows the visit to go smoothly: the guest does not have to worry whether or not the waiter will actually be motivated to bring their food, for example. A specific example case, which accurately describes the restaurant visit, is stored and used as the guide to match future cases of the restaurant visit, which in turn can be stored as new additional cases, allowing new experiences from different visits to each be represented.

Associative networks rely on the same underlying principle to understand how certain ideas are (or are not) connected. We discuss the details of how associative networks learn elsewhere in this document, but in the case of the restaurant visits, certain features (such as the finding of seats and the placing of orders) will be the same for all restaurant visits while others (such as who the waiter is or what food is ordered) will be different. Associative networks can learn to identify from this repetition (or lack thereof) which features are more and which are less relevant to the restaurant visit.

Despite relying on the same principles, case-based reasoning and associative networks use very different ways of storing knowledge of previous events. When applied to solve problems such as categorization, case-based reasoning define a formal case (a sample problem) and a solution for that case, which it either stores with full details or which it throws out if it already has a very similar case. When facing a new problem not previously encountered, it then compares the known cases to this new problem, finding the case that most closely matches the new problem and applying the solution of that case, making adjustments as necessary. Once it has found the adjusted solution to be satisfactory, it stores the current problem and the resulting solution as a new case, which is again stored with full details. To keep the case-base from growing out of proportions, redundant cases covering the same problem can be deleted from the case base.

In contrast to both case-based reasoning and memory-based learning, associative networks do not store individual cases and in fact have no recollection of individual events as such. Instead, cases are merged together into a single associative network which contains an amalgamation of all cases ever encountered. Thus, by merging the data some details are lost, potentially trading in accuracy, but in exchange it allows the associative network to filter out irrelevant details in each case more easily than a case-based reasoning system could without human intervention.

4.2.2 Case Retrieval Nets

To increase accuracy when matching problem-solution pairs as described in the previous section, it is important that cases in case-based reasoning are stereotypical examples of a problem. As in many situations there is no specific stereotypical example, some research has been done into the merging of data to help identify the key features of cases with a specific scenario such as the restaurant visit.

In the field of case-based reasoning, research on the merging of data has focussed on the retrieval of cases, not the complete merging of overlapping data as is the case with associative networks. As a result these methods have several limitations that do not apply to associative networks, especially concerning the filtering of undesirable data. For example, Chakraborti et al. [2006] describe case retrieval nets, which are networks of features and cases in the case base used to find better matches of a query with the cases in the case-base. Chakraborti et al. offer their own, faster variation of these case retrieval nets which they descriptively named fast case retrieval networks. In both regular and fast case retrieval nets, each case is linked to certain binary features such as the presence or absence of a certain piece of data – which Chakraborti calls information entities – and the features themselves are related by a similarity function. Information entities can be any of the properties normally used to describe a case so long as they are binary features.

Case retrieval nets find the relations between a query and a known case by taking the features of a query, activating these in the case retrieval net and spreading to related features to create an activation pattern which is then used to calculate the most closely related case, similar to how associative networks operate. Thus two concepts which may be related in the case retrieval net are INDEXING and CLUSTERING, which describe similar actions, for example. This type of matching allows terms with similar meaning to be matched to the query, very similar to how association concentration match related terms. However, despite this seeming similarity, there are some key differences between case retrieval nets and associative networks: case retrieval nets spread only by one step, that is, an information entity only spreads activation to the directly neighbouring node. Activation therefore does not propagate further to neighbours of neighbours – thus INDEXING would link to CLUSTERING which is a neighbour of INDEXING, but it would not link to GROUPING which is a

neighbour of CLUSTERING but not of INDEXING. This restriction – only using direct links but not indirect ones – makes sense when a limited set of features is present. Few features means that spreading further will likely provide less accurate information as information entities are likely too far apart to meaningfully influence each other, which is what associative networks rely on. With many features as is the case with associative networks, information entities are more likely to influence one another. Association concentration uses this by concentrating activation in the most relevant nodes (the associative network parallel to information entities) while leaving the less relevant ones mostly untouched.

Chakraborti also describes a use of statistical co-occurrence with manual adjustment to determine the relationship between Information Entities. Manual adjustment is not generally an issue when applied to case-based reasoning domains as these generally operate with relatively few cases, but the manual adjustment would be a significant obstacle for problems with a high number of properties such as text classification which can have hundreds of thousands or even millions of features.

Jayanthi et al. [2010] try to compensate for the lack of reach to neighbours of neighbours in case retrieval nets by using various methods of introspective learning. For example, they try to identify cases that negatively affect classification performance and remove these from the case-base or make modifications to make the cases more accurate. All these methods modify the case base, which makes sense, as this is the key knowledge store for Case-based Reasoning.

Case-based Reasoning might be further improved using associative networks and association concentration instead of case-retrieval nets. Notably, the less constrained similarity function of associative networks makes it easier for them to learn. In a case-retrieval net all relationships between concepts must be direct, thus a case-retrieval network is much denser than an associative network of similar scale. In turn, it is harder for an algorithm to accurately identify which connections should be strengthened or weakened in a case-retrieval net than it is with associative networks (see also Chapter 6).

4.3 Connectionist Models

Connectionism [Bechtel, 1988] is the label used for a set of approaches to various artificial intelligence problems. It uses a network of simple interconnected units that produce a solution through emergent behaviour. As associative networks are networks of interconnected nodes representing concepts, they too fall within the area of connectionism. In this section we look at other connectionist models and techniques and compare them to our work.

4.3.1 Neural Networks

Perhaps the most well-known connectionist model is the neural network [Dumais et al., 1998], which is a mathematical model used to solve certain artificial intelligence problems inspired by biological networks of neurons. Neural networks are generally structured into layers, with the input nodes receiving data and spreading an action potential to the next layer of nodes that is in turn modified by the weight of the connection between those nodes until a certain output value is produced at the output nodes. Back-propagation [Bryson and Ho, 1969] is a technique used to train neural networks by adjusting the weights of the edges in the network so it produces the desired output for a given input.

Both associative and neural networks thus consist of a network of interconnected nodes, but associative networks are generally bigger and do not have a layered structure like neural networks, which have an input layer, an output layer and may have one or more 'hidden' layers. In contrast, associative networks do not have layers; thus all nodes in the associative network are input, output and 'hidden' layer nodes. This lack of layers also differentiates associative networks from deep learning [Bengio, 2009], a technique that has been especially successful with neural networks. In deep learning, observations run through several layers of processing, and each layer transforms the data to extract features which are then further analysed by the next layer.

Associative networks do not store cases as they were encountered, but instead merge all information they have encountered into the weights of their edges, thus filtering out irrelevant details. This method of storing information is very similar for associative and neural networks and both associative and neural networks can learn using back-propagation. To do this, both associative and neural networks calculate the path back to the input from the

output and adjust or train the weights along this path to produce a more successful network. We compare the performance of the two in relation to the document grouping task in Chapter 8.

Hopfield networks [Hopfield, 1982] are a specific type of neural network which serves as content-addressable memory, a special type of computer memory used in certain search applications, and is sometimes used as a model for understanding memory. Hopfield networks are trained to match specific patterns such as for the categories in which a document can be sorted, but they cannot infer information about the document itself. This is very different from associative networks, which generate an activation pattern for each document or category based on sample data and generic knowledge about the language, using that knowledge to infer additional information.

4.3.2 Spreading Activation

Spreading activation is a process used in connectionism that closely resembles association concentration. Spreading activation has proven itself to be successful in the domain of information retrieval [Crestani, 1997], which resembles, but is not the exact problem we tackle – our focus is on categorization. In his work, Crestani describes semantic networks as expressing knowledge about the relationships between concepts, for example through an is-a or instance-of based tree. Crestani also mentions what he describes as an associative network, by which he means a graph rather than a tree-shaped semantic network with no formal requirements that relationships represent a hierarchical is-a or instance-of relationship. However, he does not go into much further detail there. Our definition of associative networks is mostly identical.

Unlike Crestani, Pirolli et al. [1998] do use spreading activation for document categorization, but their work requires an existing collection of linked documents, for example created by hyper-link extraction. Associative networks in contrast require no data beyond the text of the documents and an existing associative network. Another difference between Pirolli's method and our use of associative networks is that Pirolli attempts to identify the key features of various categories from the data, thus filtering out less relevant features.

Our associative networks by contrast cover all features, even those which are not relevant, using the fact that associations will have a high spread into relevant nodes (because multiple nodes feed into them) while having only a low spread into irrelevant nodes.

4.3.3 Interest Profiles

Nanas and Roeck [2009] propose another connectionist technique. It is based on user interest profiles and has been used for determining the search interests of users repeatedly using a search engine. This technique constructs a network of terms for which searches have been made in the past and adds words to the user's interest profile based on the topics of the selected result, similar to how association concentration identifies relevant related topics. The new topics are then used to identify what documents may be of interest to the user, which hopefully allows the search engine to produce better results for that individual.

In contrast to association concentration, the user interest profile method filters possible results based on word co-occurrence in previously selected documents rather than based on related meaning. Additionally, the network applies weights to concepts rather than relationships, giving higher value to more relevant terms, and propagates its activation upwards (towards terms perceived by the algorithm to be more relevant) only. This makes sense for user interest profiles, as you only want to match more relevant terms, but associative networks do not require such a limitation: the interconnectedness of the associative network already ensures that nodes which are relevant get more activation than those which are not.

4.4 Text Classification Techniques

We have used associative networks for both classification, in which documents are matched to a specific predefined class, and categorization, in which the groupings have to be chosen and named by the system without outside help. In this section we look at some of the many other works in the areas of text classification and categorization.

Before classification can begin, the associative network requires that documents are converted to a bag of words, a simplification of a document that disregards grammar and word order, described as a linguistic technique as early as 1954 [Harris, 1954]. In this,

it follows many other text classification techniques such as Support Vector Machines and Naive Bayes which also use the bag of words approach [Joachims, 1998, McCallum and Nigam, 1998]. It should be noted that some of the techniques described here are not dependent on using a bag of words, but may instead use bigrams, trigrams or even a mix of these [Collins, 1996]. Such techniques are not applicable to associative networks, which rely on the concepts expressed by the words.

4.4.1 Vector Space Techniques

TF-IDF [Salton and Buckley, 1988, Hiemstra, 2001] and other Vector Space models [Soucy and Mineau, 2005] are well known methods for text categorization that use similarity in words to categorize documents with techniques such as Support Vector Machines [Cai and Hofmann, 2004]. These solutions primarily consider the presence or absence of keywords and make a statistical analysis of word frequencies. As such, they are unable to draw upon the conceptual meaning behind the text, which limits their ability to find matches. For example, the sentences '*The fast black car was winning.*' and '*The speedy dark vehicle headed for victory.*' have roughly the same meaning, but do not share any words beyond *the*. This leaves TF-IDF and Support Vector Machines unable to find the similarity. Association concentration is not limited in this way; it will provide a very similar activation pattern for the two example sentences as activation spreads out and concentrates in the same group of nodes.

Latent Semantic Analysis or LSA [Deerwester et al., 1990, Wiemer-Hastings, 2004] is a vector based approach that appears at first glance to share some properties with associative networks, notably in that they both attempt to extract the deeper meaning from the text through linking related words. While associative networks are created with these links from an authoritative source and find weights for them, LSA constructs the links from training data. This can lead to good results, but LSA may easily make invalid connections between words that happen to coincide in the training data by chance. Associative networks, though dependent on an existing, quality network where relationships are known to be valid, do not suffer from such problems. The different creation method does mean the associative network requires more information, however (i.e. the network itself). Additionally, LSA

is unable to deal with negation – a natural result of the bag of words approach. Like LSA, associative networks use a bag of words, but since WordNet links antonyms together, activation spreads easily over the negation barrier. For example, in the sentence '*The race car wasn't slow.*' the word *fast* will receive activation not just from the word *race-car* but from the word *slow* as well. Where WordNet provides antonym links very explicitly, other sources such as Wikipedia also provide such relationships because a word and its antonym cover a similar topic even with inverted meanings. We cover this use of sources and the effects they have on associative networks in Chapter 5.

Ferilli et al. [2010] propose two methods which rely on word co-occurrence and share some similarities with associative networks. However, like LSA, they rely on finding valid connections through co-occurrence. This makes them subject to the same problem of potentially invalid connections through coincidental co-occurrence.

It should additionally be noted that associative networks operate much faster even in the worst case (See Chapter 3) and are computationally more efficient than the performance listed for both LSA and Ferilli's methods. This makes associative networks far more viable for a live environment with large libraries of documents being edited, added and deleted by multiple users simultaneously.

4.4.2 Knowledge-based Techniques

Other solutions for text categorization are knowledge-based systems that use pre-existing domain knowledge such as decision trees [Apte et al., 1998]. Such systems require additional data about the problem space, which must be provided for each domain, while an associative network will work on any set of documents and the only information that is required is an existing network and knowledge of the language in which the documents are written to initialize it.

Like associative networks, concept mining [Shehata, 2009] is able to use the underlying meaning of the text to find relationships between documents. This and other approaches based on natural language processing [Ekedahl, 2008] have produced good results, but require more information than associative networks, for example regarding grammar and sentence structure. This makes these systems more difficult to construct and some have

limitations when dealing with short sentences or incorrect English syntax. By contrast, associative networks do not require any knowledge of syntax and they do not need to correctly parse sentences to determine their deeper structure. Associative networks require only the words in the text, which can be determined swiftly and easily.

Some efforts have been made to build hybrid classification systems [Forman and Suermondt, 2008] that combine several text classification techniques, using a weighted voting system to get the best results – associative networks could be incorporated in such systems as well, as the hybridization places very few requirements on the classification techniques used.

Various systems that somewhat resemble associative networks exist, some founded on similar psychological models. Some of these [Wichert, 1998] require a predefined taxonomy or thesaurus on particular areas that is more difficult to construct than an associative network, which can be initialized using simpler and more general language data. This required predefined taxonomy or thesaurus can often be created for a specific area, but constructing a predefined taxonomy for all languages is very difficult. Other systems [Tikk et al., 2003, Bang et al., 2006] are more limited in the scope of the surrounding concepts they can access, not spreading as widely. As a result of the method used, they do not benefit from the filtering of irrelevant relationships that associative networks are capable of by using association concentration.

4.4.3 Supervised Learning Techniques

The method of associative networks described in this thesis is a supervised learning method, that is, training data is provided in the form of sample data with the desired output, and the associative network is trained based on this data. We detail our methods of training in Chapter 6. This method differs from unsupervised learning, where a hidden structure must be found in otherwise unlabelled data.

Naive Bayes [McCallum and Nigam, 1998] is a family of algorithms based on the assumption that the values of all features are independent of one another. Using the bag of words of a document for example, each word would represent a separate feature, and the value of that feature would be the number of occurrences in the document (its word

count). Naive Bayes then presumes that the probability of each word is independent of the probability of other words, and calculates the probability, based on the bag of words, that a document belongs to each class. An advantage is that this requires only a small amount of training data.

Support vector machines [Cortes and Vapnik, 1995] use documents as points in a high dimensional space and the coordinates of those points are based on features of the document. A hyperplane or set of hyperplanes is then constructed to separate classes within that space. A kernel function may be used to transform the original data into a new, higher dimensional space to allow for a better separation.

Another supervised learning method is k-nearest neighbours [Keller et al., 1985] by which documents are classified as the most common class amongst the k (a small positive integer) closest documents. Often there is some weighing factor, with closer documents being weighed more heavily than more distant documents when determining the most common class. Different metrics can be used to determine the distance between documents. Even the distance between activation patterns could be used as a distance metric.

We compare the performance of associative networks to Naive Bayes, support vector machines and k-nearest neighbours in the experiments described in Chapter 5 and 11.

Part II

Experiments

Chapter 5

Experiment 1: Creating Associative Networks

"Simplicity is prerequisite for reliability." - Dijkstra [1982]

In this section we describe our work on the creation of associative networks, specifically looking at different resources that can be used as a basis such as synonym lists, WordNet and Wikipedia. We then compare some of these methods to see which ones produce better text classification results, and compare those results to the work of others in regards to document classification.

Creating an associative network is Step 1 of the overall document categorization process as depicted in Figure 5.1. A short version of the work described in this chapter was published as *'Using Wikipedia with Associative Networks for Document Classification'* in

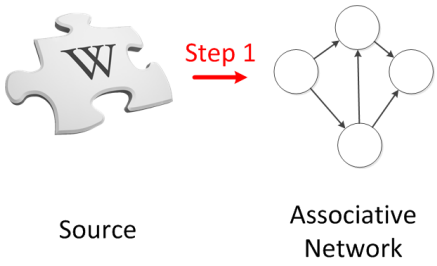

Figure 5.1: Creating associative networks in the categorization process

the proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning [Bloom et al., 2013b].

5.1 Resource Selection

Associative networks can be created using different resources. A key requirement is that a resource must specify relationships between concepts. In general, the better the resource, the better the results that can be achieved by the associative network. In this section we describe three possible resources for the creation of associative networks.

5.1.1 Synonymy-based Associative Networks

The first type of resource is synonym lists. Synonym lists have the advantage of being easily available, and provide links between words of similar meaning, which lie at the core of associative networks. However, it is not usually possible to use word sense disambiguation with synonym lists as word sense information is not usually available with those lists. This may cause links between unrelated words that happen to share a surface form to be created within the associative network (see Section 7.1.1), and association concentration can thus spread through those links and create incorrect associations.

Additionally, due to the incompleteness of synonym lists, many terms are not connected at all despite being related. Links are only made between words that share the same meaning. This generally covers most is-a relationships, but part-of relationships are skipped altogether. For example the words *bird* and *wing* are not synonymous, but they are conceptually related by association. A synonym based associative network would not contain this connection, however.

For the Dutch language, we combined several synonym lists[1] that were made available on internet sites, building a network in which each word is a node and each link indicates that two words are synonyms.

[1] `http://www.mijnwoordenboek.nl/synoniemNL.php`, `http://www.opentaal.org/projecten/synoniemen` amongst others

5.1.2 WordNet-based Associative Networks

Princeton WordNet [Miller, 1995, Fellbaum, 1998] is a well-documented lexical database for the English language that groups synonymous words into groups called synsets and records various semantic relations between them. It is used in many scientific projects. Unlike synonym lists, it does not suffer from problems such as incorrect links between words that are not actually synonymous (but which do share a surface form). Additionally, WordNet features part-of relationships that are absent in synonym lists; for example, the words *bird* and *wing* are linked as holonym / meronym, a semantic relationship representing the part-of relationship. WordNet provides both a thesaurus and dictionary data in one system, is well documented and supported, and is a de facto standard. With all these advantages, it is a natural choice for use with English documents.

A WordNet based associative network can be initialised by creating a graph consisting of a single node for each synset in WordNet. These nodes are then connected according to the relationships between the synsets. For example, the synset for *fly* which describes flies as '*two-winged insects characterized by active flight'* has the sister-term *mosquito* (which indicates the two share a hypernym) and a hypernym *insect* so the nodes for these synsets are connected.

To link words in a document to the corresponding synsets, we can use various methods, which we describe in detail in Chapter 7. Simply put we link synsets to the surface forms of the individual words and we improve on that using Part-of-Speech tagging and Natural Language Processing.

A limitation of WordNet (as well as synonym lists) is that it does not provide a way to set weights for the connections between synsets – all that is provided is the type of relationship, but the level of connectedness may vary wildly within a single type. For example, the word *jeep* has the sister terms *sports utility vehicle* and *bus*. While all three are cars, jeeps and sports utility vehicles are more closely related to one another in terms of shape, function and abilities than they are to a bus.

Moreover, WordNet is more detailed in some areas than in others. This means some concepts are further away in the graph even though they are more closely related due to the level of detail in that locality of the WordNet network. The word *bus* has hyponyms

for *minibus*, *school bus* and *trolleybus*. This means the distance from *minibus* to *car* is two steps, first from *minibus* to *bus* and then from *bus* to *car*. By contrast, the distance from *minivan* to *car* is only one step, it being a direct hyponym. Few would argue that a minibus is conceptually more distant from a car than a minivan.

In an associative network, it is not necessary for all related concepts to be connected directly – a strong connection through a different node will also spread the activation properly, but only if the weights are set correctly. Thus, while the different levels of detail in WordNet are not an issue directly, we cannot use absolute distance as weights and need to find another way to determine the weights.

In our work, we initialise WordNet based networks with each edge being equal at a value of 1. Through training (explained in Chapter 6), we can then adjust those weights to come closer to their conceptual values.

A final limitation is that Princeton WordNet covers English only. Other WordNets do exist [Bond and Kyonghee, 2012, Vossen et al., 2007]. They are not as extensive as the English WordNet, but could be used to handle different languages.

5.1.3 Wikipedia-based Associative Networks

An alternative resource for creating an associative network is Wikipedia. It covers a large number of topics and most words and concepts have their own article or are mentioned as part of a greater overlapping article. Moreover, unlike WordNet, Wikipedia covers most well-known proper names and links these to related concepts. This may not hold for very specialized fields however.

Wikipedia consists of connected articles, each explaining a concept and providing links to related concepts. At first glance, Wikipedia appears to offer less information than WordNet, not having a type (such as hypernym, synonym or antonym) associated with each relationship. However, with Wikipedia articles we can gather additional information as to how strongly two concepts are linked, based on how often the linking term is used in the articles describing those concepts.

We cannot determine the strength of the connection between linked articles purely by the number of times one article links to another, because Wikipedia authors are encouraged

not to create multiple copies of a link to the same concept within an article to improve readability. Furthermore, though this goes against Wikipedia's manual of style, some links may not be relevant to the topic of the article, but may simply be an unrelated word that happens to be used somewhere in the article.

A better indication of how strongly two articles are linked might be the number of times the title of one article is used in the other (and vice versa), even when it is not a link. However, this disregards the use of pronouns and synonyms which conceptually refer to the title without literally mentioning it, which may obfuscate a relation to the target article. Since this use of synonyms and descriptions is exactly the way associative networks compare articles, it makes sense to use an associative network to judge how closely two linked Wikipedia articles are related. But this leaves us with a boot-strapping problem – we need an associative network to establish the connection between two articles, and to create that associative network we need an associative network.

To resolve this conundrum, we create our associative network in two steps. First, we create a basic associative network using Princeton WordNet, as described in the previous section and train it (for details of the training method, see Chapter 6). This associative network relies purely on the WordNet links and training. We then use this network to create an associative network based on Wikipedia. Potentially, this process could be carried out again, building a new associative network using the improved Wikipedia-based associative network.

In our case however, this repetition of moves was considered to be outside of the scope of the research. If a single stepped solution does not create a good associative network, there is no reason to assume a second step using an equal or inferior associative network would offer further improvements.

A final advantage of Wikipedia based networks is that they are multilingual, offering articles in a large variety of languages. Though not all Wikipedias are equally exhaustive, in general Wikipedia covers the commonly spoken languages very well, which makes adding multiple languages to associative networks very simple – all it requires is the inclusion of those articles and the use of the direct translation links that Wikipedia already makes available. We cover the topic of multilingual associative networks, based on Wikipedia, in Chapter 10.

5.2 Related Work

In this section we examine some other works that have used Wikipedia as a resource of information about language.

Mihalcea [2007] uses Wikipedia's internal links for word-sense disambiguation. Since these links lead to a unique page, she used the alternative text (the text that is actually displayed in the linking article, which may be different from the title of the page it links to) to get a sense of the context that could be used to identify the correct sense of a word.

Nastase and Strube [2013] extract a network of related concepts based on Wikipedia in multiple languages, using Wikipedia's base network of articles and categories. They enhance their network with new relationships based on the base network. We also create a network using Wikipedia articles as nodes, but in our case, the link is created based on the text content of the article, not based on the Wikipedia category structure, as we believe the text content to be a more accurate and complete description of the concept described by the article.

Malo et al. [2010] use Wikipedia articles as concepts, with the links between these articles as links between those concepts. They show that Wikipedia offers a unique combination of scale and structure for this purpose. We use Wikipedia in a very similar manner, but Malo's goal is different from ours – their work covers query based information retrieval rather than classification. As a result, their methodology is different from ours.

Malo's work focusses on using Wikipedia as an ontology to expand the search space of their information retrieval system – in this there are again parallels to association concentration. A key difference is that activation using Malo's method spreads only one step outward from the search query to related terms, relying purely on the distance measure to determine relatedness. This makes sense for their goal of information retrieval, where queries may be limited to one or two words, but for our use in text matching and document grouping, where more information is available, association concentration can spread beyond the direct neighbours, which means activation can spread further and thus the points in which the activation is concentrated will be more pronounced there. The underlying principle between their work and ours is the same however: using knowledge inherent in Wikipedia to get a conceptual understanding of the text.

5.3 Description of the Experiment

We carried out an experiment to compare text classification using WordNet and Wikipedia based associative networks both to one another and to other text classification techniques. To do this, we use a Reuters set, a standard test set for classification, to test our performance. We generated a micro-F1 score[2] for a WordNet-based and a Wikipedia-based associative network. The WordNet-based associative network was also used to construct the Wikipedia-based associative network. We did not test a synonym based associative network because WordNet already contains synonymy information.

5.3.1 Classification and Training

In the Reuters set, documents can belong to more than one class, which means some adjustment of our algorithm as described in Chapter 2 is required. To deal with multiple classes, we created a merged class activation pattern for each class, consisting of the combined and averaged activation patterns of each training article in that class. This means that an article that belongs to two classes is used in the class activation pattern of both.

To allow the associative network to sort documents into multiple classes, a threshold was also added to each class (see Chapter 3). An article is matched to a class if the match between the article's activation pattern and the class activation pattern exceeds the threshold value.

Training was used to adjust the weights in the associative network based on the training documents in the dataset. Depending on whether or not the associative network identified the correct classes for each document, positive or negative reinforcement was applied through back-propagation to the paths between the actual text input and the matching between the document's activation pattern and the class activation pattern. Thus, certain links in the associative network were strengthened while others were weakened. We describe the training of associative networks in more detail in Chapter 6.

[2]A micro-F1 score is the mean of the micro-average precision (average precision over all instances) and micro-average recall (average recall over all instances) .

5.3.2 Reuters Dataset

To be able to compare our algorithms against competing systems, we tested them using the Reuters-21578 dataset. This dataset was compiled in 1996 by David Lewis in collaboration with Steve Finch, specifically for the purpose of allowing clear comparisons between different text categorization methods [Lewis, 2004]. We used the *ModApte* split, a standard split of the Reuters set, which consists of a training set of 9603 documents and a test set of 3299 documents over 90 different categories. To evaluate the performance of our networks, we compared our algorithm against the various results reported by Joachims [1998] who used the exact same dataset. We were unable to find a more recent evaluation on the same exact dataset, as many researchers either did not specify exactly or used modified or different versions of the Reuters dataset. In Chapter 11, we compare associative networks to a more current state of the art.

The Reuters dataset has been used countless times to compare text classification methods [Debole and Sebastiani, 2004], such as rule-based models [Apte et al., 1994], statistical approaches with Support Vector Machines using unigrams or bigrams [Yang, 1999, Bekkerman and Allan, 2004], and various inductive learning methods [Dumais et al., 1998]. Rather than discussing them all in detail, we merely give a short listing of the methods drawn from [Joachims, 1998] which we use for comparison in our experiment. The naive Bayesian classier [Lewis, 1998] is a probabilistic model of the term density, commonly used for text classification and information retrieval. The Rocchio algorithm [Rocchio, 1971] is another popular learning method based on relevance feedback. We also compared to a k-nearest neighbour classier [Yang, 1999] and the C4.5 decision tree / rule learner [Quinlan, 1993]. Finally we compared to the best Support Vector Machine result from Joachims [1998].

5.4 Results

Table 5.1 lists the micro-F1 scores of the other algorithms listed above, as gathered by Joachims [1998]. Table 5.2 shows the results for the methods we proposed on the same Reuters corpus.

	Micro-F1 score
Bayes	72.0
Rocchio	79.9
C4.5	79.4
k-NN	82.3
SVM	86.4

Table 5.1: Results on the Reuters Set using various methods [Joachims, 1998]

	Micro-F1 score
WordNet-based associative network	83.4
Wikipedia-based associative network	85.8

Table 5.2: Our own results on the Reuters Set using associative networks

From the results in tables 5.1 and 5.2, we can see that Associative Networks perform at a level that is on a par with other text classification techniques of 1998 (we make a more up-to-date comparison in Chapter 11). Though not beating support vector machines, especially the Wikipedia-based associative network managed to attain similar scores, and those scores can still be further improved using the techniques described in our other experiments.

A second notable result is that Wikipedia-based associative networks outperform WordNet-based associative networks by several points.

5.5 Conclusions

We have shown that associative networks are a match for other techniques, scoring similar to the top algorithms gathered by Joachims [1998] on the Reuters set. In other experiments we made improvements using Natural Language Processing (see Chapter 7) which allowed better disambiguation between homonyms and better detection of relevant words, thereby increasing performance. These techniques were not used in this work. Combining a Wikipedia based network with Natural Language Processing techniques should allow associative networks to outperform all their competitors. We test this in Chapter 11.

With Associative Networks based on Wikipedia outperforming WordNet-based versions, we furthermore show that being able to model the strength of relationships between

concepts can help improve the quality of associative networks. Our method for qualifying the weight of article relations based on the article's textual content might benefit further from iterative improvement of the network as described earlier in this Chapter, though we did not test this further.

Another possibility that using Wikipedia-based associative networks opens up is the use of multilingual data (see Chapter 10), based on how Wikipedia articles are linked to their equivalents in other languages, which will allow documents in multiple languages to be categorized together.

Chapter 6

Experiment 2: Training Associative Networks

"Muad'Dib learned rapidly because his first training was in how to learn."
- from The Humanity of Muad'Dib by the Princess Irulan [Herbert, 1965]

In this chapter we describe the methods we have developed to train associative networks once they have been created. Training an associative network takes place in Step 2 of our general process as depicted in Figure 6.1.

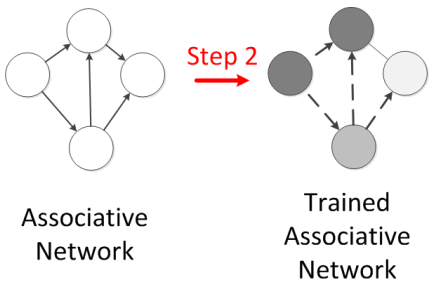

Figure 6.1: Training associative networks in the categorization process

6.1 Back-Propagation

In the previous chapter we described how an associative network can be created by using synonym lists and WordNet as potential sources to tell us which words are related. But while these sources provide crucial information about whether or not two words or concepts are related, they do not provide any information about how closely they are related; after all, it cannot simply be presumed that all links provided by these sources are of equal value, as some pairs of words will be more closely related than others. One solution proposed in Chapter 5 is to use Wikipedia as a source to help establish the strength of the link between concepts, but even here the distance between concepts discovered using our recursive method of extraction (see Chapter 5) may not be accurate in all cases.

Knowing this, we wish to improve the associative network after creating it to provide a more accurate representation of the conceptual distance between words. Arguably, there exists no perfect answer to what the conceptual distance is between CHAIR and BENCH for example. Even if we merely wish to compare it relatively to the conceptual distance between CHAIR and SEAT, such an answer may not exist. However, we can say whether or not certain values for a conceptual distance are more or less successful in creating an associative network useful for document categorization.

To make associative networks learn the most effective conceptual distance between nodes, they can be trained to give more weight to closer connections and lower weight to more distant ones. As a result, when a node in the trained associative network is activated, activation is spread towards more closely related nodes more easily.

In our framework we use text as training data. Each document from a library to be categorized is used to create an activation pattern by spreading activation from the words in the document (see Chapter 3). The activation patterns of the input document and the other documents can be compared, to find the most closely related document which we call the output document. A supervisor can then inform the network whether or not the association between those two documents was correct and if they were indeed the most closely related documents in the set. If so, we strengthen the associative network through the relationships between the two documents. If not, we weaken them instead.

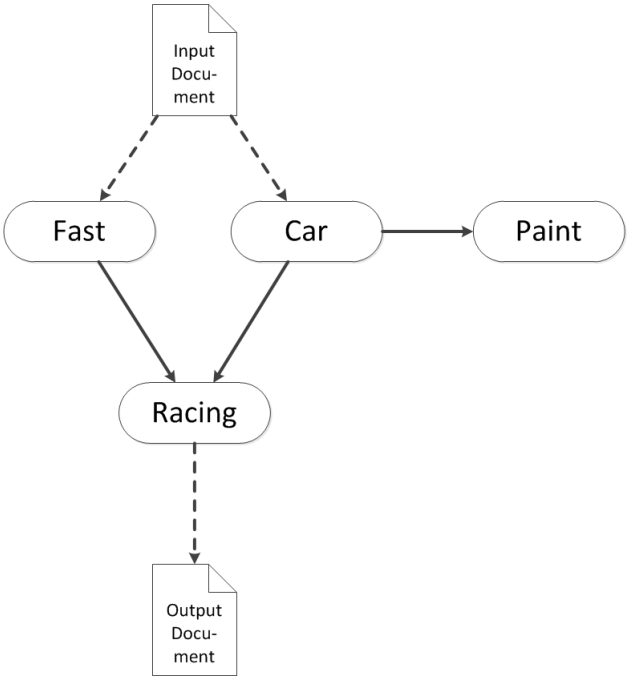

Figure 6.2: Simplified association sub-graph representing the link between two documents

To know which edges in the associative network to strengthen or weaken, we need to find the connections through the associative network between the input and the output document, as these paths represent successful connections. To find the exact paths, and to calculate by how much we should modify the various paths, we use a method originally developed for neural networks: back-propagation [Rumelhart et al., 1988].

Before we can apply back-propagation, we need to determine the way activation could spread from the words in the input document to the words in the output document. To do this, we first make two virtual nodes, one for each of the documents, connected to the nodes for each term in the document by a value proportional to that term's presence within the document. From this, we then construct what we call an association sub-graph of the document pair,

An association sub-graph is a directed a-cyclic sub-graph of the associative network that represents how an activation pattern was created using association concentration. The details on how it can be created are described in Chapter 8 and a strongly simplified example is depicted in Figure 6.2.

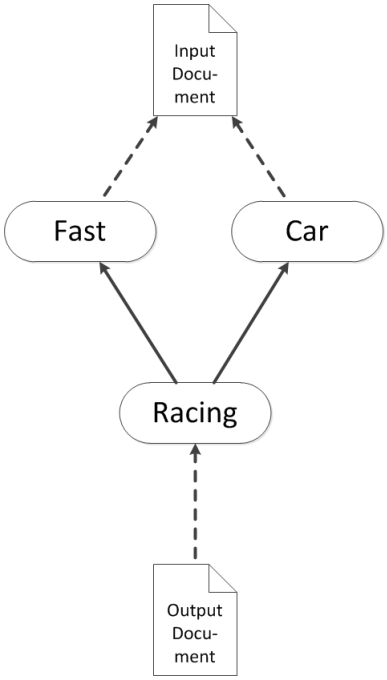

Figure 6.3: Pruned and reversed association sub-graph

As the association sub-graph contains all the paths going out of the virtual input document node (even those that do not lead to the virtual output document node), and we only need the paths between the input and the output document, we modify the association sub-graph by pruning all edges and vertices that are not part of a path between the nodes in the activation patterns of the two virtual document nodes. As the graph is directed and a-cyclic, we do not have to worry that circular paths might lead to the inclusion of irrelevant nodes.

After we prune those edges and vertices we are left with an even further simplified graph representing the connections from the input document to the output document. We then reverse the edges, as depicted in Figure 6.3. This reversal, a key feature of back propagation, allows us to reinforce connections from the answer (of which we know the correctness) back towards the input.

We reinforce the connections in the trimmed sub-graph if the document pair was correctly linked, by increasing their weight in the associative network. If the result was incorrect, we weaken those connections by lowering their weight in the associative network.

For future categorization, the network should then be able to generate associations along correct lines more quickly, while making associations along incorrect ones less easily.

When a document was incorrectly linked in an earlier cycle, after adjusting the network we recalculate the association sub-graph and generate a new result based on the new association sub-graph, which is again evaluated automatically by the supervisor allowing the weights of the connections to be further adjusted. This can be repeated until the association sub-graph produces the correct results.

The method described above was used as a training method in most of our experiments. In the experiments in Chapter 7 and 8 it was applied by creating a training library composed of 30 manually selected Wikipedia articles, with each article being closely related by topic to exactly one other article and not related to the other 28 articles, thus forming fifteen pairs of articles. The topics of these articles were similar to the ones we wished to categorize. A newly initialised associative network was activated for each of the 30 articles in random order to determine which were the most closely related. Depending on the correctness of the result, positive or negative reinforcement was applied by using back-propagation to adjust the weights in the network. This cycle was repeated until the associative network produced the correct matching article for the entire library. By training the associative network in this way, we were able to increase the weight of edges between closer concepts such as FLY and MOSQUITO while reducing the weight of edges to more distant concepts such as INSECT which also includes much less similar creatures such as ants and crickets.

6.2 Montessori Method of Training

Based on the idea that associative networks model the language of the nervous system (see Chapter 1), we wondered if advances from the field of education might be used to improve the training for associative networks, as they have for humans. We specifically looked at the work of Maria Montessori, who at the start of the 20th century developed a method of education based on her own model of human development. This method is characterised by an emphasis on independence, freedom within limits and respect for the child's natural psychological, physical and social development [Montessori, 1909].

Montessori's method calls for free activity within a "prepared environment". Thus, rather than specifying the exact lessons to be learned, the child is free to explore an environment that has been set up specifically to teach certain types of lessons. The function of the environment is to allow the children to develop their skills using their own inner psychological desires as incentive.

As an example how the environment is prepared for the child, Figure 6.4 depicts a set of 10 pink blocks used in teaching children the concepts of bigger and smaller. Each block is identical in shape, colour and material. These properties are kept equal to reduce the complexity of the problem, in turn allowing the child to learn about bigger and smaller without those other factors getting in the way. Unlike colour, shape and material, the weight of the blocks is deliberately allowed to vary, with bigger blocks being heavier than smaller blocks. This is done, as unlike colour, shape and material, the weight of objects is in fact related to their size and thus, it is a property that is relevant to the lesson of bigger and smaller.

The Montessori method of education teaches a young child about the concepts of bigger and smaller by eliminating unrelated factors. Using that same philosophy, we propose a method for training associative networks that we have dubbed the Montessori method of training. It uses a prepared environment with a limited set of training data. Rather than using documents from a real environment to form a set, training is done on an especially prepared set of examples.

Like the ten blocks in the pink tower, these examples should be as similar as possible, except for the property we are trying to teach to the associative network. For example, the article for carnivorous animals was as follows: *"Carnivorous animals eat the flesh of other animals exclusively. They often kill their pray with teeth or claws, but may scavenge for meat instead. Many have forward facing eyes to aid in hunting. Carnivores often occupy the top of the food chain."*. It contains a reference to *'hunting'*, something most carnivorous animals do, as well as *'teeth'* and *'claws'*. These related concepts, in parallel to the weight of the pink tower blocks, are used because they are relevant to the core concept being trained.

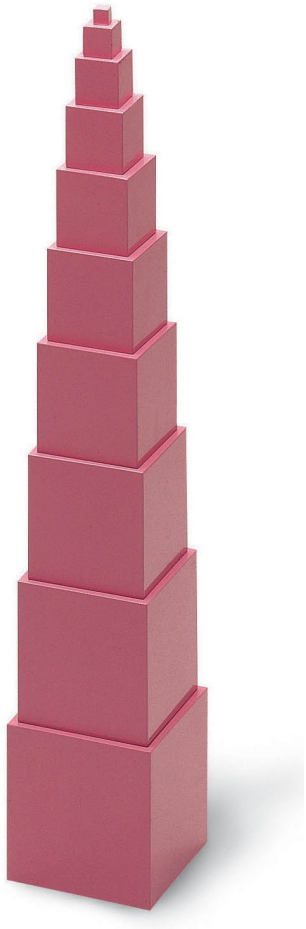

Figure 6.4: Pink Tower used in Montessori Education

6.3 Description of the Experiment

To test the difference between training on regular data and Montessori training, we compared how associative networks, trained using these methods, performed in terms of categorization accuracy.

We constructed two WordNet-based associative networks in the manner as described in Chapter 5. Both these associative networks were identical after creation, but were given training on different datasets, one being trained on actual sample data and one being trained on specially prepared training data following the Montessori method (see below). The experiment was repeated five times with different categories.

6.3.1 Design of the Experiment

We created five test sets, each containing 50 articles and each set within a single Wikipedia category. The five categories were manually selected from Wikipedia, based on the criterion that they should correspond to general topics with many articles and sub-categories. Those categories were *Animals, Biology*, *Chemistry, Nature* and *Philosophy*.

Next, five sub-categories were randomly selected within each of those categories. Sub-categories for which there were less than twenty articles were excluded. Additionally, articles which had less than one thousand words, or which were marked as a stub or list article were also excluded.

Once these sub-categories had been selected, ten articles were randomly selected for the test set. For five sub-categories, this gave a test set of 50 articles per category. For example, the category *Biology* might have *Genetics*, *Biochemistry*, *Mycology*, *Neuroscience* and *Ecology* as sub-categories, and thus would have ten articles from each of those sub-categories.

Two trained associative networks were then set up to establish which article belonged in which of the five sub-categories. The sub-categories were found based solely on textual content of each article. No additional information such as links was used, and an accuracy score was established based on how many articles were sorted correctly, using the following formula.

$$Accuracy = \frac{CorrectlySortedArticles}{TotalArticles} * 100\% \tag{6.1}$$

Thus, for example, if 40 out of 50 articles were sorted correctly, the accuracy score would be 80%. Accuracy is used as it represents the closeness of the categorization to the true value [BIPM, 2008]. which is crucial when making systems to be used in practice. In unbalanced datasets, the accuracy paradox describes a problem that a higher accuracy score does not necessarily reflect a better classifier. Because of the balanced nature of the created dataset, we avoid the complications caused by the accuracy paradox.

Sorting with the regularly trained associative networks was done by comparing the activation pattern of each article from the test set to the training articles for which the correct sub-category was known. The article from the test set was sorted into the sub-category of the closest article from the training set.

6.3.2 Creating the Training Data

Creating the training data for the regular training associative network was reasonably straight forward. A training set was constructed by extracting a further ten articles from each subcategory, different from the test set. Those articles were used to train the associative network using back-propagation, in the manner described earlier in this chapter. Thus, the training of the regular associative network was similar to training for the experiments in other chapters.

Because the Montessori method of training requires a specifically prepared training set, the training set used for the regular associative network cannot be used for training with the Montessori method. Instead, a training set was manually constructed. Each sub-category was associated with a single article, which we manually constructed, to represent the concepts presented by that sub-category. This article would consist of roughly three sentences describing the core of the concept and related properties. This was done based entirely on our own insights, so to some degree the constructed articles were subjective. In writing these articles, we took a careful look at the dictionary definition of the term to aid us however. Negative forms (such as *"Cats cannot fly"*) were avoided.

	Regular Training	Montessori Training
Accuracy	56%	60%

Table 6.1: Average results of the two training methods

The articles were designed such that each manually constructed article followed the same structure and described only the key aspects of the sub-category. This focus on the core essence also meant that these manually constructed articles were smaller than the regular training articles, each containing significantly fewer words than the minimum of 500 words set for the training samples, usually around 25-50 words.

6.4 Results

The averaged results of our five experiments are listed in Table 6.1. The score for the Montessori training is higher than for the regular training, but the difference is not very large: on average, the Montessori trained network managed to get two more articles classified correctly than the regularly trained one, on a rather small test set.

One thing to note is that the results of this experiment are lower than the results of other, similar experiments in different chapters. The reason for this lies within the datasets gathered. We found that the articles that were randomly selected relatively often lay close on the border between two categories.

In one case an article *'Shark'* could be argued to be present in two categories, which was something we had not accounted for in our random data collection. The two subcategories, *'Aquatic animals'* and *'Carnivorous animals'* arguably both fit an article about sharks. Moreover, however, Wikipedia categorized the article under *'Carnivorous animals'* directly and under *'Aquatic animals'* via the sub-categories of *'Fish'* and *'Commercial fish'*. Since this categorization is hierarchical, this means both subcategories would in fact apply to the article on sharks.

In this experiment, the Montessori training samples were made based on our own insights. It would be necessary to more formally define and examine different methods of constructing these samples to compare which method improves the results the most. Some possible sources for such training samples might be constructed from ontology databases

or common sense knowledge bases such as those created by Lenat [1995]. Dictionary definitions might also be helpful, though the mere dictionary entry - which by necessity of the medium will be quite short - is unlikely to contain enough information to create relevant training data. One advantage however would be that such a dataset would need to be constructed only once for a language, after which it would be reusable for many different projects.

6.5 Conclusions

The differences found between the two types of training in our experiment were small, but enough to suggest that Montessori training provides equal or better results than regular training.

Because of the time required to create a dataset for Montessori training, and the relatively small improvement made however, it is difficult to put Montessori training forward as a viable alternative for regular training. A key factor to make Montessori training a success in the future is therefore to reduce the time required to create the training dataset. Since such a dataset only has to be made once for a language, after which it could be reused in many different situations, it might be a worthwhile investment.

Chapter 7

Experiment 3: Natural Language Processing

> *" 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' In one case a member of the Upper, and in the other a member of the Lower House put this question. I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question."*
> *- Babbage [1864]*

An information system, no matter how advanced, is only as good as the quality of the data that is fed into it. This principle was already described by the inventor of the first programmable computation device, Charles Babbage, and has since become known as the *'garbage in, garbage out'* principle [Lidwell et al., 2010]. When using associative networks and association concentration to categorize documents, the primary system input consists of bags of words, extracted from documents. If we can improve the quality of those bags of words, for example by removing incorrect or irrelevant entries (the proverbial *'garbage'*) from it, we should in turn be able to improve the output of the system.

We created two methods by which we can improve the quality of a bag of words using Natural Language Processing techniques. In this chapter we describe the two techniques we used, specifically part-of-speech tagging [Toutanova et al., 2003] to remove incorrect words from the bag and natural language parsing [Klein and Manning, 2003] to give additional weight to key words in the sentence.

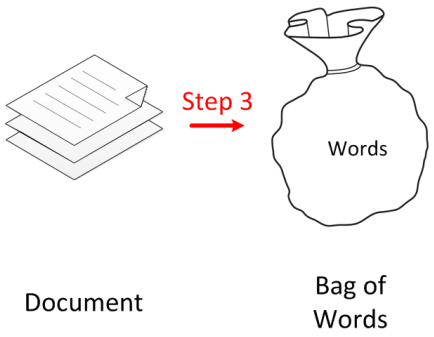

Figure 7.1: Natural Language Processing in the categorization process

These techniques are used in Step 1 of the process of document categorization that was described in Chapter 2, and that is depicted in Figure 7.1. A short version of the work was published as '*Using Natural Language Processing to Improve Document Categorization with Associative Networks*' in the proceedings of the 17th International Conference on Applications of Natural Language to Information Systems [Bloom, 2012b].

7.1 Building a Better Bag of Words

Before we can activate the associative network for a document, we need to convert the words in that document to a bag of words. Most methods of constructing an associative network (see Chapter 5) use the dictionary form of words, rather than having a separate entry for each surface form of each word. As the dictionary form is often referred to as 'lemma', the converted input could also be described as a 'bag of lemmas'.

Natural Language Processing techniques such as part-of-speech tagging may aid in this conversion, and have been used in the past to improve document categorization. An example was presented by Ekedahl [2008], who reported a positive impact from part-of-speech tagging for both support vector machines and string matching categorization. Based on his results, it is reasonable to presume that document categorization by associative networks can benefit from that technique as well.

In our experiment, we use the same Natural Language Processing technique to build a better bag of words for use in association concentration. We applied part-of-speech tagging to better match between the lemmas and the surface forms found in the document. As a second method, we used dependency parsing to identify which words are more (or less) relevant to the text in the document, and give more relevant terms a boost in terms of the strength with which they are fed into the associative network.

7.1.1 Matching Surface Forms to Lemmas

It is quite easy to convert a document into a bag of words. We can use a naive tokenization method to decompose the document into words by using punctuation marks and spaces as word boundaries. However in practice this is not enough to get an accurate bag of words [Habert et al., 1998].

A lemma can have more than one surface form; for example, both *flying* and *flew* are surface forms of *to fly*. From the perspective of the associative network, there is very little semantic difference between those forms for purposes of activating the network; after all, the values of the relationships that each of those surface forms would have in an associative network should be very similar. Thus, the benefit of having separate entries for each form is minimal.

Using basic morphological rules, such as those which describe how to construct plurals from nouns, or how to conjugate regular and irregular verbs [Quirk, 1985], we make a list of surface forms for each lemma in the associative network, which we then use to construct the bag of lemmas. Compound words which include spaces (such as *punctuation marks* or *surface forms*) are detected only if their individual parts appear consecutively[1]. This naive scheme is effectively a simple morphological dictionary and facilitates a somewhat reasonable approximation of the lemmas that are used in the document. Morphological dictionaries have proven useful for text categorization tasks in the past, such as Nakov et al. [2003]. While this method can be helpful, it does have its limitations.

[1] In English, used in this experiment, all compound words are consecutive, but in other languages such as Dutch they do not need to be (See also Chapter 10)

A very important limitation is that using this method alone, it is not always possible to handle ambiguity and to determine to which lemma certain surface forms actually corresponds. For example, the word *fly* can be a surface form for the noun describing the insect or a verb form expressing the act of flying. Likewise, the word *bat* can refer to a club used in baseball as well as to the flying mammal of the same name. The simplest way to deal with lexical ambiguity is to activate all concepts that correspond to the word - in the case of the word *fly*, this would activate both the verb and the noun, for example, regardless of which one is correct, which of course means adding some garbage to the bag of lemmas.

Associative networks are usually not very sensitive to the potential noise generated by this strategy. As they concentrate activation towards terms surrounding the general topic of a text, conceptually unrelated concepts that are erroneously extracted due to shared surface forms will filter out naturally already as they receive no further activation from the other, conceptually unrelated lemmas extracted from text. For example, if a document describes various insects, a single mix-up on the word *fly* will not change the general way the document is understood by the associative network very much, as little activation will spread to the verb *to fly*. Likewise, a document listing terms such as *sports*, *home-run* and *baseball* can be linked to its general topic by an associative network, even if the bag of words does contain multiple lemmas for the word *bat*.

However, despite the ability of associative networks to compensate for this type of noise, each irrelevant entry in the bag of words causes incorrect associations, reducing the overall accuracy of the associative network. Thus, even if associative networks are able to deal with the occasional irrelevant term in the bag of words, too much noise can shift the balance and cause incorrect categorization results. By improving the recognition of the intended lemma from the surface form, we can eliminate irrelevant data extracted from the input document and thereby improve the quality of the input, which in turn should lead to better results. In the experiment described in this chapter, we test the use of part-of-speech tagging to demonstrate that it indeed improves categorization.

It should be noted that various word sense disambiguation techniques [Mitkov, 2005], such as those used by Mihalcea [2007] might provide additional benefits to the basic part-of-speech tagging we use and many of those word sense disambiguation techniques are in fact direct improvements applied on top of the part-of-speech tagging technique. For

example, the word *bat* has different lemmas that share the same part-of-speech, so more advanced disambiguation techniques would allow more irrelevant lemmas to be eliminated from the bag of words than by using only part-of-speech tagging. We did not use these techniques in our experiment, however, leaving this for future work.

7.1.2 Weighted Lemmas

Not all words in a sentence carry the same importance. Certain words will have limited semantic value, such as the word *the* being unlikely to have much relevance to the topic of an article.

One simple method of identifying what is and is not important is by filtering out stop words [Fox, 1989] such as *the*, which generally add very little meaning to a text. This was one of the methods explored for document categorization by Ekedahl [2008]. An inverse alternative might be keyword detection, already used successfully for the task of document categorization by Wartena and Brussee [2008]. In their approach, certain keywords are identified as especially relevant through analysing a training set. Keywords could be given additional weight in the bag of words. However, in general neither stop words nor key word detection are be necessary when using an associative network: the associative network itself can – through the process of training – figure out which words are highly relevant (content words) and which are unimportant (stop words). Simply put, if words do not convey much meaning, their connections in the associative network will be weak and if they are important, the connections should be strong. Identifying content words that may or may not be important depending on the context in which they are used requires a more advanced method.

The advanced method proposed here (which is described in more technical detail in Section 7.2.4) uses the fact that in general certain parts of speech are more relevant than others (salience). The subject of a sentence is generally more salient than the indirect object or the adverbial. Likewise, a word in a sentence may be centrally linked to all or most of the other words in the sentence (by having more dependency relations within the sentence than any other word) which indicates that word may be more important.

For example, in the sentence '*I want to fly to Paris in a jet'*, the word *fly* is more important than the word *Paris* as the word *fly* is more central to the sentence: the subject wants to fly, they want to fly to Paris, and fly in a jet. *Fly* is thus related to each part of the sentence, marking it as the most important word in the sentence.

Identifying in this way what is important is somewhat subjective, as in many cases one could argue about which word is more important in a sentence. We do not propose to solve this with perfect accuracy – rather we try to make an informed guess as to which words are key words that can help identify the topic and thereby the category of an article. Since a document consists of many sentences and associative networks can compensate for some margin of error, the technique to help identify important words in each sentence can still be useful in improving the overall quality of the input data, even without perfect accuracy.

7.2 Description of the Experiment

In this experiment we examine the effect on classification performance of using natural language processing methods for matching the surface forms to lemmas and for the detection of the most relevant text elements to weigh lemmas, thereby creating a better bag of words.

The experiment is based on the hypothesis that eliminating incorrectly matched surface forms and adding extra weight to key words will improve the quality of the associative network's output. To test this hypothesis, we used the Stanford Part-Of-Speech Tagger [Toutanova et al., 2003] to help exclude lemmas that do not match the part-of-speech of the word in the document. By only matching surface forms to lemmas if their part-of-speech corresponds to the meaning of the lemma, we were able to eliminate noisy lemmas from the bag of words. We also used the Stanford Natural Language Parser [Klein and Manning, 2003] to detect key words in each sentence and boost their activation. In this way we hoped to create a bag of words which gives a greater weight to terms more indicative for the topic addressed in the document and thereby for the classification.

Additionally we show that for the task of document classification, associative networks outperform a TF-IDF [Salton and Buckley, 1988, Hiemstra, 2001] baseline and provide a qualitatively better document classification. In this experiment, we chose a TF-IDF

baseline as it is simple to implement (as explained in Section 7.2.5). In Chapter 11 we compare associative networks to the state of the art.

7.2.1 Design of the Experiment

We tested the use of associative networks without natural language processing (1), as well as a version with a part-of-speech tagger (3) and a version with full language parsing (5) and compared the results to each other and to parallel TF-IDF baselines without natural language processing (2), with a part-of-speech tagger (4) and with full language parsing (6), measuring the accuracy of the categorization.

In all cases, Princeton WordNet [Miller, 1995, Fellbaum, 1998] was used in Step 1 (extracting the bag of words) for creating the bag of lemmas for each document to be categorized. In WordNet, lemmas are represented by synsets, which are sets of synonymous words that have been grouped together, which motivates the requirement for semantic disambiguation examined in this experiment. Besides using WordNet to create the bag of lemmas, we also used it to construct the associative network used in this experiment. Chapter 5 describes the exact details of creating an associative network based on WordNet and also explains how other sources can be used for this purpose.

As said, we measured the performance of associative networks with two types of natural language processing (part-of-speech tagging and a natural language parser) and compared it to a version without any form of NLP. Each of these options was tested with associative networks and a TF-IDF baseline, for a total of six different settings. Note that settings 5 and 6 used both part-of-speech tagging for word disambiguation and the natural language parser (which also uses part-of-speech tagging to be able to parse the text).

1. A basic associative network without natural language processing
2. A TF-IDF baseline without natural language processing
3. An associative network with part-of-speech tagging
4. A TF-IDF baseline with part-of-speech tagging
5. An associative network with a natural language parser and part-of-speech tagging

6. A TF-IDF baseline with a natural language parser and part-of-speech tagging

The task we set for these systems was to sort 50 related Wikipedia articles into five predefined categories. The reason related articles were chosen was to more increase the difficulty of the categorization – if articles are very different between categories it would be too easy to classify them. Performance was evaluated by determining how well the article classifications matched the manual categorization made by the Wikipedia user base. The conversion of words in the text to lemmas was identical between the associative network and its TF-IDF counterpart in the matching conditions.

7.2.2 Creation of the Test Sets

To provide structure to its many articles, Wikipedia allows articles to be sorted into categories. Categories may contain not just articles but also other categories, thereby forming a hierarchy of categories and sub-categories. Many categories have a main article, often sharing a title with the category, which describes the basic concept that the category describes.

In each experiment, we selected a category (such as *'Animals'*) in Wikipedia and 5 subcategories of that category, similar to what we did for five categories in Chapter 6, though with different sub-categories. From each of the subcategories, a set of ten articles was randomly selected (filtering out articles with less than 500 words), for a test set of fifty articles in total. Additionally, the main article of each of the selected sub-categories was used to help classify the documents later on in the experiment, thus making a total of 55 articles. From each of these articles, a bag of words was extracted (see Chapter 2).

7.2.3 Set-up of the Experiment

First each individual article, as well as the individual main articles, was converted into separate bags of words (Step 1, see Chapter 2). To do this, a set of lemmas was extracted from the text of each of the Wikipedia articles in each test set. This was done three times for each article, once without natural language processing, once with part-of-speech tagging and once with natural language parsing, to match the three pairs of classifiers (TF-IDF

and associative network), resulting in a total of 825 sets of lemmas[2]. In each case, every lemma extracted from the article was given an activation value based on the frequency of corresponding words within the article. In the latter two cases (5 and 6) where the natural language parser was used, the frequency was then further modified by a factor based on the relevancy of each word in the sentence, based on the number of dependency relations (see Section 7.2.4) of the individual entries, to give additional weight to lemmas that are more important in the sentence.

For the TF-IDF baseline, the set of lemmas was then used directly to find the correct sub-category for the article, by comparing the TF-IDF values of each article against the TF-IDF values of the main articles. See Section 7.2.5 for details.

For the associative network method, the associative network did not compare the set-of-lemmas directly, but instead first generated an activation pattern for the set of lemmas, comparing it to the activation patterns of the five main articles of those sub-categories. To create this activation pattern, the set of lemmas was used as input for the trained associative network (Step 2, see Chapter 6). Association Concentration (Step 3, see Chapter 8) was then used to create an activation pattern for each of the 55 articles, repeated for each of the 5 categories and 3 different sets of lemmas, generating a total of 825 activation patterns. Next, the distances between the test and target articles were calculated (Step 4, see Chapter 3). Finally, categorization (Step 5, see Section 3.3.1) was done by sorting each test article into the category of the target article which it most closely matched: the associated sub-category was determined as the one in which the test article should be categorized.

As mentioned, three methods were used to construct the set of lemmas. They are described in the next section. Each of these sets of lemmas was used in a system that was made to determine which articles matched which sub-category, based on their textual content only. No other information, such as links in the Wikipedia articles, was used, and an accuracy score was established based on how many articles were sorted correctly, identical to the method used in Chapter 6.

[2] 50 articles per category, plus 5 main articles for the sub-categories, times 5 categories, times 3 extraction methods: (50+5)×5×3 = 825

7.2.4 Methods to Construct the Set of Lemmas

Here we describe the three methods for constructing the set of lemmas compared in our experiment, a basic associative network without natural language processing, an associative network with part-of-speech tagger and an associative network with a full natural language parser.

The set of lemmas, together with the weight, forms a bag of lemmas, which serves as input for the associative network (see Chapter 2), as described in Section 7.2.3 above.

No natural language processing

The text in each article was split into words, which were matched with the corresponding lemmas by comparing the words to surface forms of the lemmas. If a word form could be linked to multiple lemmas, it was not disambiguated, but all matching lemmas were used. Thus, for each article a set of lemmas was established, with a weight based on how many times each lemma was matched.

Associative network with part-of-speech tagger

Linking all words to lemmas based on their surface forms is not very precise, as explained earlier. To help correct for this and to reduce the impact of lexical ambiguity, the Stanford Part-Of-Speech Tagger [Toutanova et al., 2003] was used to get a better match between words and lemmas: using the default tagger, the part-of-speech of each word was established. For every word, the lemmas were filtered based on this type by eliminating lemmas that did not match the established part-of-speech. In the earlier example of *'I want to fly to Paris in a jet',* the word *fly* would be tagged as a verb, so it would not match the part-of-speech for the insect.

Note however that this method cannot remove all noise. The word *bat* can mean both the flying mammal and the sports implement used in baseball, but since both are nouns we cannot use part-of-speech tagging to determine which sense of the word is correct.

nsubj	(want-2, I-1)
nsubj	(fly-4, I-1)
root	(ROOT-0, want-2)
aux	(fly-4, to-3)
xcomp	(want-2, fly-4)
prep_to	(fly-4, Paris-6)
det	(jet-9, a-8)
prep_in	(fly-4, jet-9)

Table 7.1: Example Collapsed Typed Dependencies by the Stanford Natural Language Parser [Klein and Manning, 2003]

Associative network with natural language parser

Not every part of a sentence is equally important. In the sentence '*I want to fly to Paris in a jet'*, the word *fly* is more important than the word *jet*. We would like to know which words are the most important before activating the network; if words that have less relevance in the sentence are activated less, the associative network will be able to identify relevant lemmas more accurately in turn.

Using the Stanford Natural Language Parser [Klein and Manning, 2003], each sentence in the documents was tagged and parsed to determine the dependency relations between its words. In the example sentence eight connections are made (see Table 7.1), five of which include the word *fly*. This implies that the word *fly* is crucial in the sentence. Collins [1996] uses a similar metric for his parser, enumerating the dependency relations within a sentence as a central part of his statistic model.

Based on this idea that words connected to many other words in the sentence are more relevant to the sentence, the weight of each word's associated lemmas was increased with a factor of the total number of connections in the sentence to the number of connections involving that word. Thus, in the case where 5 out of 8 connections made include the word *fly*, the total activation of the word would be increased by 62.5% (5/8th).

7.2.5 TF-IDF Baseline

TF-IDF, or Term Frequency-Inverse Document Frequency, is a statistical method used to show how important a word is in a document in relationship to a library of documents.

In effect it is used to show a ratio of how common the word is in a document versus how common it is in the entire library of documents. If a word is common in a specific document but not in other documents, it will be more characteristic of that document than if the word was common in all documents.

To establish the baseline, we calculated the TF-IDF value based on the lemmas as determined by the three methods described earlier[3]. The resulting values were used to determine the distance between each test article and the five target articles using the following equation:

$$D(a_x, m_y) = \sum_{n=1}^{s} \left| tfidf(a_x, S_n) - tfidf(m_y, S_n) \right| \tag{7.1}$$

where $D(a_x,m_y)$ is the distance between the article a_x and the main article m of sub-category y. S is the set of lemmas 1...s in the corpus and $tfidf(a,S_j)$ is the term frequency-inverse document frequency of the lemma S_j in article a. Each test article was assigned the sub-category of the target article to which it had the shortest distance.

We use this form of TF-IDF as it most closely resembles the method used by associative networks to establish the distance between documents (see Section 3.3.1), and note that this method is not the most optimal form of TF-IDF. Additionally, as TF-IDF is an unsupervised learning method (basing the categorization purely on term counts), while associative networks are a supervised method (using training to optimize the weights in the network as described in Chapter 6), the baseline is likely to have a lower level of performance in each of the three cases than the associative networks.

7.3 Results

Table 7.2 lists the average results of our experiment for both the TF-IDF system and the associative network, over five sets of 50 documents. As expected, the associative network shows a consistently higher accuracy, with over twice as many articles from the test set sorted into the correct sub-category. Given the additional information regarding language

[3]In most existing work, TF-IDF is applied directly to the words in a document, but in the interest of a more accurate comparison, we chose to use the same bag of lemmas as was used for the associative network.

	TF-IDF accuracy	Associative Network accuracy
No natural language processing	34%	78%
Part-of-speech tagging	31%	84%
Natural language parsing	34%	87%

Table 7.2: Natural Language Processing results

information and the benefit of supervised learning, it is no surprise that the associative network approach outperformed the TF-IDF baselines.

What is more interesting is that the TF-IDF approach gained nothing from natural language processing and in the case of part-of-speech tagging the results actually worsened. We believe this is most likely the result of relevant words now activating fewer lemmas: in turn fewer lemmas match with the target articles – in the case of associative networks, the spread of activation through the network compensates for this.

By contrast, associative networks clearly benefit from the improved match between word forms and lemmas, gaining six percent-point with part-of-speech tagging and another three using the parser. This also shows that the improvement to the bag of word offers significant benefit ($p < 0.05$ based on a Student's t-test) over an unmodified bag of words.

7.4 Conclusions

In this experiment, we have examined the effect of two natural language processing techniques to improve the accuracy of associative networks when categorizing documents. We found that by using part-of-speech tagging we can eliminate incorrect surface forms, which improves accuracy. Additionally we found that we can indeed use natural language parsing to identify central words by using dependencies, which we can then use to give a higher activation to those central terms. In both cases, the bag of words used as input for the associative network is improved in quality, resulting in higher accuracy for the associative network overall. These results confirm our hypothesis that we can use natural language processing to improve the accuracy of associative networks by removing irrelevant entries from the bag of words.

As expected, the associative network outperforms a TF-IDF baseline, since TF-IDF is an unsupervised learning method, while associative networks are a supervised learning method. Moreover, our implementation of the TF-IDF baseline is arguably not the best possible version of this method. Regardless, the large difference between the two – associative networks consistently have an accuracy that is over twice as high as the baseline – shows that associative networks at the very least have the potential of becoming a competitive technique for document categorization. In other experiments, we set associative networks up against other supervised learning methods, showing that the semantic relations between concepts expressed in associative networks can help them outperform these as well (see Chapter 5 and Chapter 8).

Chapter 8

Experiment 4: Association Concentration

"The second rule is to concentrate our power as much as possible against that section where the chief blows are to be delivered" - Von Clausewitz [1832]

In this Chapter we describe our work on association concentration, a technique we created to use associative networks to determine which terms are related to a specific document. Figure 8.1 depicts where in the process of categorization described in Chapter 2 we apply association concentration.

A short version of the work described in this chapter was published as *'Hierarchical Document Categorization using Associative Networks'* in the proceedings of the 12th IASTED International Conference on Artificial Intelligence and Applications [Bloom et al., 2013a].

8.1 Concentrating Activation

Association concentration works by activating an associative network with a collection of words. This activation spreads through the network, and the words that are most related to the nodes used as input will receive the most activation.

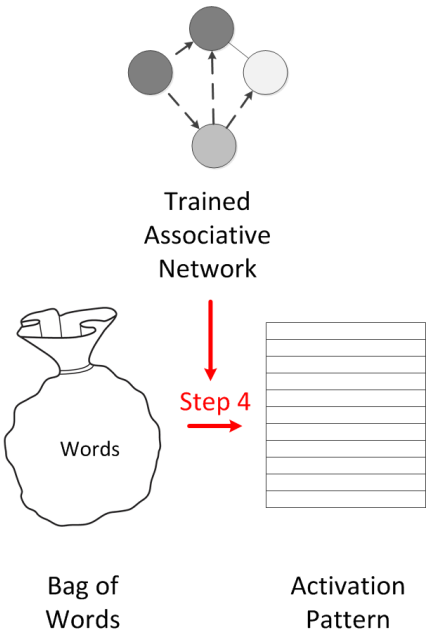

Figure 8.1: Association concentration in the categorization process

8.1.1 Basic Concept

As described earlier in Chapter 3, words from a text are used as the basic observations, in other words the bag of words used as input for the associative network. Words that are observed in the text automatically activate other nodes in the network if they are associated through their associative relationships, thus spreading activation throughout the associative network. This activation allows us to create a set of concepts that are related to the text, even though they are not in the document themselves.

High weight edges within an associative network spread activation more strongly than low weight edges and activation thus reaches more closely related terms more easily. As mentioned in Chapter 3, associative networks form a sparse graph and this property continues to hold as activation propagates further through the associative networks. Since words that are more closely related to the input will receive a high activation from multiple sources, the activation concentrates in words that are closely related to the document while those that are not very related receive only very little activation.

The associations found using this method are not always perfect. They allow us to make an educated guess about concepts related to the document, rather than providing absolute certainty. Thus the additional terms which are not in the input text but which are related to the document by association are inferred rather than deduced. However, the more words in the text we can link with a word, the higher the probability that this word is relevant to the document.

8.1.2 Flow versus Spread

When applying association concentration, the input from each document and the associations of that input form an association sub-graph which allows us to compare documents based on their conceptual content. The association sub-graph is a directed acyclic sub-graph of the associative network, consisting of the nodes activated by a specific document, and the edges between them. Rather than copying the edge weights of the associative network, the association sub-graph stores a weight for each node with its activation value. When comparing documents, we use those activation values while the association sub-graph itself is used for learning (see Chapter 6). The exact method by which the association sub-graph is constructed can vary. We use two different methods to construct it from the associative network, which we previously described in Section 3.2.2.

The first method is to use spreading activation (see Algorithm 3.1), which is the method we have used in most of our experiments. In this case, the sum of the weights of all outgoing edges for a node is equal to one. When a node is activated, it spreads the power by which it is activated amongst its outgoing edges according to these weights. Thus, if a node CAR has two outgoing edges, to RACING with weight 0.75 and to PAINT with weight 0.25, if the CAR node is activated with power 4, it will activate the RACING node with power 3 and the PAINT node with power 1.

Spreading activation sometimes overspreads through nodes that have very few edges, as the activation loses little or no power by spreading through these nodes. This means activation might spread too easily from one node to a distant one by virtue of a lack of neighbours in the nodes in between the two, thereby spreading activation to unrelated concepts. We also created an alternative method which compensates for this which we based on a flow

network, with some adjustments (see Algorithm 3.2). A flow network is a directed graph where each edge has a limited capacity and each node receives input from incoming edges called flow [Ahuja et al., 1993]. Flow originates from nodes called sources and is absorbed by nodes called sinks. The amount of flow going into a node must equal the amount of flow going out of the node.

We modified the principles of the flow network for associative networks, and such an 'associative flow network' acts in a similar manner, with an important difference: in regular flow networks there is a predefined sink, but in an associative flow network every node is a partial sink. This means every node absorbs a small amount of flow, thereby compensating for the problem of overspreading. During an operation, once a node has absorbed its sink amount, it is saturated, and from then on acts as a regular node in a flow network, passing all incoming flow through the pipes. Based on a certain input of flow, we calculate the flow pattern through the network and take all output nodes that receive flow, ordering them by the amount of flow received. The results are saved along with the flow paths between the input and output nodes, just as they are saved along the activation paths for the spreading activation method.

8.2 Description of the Experiment

In earlier experiments (see Chapters 7 and 5), we used associative networks to classify documents into predefined classes. In this experiment, we use the two methods of activating associative networks. In contrast to some of our other experiments, we categorize rather than merely classify documents: the system determines its own hierarchical structure of categories to order the documents instead of relying on predefined classes as we did before.

To test the effectiveness of our approach, we created two identical associative networks as described in Chapter 5, one using flow activation and one using spreading activation, both based on WordNet. These systems were then used to categorise a library of Wikipedia articles. Finally, those categorizations were compared to a baseline and a gold standard to determine which of them gave the most accurate and most useful results.

In all of our tests, we compare three different methods of categorization:

- an associative network based on spreading activation

- an associative network based on a flow activation
- and a neural network baseline (see Section 8.2.4)

8.2.1 Task

Given libraries of English Wikipedia articles, the aim was to create a categorization of the documents in those libraries. No information about the categories to be used was provided beforehand: the system had to create its own categories. However, the system was provided with a guideline for the number of documents a category should roughly contain: around 5 categories for small libraries and between 300 and 400 categories for large libraries (see below). No techniques for balancing the number of articles between categories were used [D'Alessio et al., 1998] nor were hierarchies adapted after construction [Li et al., 2007]. Categories are hierarchical but a document may only be sorted into one category.

8.2.2 Creation and Training

As we did in other experiments (See Chapters 7 and 5) we used Princeton WordNet [Miller, 1995, Fellbaum, 1998] to initialise our associative network, using lemmas as nodes. Edges between the lemmas in the associative network were made based on syno-/antonym, hyper-/hyponym, holo- /meronym, troponym and entailment relations and were assigned a default weight of 1 as there is no information in WordNet about the importance of the connection between the lemmas.

To link the concepts expressed in WordNet by synsets (sets of synonymous words describing the same concept, see Chapter 5) to the words in a document, each node was associated with an automatically generated list of the surface forms (plurals etc.) of each of the synonyms based on the English grammar rules. For example, the synset for the noun *fly* was linked to the singular surface form *fly* and plural form *flies*. The surface forms of the words in the synset were used in linking the synsets to the raw text of the input document. Words from the input text were not associated with just one lemma, but also with synonymous lemmas. No additional NLP techniques were used to improve these links, unlike in Chapter 7.

Training was done by creating a training library composed of 30 manually selected Wikipedia articles, with each article being closely related by topic to exactly one other article and not related to the remaining 28 articles. After initialization, the associative network was activated for each of the 30 articles in random order to determine which of the articles were the most closely related. Depending on the result, positive or negative reinforcement was applied to the network as described in Chapter 6. This cycle was repeated until the associative network produced the correct matching article for all 30 manually selected articles.

8.2.3 Categorization Process

To categorize a library of documents, following the process sketched in Chapter 2 we started by scanning the text of each document, removing meta-data to acquire the raw text of the document. A list of lemmas corresponding to the words in the document was then generated from the raw text by matching surface forms. If multiple lemmas shared the same surface form, they were all activated. Thus, the word *fast* would activate lemmas for abstaining from food as well as high speeds. The associative network filters out the correct lemma by the activation spread – lemmas that are not related to the rest of the text automatically get less spread from the rest of the input.

The lemmas of each document were then used by the associative network to construct an association sub-graph for each document. Next, these association sub-graphs were compared to determine the distances between documents. This distance was determined based on the total activation value that each node received after association for each document, following Equation 3.2 on page 45.

Finally, a clustering algorithm based on multi-level graph partitioning [Karypis and Kumar, 1998] was used to identify subsets of documents that were closely related to one another based on the distance between them. Each cluster became a category, together forming a hierarchy. The same clustering algorithm was used for all of the methods we tested. Alternative clustering algorithms, such as the divisive method of [Zhong et al., 2011] may allow more diversely shaped clusters of documents to be found more easily,

while different methods of merging or splitting clusters during the construction of the hierarchy can significantly affect the final hierarchy [Ding and He, 2002]. Exploring these alternatives might be interesting future work but falls outside of the scope of this experiment.

To determine the name of the category, the association sub-graphs in each cluster were analysed to find the common denominator between the documents, which is the node that received the highest input value after association. As this is a concept which received high activation by all documents in the cluster, it was used as a name for the category.

8.2.4 Baseline and Gold Standard

In an earlier experiment we already established that associative networks outperform a TF-IDF baseline (see Chapter 7), so in this experiment we chose to compare our methods to neural networks. We chose neural networks as a baseline because of their structural similarity to associative networks and the shared learning method of back-propagation (see also Chapter 3). Using earlier work by Jeschke and Lalmas [2002] and by Chen et al. [2005] on hierarchical document classification as inspiration, we created the baseline by taking a total of twenty large scale neural networks of different sizes that were constructed and trained analogously with the associative networks. The network that performed best after training was used in the experiment.

As the golden standard in the automatic evaluation, we used the categorization made manually by the Wikipedia authors. In Wikipedia, articles can be in more than one category, but for our experiment we removed all categories other than the one from which the articles were selected (see below) to leave each article in a single category.

8.2.5 Small and Large Libraries

Two types of test libraries were generated: small libraries with a small number of articles for evaluation by humans and large libraries with a much larger number of articles, which were only evaluated automatically, due to their size.

Libraries were generated from a random selection of English Wikipedia articles. Articles were selected from different, related subcategories in Wikipedia. The subcategories

themselves were selected by first randomly selecting a primary category and then selecting random subcategories recursively to ensure a hierarchical structure. Stub-articles, lists and disambiguation pages, as well as articles with fewer than 1000 words were excluded from the test. Articles (documents) were stripped of all meta-information such as links and categorization and were converted to raw text.

Sixteen small libraries of articles were constructed, with a total of 290 texts, an average of 18 per library. Sixteen large libraries were also constructed, with between 10.000 and 15.000 articles each. Both sets of libraries were then passed to each of the three algorithms (spreading activation, flow activation and neural network based) to be categorized.

8.2.6 Evaluation Method

Two forms of evaluation were used: automatic evaluation and manual evaluation. For the automatic evaluation, a *human-likeness score* was generated by an automatic comparison of the resulting categorization with the Wikipedia user base categorization. This score, based on methods for comparing trees [Zhang and Shasha, 1989], was calculated by taking the number of elementary transformations (insert, delete, modify and rename) necessary to morph the result categorization into the Wikipedia user base categorization.

Based on the human-likeness score, the four methods were ordered from 1 (best) to four (worst). Due to their size, large libraries were evaluated only by means of the human-likeness score; there was no evaluation by human judges.

Besides automatic evaluation which was carried out for both small and large libraries. A *human evaluation* was also carried out on the small libraries, since the quality of categorization is often subjective [Sebastiani, 2005]. The human evaluation was only carried out for the small libraries because having humans evaluate 10.000 to 15.000 articles would be impractical.

The categorizations of the small libraries were evaluated by 37 human judges in two ways: on paper (by 12 judges) or via the Internet (by 25 judges). The judges evaluated the categorizations on two criteria: correctness and usefulness. Correctness was defined to the judges as a combination of articles being in the correct category and those categories being correctly named. For example a category *flowers* has no business being a sub-category of

vehicles and a document *swimming techniques of aquatic mammals* does not belong in the category *air-planes*. Usefulness was defined to the judges as the quality of the hierarchy of categories with regards to ordering the articles in groups as well as the overview the hierarchy provided of the information. Simply declaring a category *universal* and sorting everything in it is accurate but not useful. Likewise, giving each document its own, independent category is not useful. A useful categorization creates sub-categories of roughly similar size, does not subdivide sub-categories that are already very small and does not nest sub-categories too deeply. 'Balanced' might also be used as an alternative term to useful in this instance. The composition of the hierarchy should moreover make the information easy to find.

The group of judges consisted of men and women from age 20 to 60 with educations ranging from high school level to university educated and diverse backgrounds such as Linguistics, Computer Science and Medicine. Each judge was given the categorizations for a random library and was asked to provide a ranking from one (best) to four (worst) of the three automatic categorizations (neural networks, associative network based on flow activation and associative network based on spreading activation) as well as the original Wikipedia categorization. The average position was then calculated by adding all positions and dividing by the number of judges. An absolute ordering rather than a scoring system (such as each judge assigning a score of 1 to 10 to the two criteria for each categorization) was used to make the results easier to compare between different judges. The judges were not told which method generated which categorization and the author was not present during the test to ensure double-blindness.

Instructions were given beforehand with a simplified example and judges were asked to review multiple libraries. Each library was reviewed by two judges in paper format and by at least two judges online. No library was reviewed more than once by the same judge. For the off-line evaluation, the results were discussed informally afterwards to get some idea of the reason why certain categorizations were considered better (see below).

The method of ranking each categorization by usefulness and by correctness was selected based on the subjectivity of the problem and the underlying goal of categorizing documents to allow easier access to new users (see Section 1.3.2). New users would generally have only limited knowledge of the topic but would still wish to find information

	Wiki	Flow	Spread	Neural Network
Correctness	1.5	2.5	2.9	3.1
Usefulness	1.9	2.1	2.7	3.3

Table 8.1: Correctness and Usefulness, average over 16 small libraries (Manual) – lower is better

as fast and intuitively as possible. To simulate this limited knowledge in our test, libraries were assigned to judges randomly.

8.3 Results

In this section we describe the results of our experiment and offer some discussion as to improvements that might be made to the experiment itself.

8.3.1 Correctness and Usefulness

In Tables 8.1, the averages of the outcomes of the human evaluation are shown. The column labelled *Wiki* is the gold standard categorization made by human Wikipedia authors. The *Flow* and *Spread* columns list the results for the two types of associative networks while the column marked *Neural Network* represents the baseline results.

The associative networks had a lower correctness than the gold standard of the Wiki categorizations – many judges informed the author after their reviews that they found small errors in the categorization that caused them to rate the networks lower on this measure. In several cases the subdivision was considered good, but the name assigned to the category did not reflect the content, but regardless of this, neural networks were outperformed in all cases. Flow-based associative networks produced better results than spread-based associative networks on average.

In regards to usefulness, associative networks based on flow were especially successful, getting close to the gold standard Wiki categorization. As with correctness, flow-based associative networks outperformed spread-based associative networks, and both types of associative networks outperformed neural networks.

	Wiki	Flow	Spread	Neural Network
Small Libraries	0	2.0	1.7	2.4
Large Libraries	0	1.9	1.8	2.3

Table 8.2: Distance to Wikipedia categorization, average over 16 libraries (Automatic) – lower is better

8.3.2 Small and Large Libraries

In Table 8.2, we automatically calculate the distance between the gold standard of the Wikipedia categorization and the various categorization methods. Comparing the results of small libraries to those of large ones, we can see that the distance to the human categorization is fairly similar. Based on this we expect that if the large libraries were evaluated by humans, the results on correctness and usefulness would also be similar to those of the small libraries.

8.3.3 Discussion

Several aspects could be improved in the experiment: the general goal of the categorization as mentioned in the introduction to the judges is the ordering of the information, but various judges stated a different task, such as searching for the answer to a specific question, might have influenced their decision.

Correctly naming the categories using the associative network is a topic that could benefit from further research; see for example, [Fukumoto and Suzuki, 2011], and in later experiments of this nature it would likely be better to split correctness and usefulness to more specific qualifications.

8.4 Conclusions

In our evaluation experiments, the associative networks were found to perform consistently better than the baseline neural network, with the new flow-network-based associative networks being an improvement on spreading activation based associative networks and at the

very least producing similar quality results. We estimate that the reason flow-networks perform better is that they keep association concentration more confined within the network, resolving the before-mentioned overspreading problem (see Section 8.1.2).

Furthermore, human judges rated the categorizations created through associative networks, especially the Flow Activation variant, close to the gold standard in terms of usefulness and higher than the baseline in terms of correctness, providing some validation to our hypothesis that associative networks may be used to categorize large libraries of documents, as described in Section 1.3.2.

As the results produced by associative networks produce similar average distances to the categorization of Wikipedia for both small and large libraries, we expect that the level of correctness and usefulness which was examined only for small libraries would continue to hold for large libraries. Compared to the amount of time it took to create the categorization of Wikipedia, being able to make these large scale categorizations automatically is a significant advantage.

Chapter 9

Experiment 5: Power Graph Analysis

> *"The attempt to combine wisdom and power has only rarely been successful, and then only for a short while."* - Albert Einstein, as quoted by Calaprice and Einstein [2005]

In order to improve the results of our categorizations and to better identify categories within a set of documents, we have extended power graph analysis (a technique from bioinformatics that is used to analyse protein network) to cover weighted graphs, which more closely matches our usage scenario.

In this section we describe how we use Power Graph Analysis for categorization and what improvements this provides. In general, we are able to use power graph analysis to help identify key document groups that should be categorized together. Figure 9.1 depicts where in the process the categorization applies. We additionally describe how Power Graph Analysis is useful for understanding associative networks, by providing insight into the structure of the associative network.

We end the chapter with a description of a quick scan method we developed based on power graph analysis, which can be used to informally analyse associative networks.

A short version of the work described in Sections 9.1 to 9.4 of this chapter was published as *'Applying Power Graph Analysis to Weighted Graphs'* in the proceedings of the 34th European conference on Advances in Information Retrieval [Bloom, 2012a].

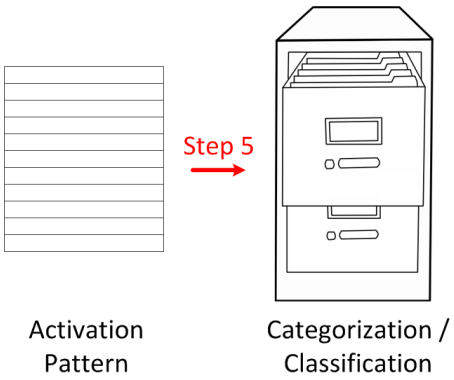

Figure 9.1: Power Graph Analysis in the categorization process

9.1 Power Graphs

Power Graphs are abstractions of unweighted undirected graphs that mark star, clique and bi-clique motifs in the graph (see Figure 9.2). These patterns are represented using power nodes, which are sets of nodes grouped together, and power edges, which signify relations of these sets with individual nodes and with other power nodes.

9.1.1 Power Graph Analysis

Power graph analysis has been used to help analyse and understand protein networks [Royer et al., 2008], specifically to gain insight into the biological relationships between proteins based on the similarities of their interactions. Beyond analysing protein networks, power graph analysis can also be used with associative networks to help with categorization.

Using associative networks, we make a comparison between the activation patterns of documents in a library to help us create a categorization. Based on that comparison, we can see which documents are closely related and which ones are not. Modelling the distances between documents as a graph can give us a very similar structure to those of protein networks. By using power graph analysis, we hope to be able to reveal aspects of the underlying structure of the network of documents, just as it did for the underlying biological structure in case of protein networks. This structure can then be used to help

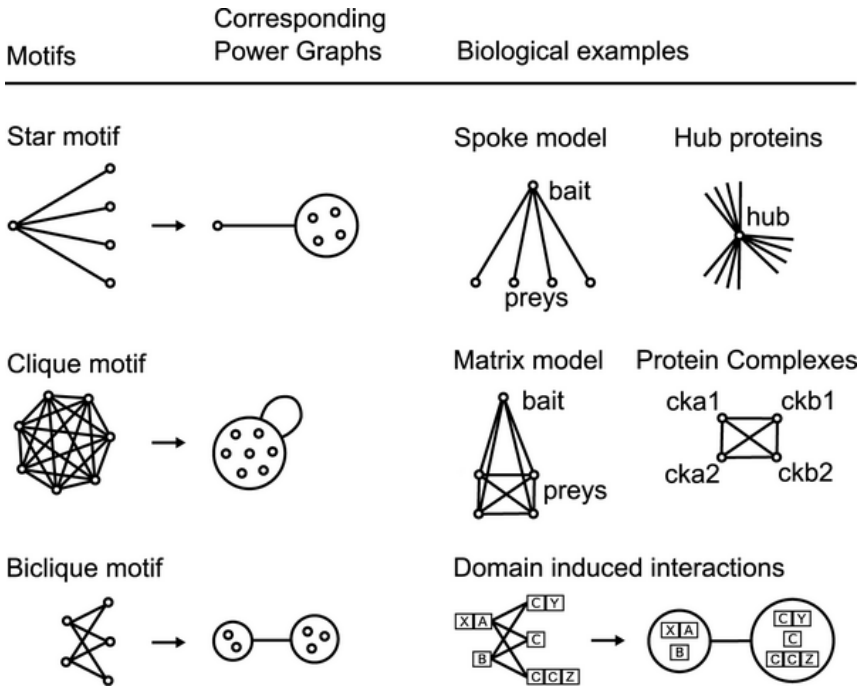

Figure 9.2: Power Graph Analysis - *image by* Royer et al. [2008]

create a categorization as it represents the way documents relate to one another, which is what a categorization should model.

Moreover, the technique can be applied to associative networks themselves: the words in associative networks can have similar roles as the proteins do in protein networks. For example, two words may represent similar concepts, may both be connections between sets of words or may together represent a specific, grander concept. Because of this similarity, we wish to see if power graph analysis can reveal aspects of the underlying structure of the associative network as well.

9.1.2 Extending Power Graphs

One limitation of Power Graph Analysis is that it only applies to unweighted graphs. Both the categorization and the associative network are weighted, with potentially large differences in weights. As edges in weighted graphs contain this relevant information, our weighted graphs are not immediately obvious candidates for power graph analysis: edges

can only be merged into a single power edge without loss of that information if they have exactly identical weights.

Fortunately, despite this loss, it is still possible to use power graph analysis to gain insight into the underlying relationships and find groups and clusters of related documents.

When creating a power graph from a weighted graph, rather than judging each power node only by the number of edges removed, we use the weights to determine which power node is actually the best candidate, its value determined by the total weight of the edges removed. This means that strongly correlating nodes (with a high weight on the edge between them) are more likely to be grouped into power nodes together.

9.1.3 Related Work

Royer et al. [2008] found power graphs to reveal aspects of the underlying biology by simplifying the representation of the data without loss of information. Their results are in line with other motif finding algorithms that perform similar functions [Andreopoulos et al., 2007, Bader and Hogue, 2003].

The protein networks used by Royer et al. represent interactions between proteins in various biological processes, in which some proteins perform similar functions, some make key connections between other proteins and some proteins act as a group, together serving a specific purpose.

Andreopoulos et al. [2007], who also works on protein networks, created an algorithm to find neighbourhoods, which are clusters of proteins with similar interaction patterns, and mediators, which are proteins that enable other clusters of proteins. In terms of graphs, Andreopoulos et al. build up clusters incrementally by their level of similarity in terms of edges in the graph. Bader and Hogue [2003] use a different algorithm that relies on the number of connections of a node and its neighbours to build up clusters efficiently. Both methods results in slightly different clusterings than Power Graph Analysis, yet the effective patterns discovered are similar for all three. All three methods operate on unweighted graphs. Bader and Hogue additionally offer support for directed graphs, but none of the three algorithms operate on weighted graphs.

The construction of power graphs from regular graphs is an NP-complete problem: it encompasses the maximum edge bi-clique problem, which was established as NP-complete by Peeters [2003]. Because of this, Royer et al. use a series of heuristics to establish candidate power nodes - sets of nodes that could potentially become power nodes. They then select the actual power node by finding the candidate with the highest edge reduction, which is a measure for the number of edges replaced by power edges.

9.2 Description of the Experiment

In this experiment we test whether the use of power graph analysis can improve categorization. When Royer et al. [2008] did their work on protein networks, they used statistical analysis to prove that power graph analysis revealed aspects of the underlying biology in their data. We use the same statistical analysis to prove that power graph analysis reveals aspects of the underlying structure in our data.

As our method uses power graph analysis based on weighted graphs rather than unweighted ones, we expect that our total edge reduction (the measure used by Royer et al.) will be worse than Royer et al.'s. Power nodes with a high edge reduction are still likely to have a high combined weight, so making power nodes based on the highest combined edge weight should still allow for a high edge reduction, but sometimes creating power nodes based on the highest combined edge weight will result in a few high weight edges being removed instead of many low weight edges. Simply put, we expect that the total edge reduction will be lower than for unweighted graphs because candidates with high correlation will be preferred over candidates with more, lower valued edges. However, using the same hypothesis as Royer et al., we still expect to see the total edge reduction remain above the average of random graphs.

9.2.1 Method

Royer et al. evaluate their results on various protein networks by comparing them to randomly generated networks of the same size and edge density. Their hypothesis is that power graphs will have a lower edge reduction for randomly wired graphs than they would for the

graphs with real data, as good power edges should be more easily created based on the underlying structure, and we use the same method and reasoning for our experiment.

To compare to their work, we used Royer et al.'s method of establishing the random baseline by means of 1000 randomly rewired networks of the same size and edge density to estimate the variance of the edge reduction and establish a z-score. The z-score or 'standard score' is the number of standard deviations an observation is above or below the mean, which thus allows us to estimate if it is indeed the case that the edge reduction is above average for the real data. Additionally, we wanted to compare the use of weighted graphs to unweighted graphs, so we processed each generated dataset both in its original form and with all weights stripped.

9.2.2 Dataset

To construct a test set, we took a corpus of thirty related articles from a single category in Wikipedia and converted these to plain text. Only articles with more than 1000 words were selected. For each article, we calculated the TF-IDF values [Salton and Buckley, 1988, Hiemstra, 2001] of all words in the corpus. We then used the sum of the difference of these values to determine the relationship between each pair of articles, as done earlier in Chapter 7.2.5:

$$D(d_x, d_y) = \sum_{n=1}^{w} \left| tfidf(d_x, W_n) - tfidf(d_y, W_n) \right| \tag{9.1}$$

where $D(d_x, d_y)$ is the distance between the documents d_x and d_y, W is the set of words 1...w in the corpus and $tfidf(d_i, W_j)$ is the term frequency / inverse document frequency of the word W_j in the document d_i.

We established this relationship between all pairs of documents, removing all but the top 100 links between articles to keep only relevant relations. This process was repeated five times with different sets of articles, each time resulting in a different graph of 30 nodes and 100 edges with different weights.

	Edge Reduction	Conversion Rate	z-score e.r.
Weighted graph avg. performance	89.8%	12.319	5.764
Unweighted graph avg. performance	88.2%	11.437	4.910
Random rewire baseline	84.9%	6.208	-
Royer et al. data sets best	85%	13	242.7
Royer et al. data sets average	55.8%	6.0	43.4
Royer et al. data sets worst	38%	3.8	2.2

Table 9.1: Power Graph Analysis results

9.3 Results

Table 9.1 lists the edge reduction and conversion rates (a ratio between the edge reduction and the number of power nodes, indicating the average reduction per created power node), which were calculated in the same way as Royer et al. [2008] to allow for easier comparison. The z-score (see above) was calculated by comparing the datasets to the randomly rewired baseline samples.

At almost 90%, the edge reduction is higher than even the best datasets of Royer et al., both for weighted and unweighted graphs, with matching high conversion rates around 12. The corresponding z-score is low compared to Royer et al. Notably all scores for the unweighted graphs are lower than for the weighted graphs and both are higher than the random baseline.

9.4 Conclusions of the Experiment

Power graph analysis established z-scores that suggest the performance on actual document networks is significantly higher than performance on random networks, just as it did for protein networks in the work by Royer et al. [2008]. Comparing our results to their result on thirteen Protein Interaction Networks, the edge reduction and conversion rates for our data were high, some even above all of Royer et al.'s networks. This suggests that power graph analysis indeed picks up on the underlying structure of the categorization data, just as it did for protein networks, identifying meaningful connections and groups. Thus, though our sample size is small, our results suggest that like protein networks, document categorisation

may indeed benefit from using power graph analysis as it can uncover hidden information about the structure of networks of documents which we can use to make a categorization.

The worry that using weighted graphs might harm the total edge reduction appears to be unfounded, with the weighted versions of the graphs actually performing slightly better than the unweighted versions, lending further credence to the hypothesis that power graphs identify underlying patterns in the data, which we believe is the cause of this increase in performance: using the weights, power graph analysis identifies the key patterns more easily and the results benefit from this.

One note must be made regarding the high conversion rate and edge reduction: it is likely to be at least in part a result of the slightly smaller network size of 30 nodes and 100 edges compared to the protein networks, which can have up to a few thousand nodes and edges. Smaller networks mean it is more likely to find power nodes with a high edge reduction. Thus, it would not be justified to conclude that the high results imply that power graph analysis is more suited for our problem than it is for unravelling protein network as was done by Royer et al. This is furthermore confirmed by the fact that the z-scores of our results are relatively low compared to Royer et al. This too is to be expected with the smaller network size, as very large deviations from the random baseline are less likely with smaller networks.

9.5 Quick Scan of the Associative Network

Besides using power graph analysis for finding document clusters, in practice we can also use power graph analysis to make a quick scan to examine associative networks themselves.

Associative networks can be created from different sources and trained using different methods and training sets. During our work, we found that it is sometimes desirable to be able to examine the generated associative network manually, and determine if the values and relationships stored within the network are conceptually sound. The goal of such an examination is not to verify the validity of the entire network, but rather to take one or two samples to get a generic impression of the quality and to decide if it can be expected that a source or training method is producing sensible results. We call this a quick scan for associative networks.

One problem with such a quick scan is that associative networks are generally rather large, with hundreds of thousands of nodes and connections (see Chapter 3). Because of its size, it is not practical to simply display a graph of the entire network. Moreover, studying individual nodes and their relations is relatively difficult without context and when adding more nodes to provide that context, the displayed part of the graph becomes more difficult to inspect due to the increase in nodes and connections. Thus, we want a way to provide a simplified perspective.

As mentioned above, power graph analysis has been used to help analyse and understand protein networks by providing insight into the biological relationships between proteins [Royer et al., 2008]. Like associative networks, these protein networks can consist of many different nodes (representing proteins instead of words), some of which may fill similar functions. Power graph analysis can groups the proteins that perform similar functions and roles in a protein network by grouping proteins with similar interactions together. Based on this, we presume that power graph analysis might likewise be able to highlight groups of words that perform similar functions and roles in an associative network.

In Figure 9.3, we show an example of a limited selection of connections in an associative network, gathered from an associative network based on WordNet (see Chapter 5). As this example has been simplified by removing weights as well as many connections and nodes which we believed to obscure our example, most people would immediately see that several terms depicted are conceptually similar to one another and could be grouped together. In the original associative network, all additional nodes and connections obscure the conceptual relations and make it much harder to assess the quality. This holds especially for the weights, as certain connections in the network might be very weak and therefore not very applicable, while others are very strong and therefore highly relevant.

Power graph analysis can help us show the underlying relations between concepts within the associative network, as they simplify the structure by capturing nodes together in the same way that power graph analysis does for protein networks, and taking into account the weights of relations between concepts, thanks to the extension we made to allow power graph analysis to deal with weighted graphs (see Chapter 9). In Figure 9.4, we see an example of the power nodes and edges discovered by power graph analysis, which indeed highlights and groups similar terms together. The reduction in the number of nodes and

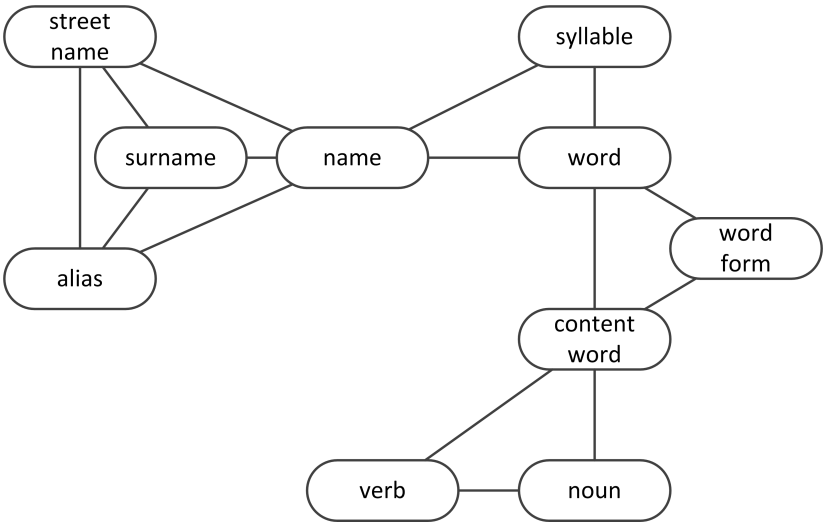

Figure 9.3: Connections in an associative network

edges makes it a lot easier to read at a single glance and the benefits grow as the network which is being evaluated is more complicated.

In our work, we have used power graph analysis as a generic test to make a quick evaluation during the development of various types of associative networks. Power graph analysis applied directly to the associative network has allowed us to informally validate the methods used to create or train associative networks at a conceptual level. This type of validation would have been impossible if we had not expanded power graph analysis to also cover weighted graphs. This evaluation served only as a quick scan, we did not used power graph analysis for a detailed validation of associative networks, leaving this as future work.

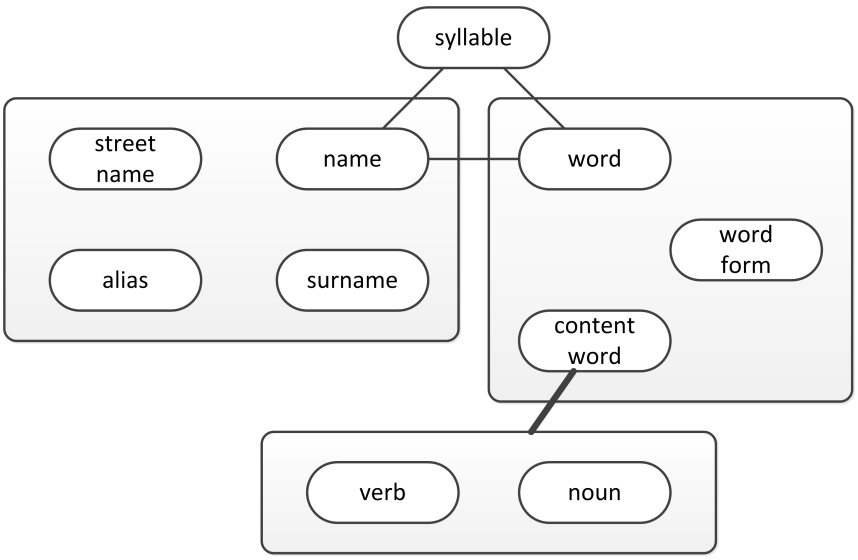

Figure 9.4: Power Graph Analysis on connections in an associative network

Chapter 10

Experiment 6: Multilingual Networks

> *"You have not experienced Shakespeare until you have read him in the original Klingon."* - Chancellor Gorkon [2293]

So far we have presumed that networks deal with documents all written in the same language, but there is no reason why this needs to be the case. As it turns out, using multilingual networks can actually be beneficial not just to allow the associative network to cover documents in multiple languages but in fact to get better results in general. Thus, using associative networks in this manner does not just allow articles in different languages to be classified without translation, but it also allows the connections in each language to represent subtle differences in meaning more accurately.

Figure 10.1 depicts which parts of the process are affected when changing from a monolingual to a multilingual set-up, starting with the documents themselves and the extraction of lemmas or words but also affecting the creation and structure of the associative network and the way association concentration spreads through the network. The exact details on how each of these steps is affected are described below.

A short version of the work was published as '*Document Categorization using Multilingual Associative Networks based on Wikipedia'* in the proceedings of the 1st International Workshop on Multilingual Web Access [Bloom et al., 2015].

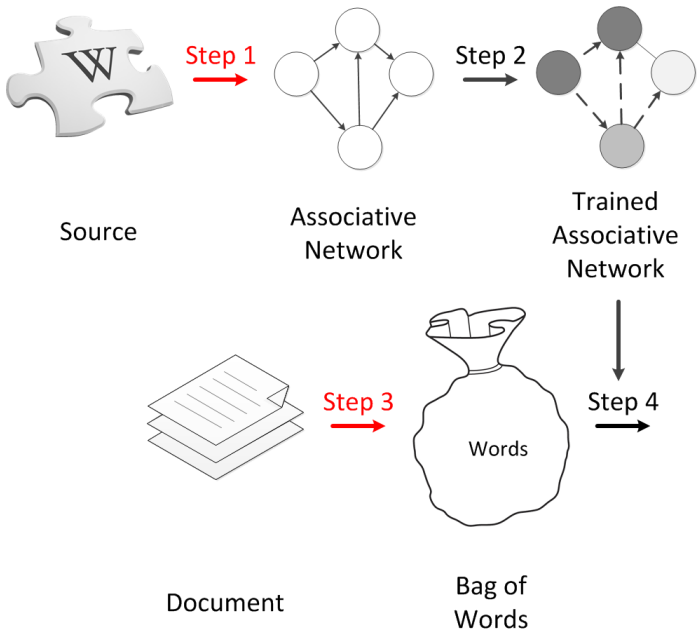

Figure 10.1: Multilingual networks in the categorization process

10.1 Multilingual Associative Networks

As we explained in Chapter 3, associative networks model relations between concepts that are linked in various ways. They could be linked because the concept represented by one word is a part of the concept represented by another such as CAR and WHEEL, because the concept represented by one word is a sub-category of the concept represented by another such as RAVEN and BIRD or because the concept represented by one word is affiliated with the concept represented by another such as CHRISTMAS and ADVENT. However the central relation, which we used to create our simplest associative networks (see Chapter 5) is synonymy, such as between the words *liberty* and *freedom*, which represent the same basic concept.

10.1.1 Simple Translation

In English, many synonyms emerged in the Middle Ages after the French-speaking Normans invaded and conquered England, the languages of the Norman ruling class and the Anglo-Saxon lower class mixing up over time [Algeo and Pyles, 2009]. The words *liberty*

and *freedom* are a perfect example of this. Since associations between such words can be used in an associative network, perhaps it is also possible to use translations of words from different languages.

Based on this idea we might expand associative networks to include multiple languages by counting both the original and the translated version of each word as one entry in the bag of words. We could thus activate nodes when either their original word or the translated version is present in a document.

However, this simplistic approach presents a problem: words often do not have an absolute translation, but rather the translations are approximations that carry a slightly different meaning [Nes et al., 2010]. That meaning, sometimes described as the undertone of the word, can be quite difficult to translate [Rohde, 2011], and can force a different impression from the original meaning which is lost in the translation. For example, Nes et al. describe how the Dutch word *wandelen* can be translated as *walking*, but in turn the word *walking* is more accurately translated to the Dutch word *lopen* which is the act of walking while *wandelen* is generally considered to be walking for the purpose of enjoying the act. A more accurate translation of *wandelen* might be *going for a walk*, which is a bit wordier but describes the act more accurately. This shows that the word *walk* when translated from English to Dutch can have different meanings.

Some words do not even have an equivalent in the other language. The Dutch word *gezellig* describing the feeling of comfort and doing things together often in the own home might be awkwardly translated with *cosy*, but the latter word loses a lot of meaning compared to the original such that they really cannot be said to cover the same concept. This problem does not just exist for capturing the relation between texts in Dutch and English. The German word *Schadenfreude*, the feeling of joy or pleasure at seeing others fail or suffer misfortune, has been adopted into the English language as a loan-word as it held no common English equivalent [Harper, 2013]. Likewise, the word *Fahrvergnügen* (which expresses the joy of driving) was borrowed from German to play a central role in the 1990 U.S. Ad campaign by German car-manufacturer Volkswagen [Stanfel, 2000].

These different undertones and meanings, combined with the fact that some words are completely missing in one language or the other mean we need a more complicated model for including a second language into an associative network.

10.1.2 Combining Associative Networks

We know that a monolingual associative network already captures many of the undertones of words in the links between the words and the weights of these links. Coming back to our earlier example, the Dutch words *wandelen* and *lopen* will both be linked to *rennen*, as both the idea of *going for a walk* and of *walking* can be related to *running*. However, the link between *wandelen* and *rennen* will be more distant than the link between *lopen* and *rennen* as *going for a walk* and *running* are less closely related than *walking* and *running*; the latter case is simply speeding up the motion, while the former both refer to ambulation but would be used in different contexts. As monolingual networks can already capture these subtleties, it might be sufficient to make multiple monolingual networks, one for each language in the corpus.

This method would certainly work when examining two independent libraries that are in different languages: each network could categorize its own language library. However, such a situation does not allow for finding groups of documents that cover the same topic in different languages, which is the task we are trying to carry out. If we have two separate associative networks, they will have no links to one another.

To resolve this final obstacle, we want to create links between the two associative networks. This appears at first glance to bring us back to the original problem that terms cannot be translated one-to-one, but even though that remains true, the other associative network can provide additional context to the translated term. Instead of treating these translations as special links between two associative networks, we could treat the combined networks as a single associative network, which can be trained as a whole, rather than as two separate networks with some links between them. This also allows us to cover words with multiple possible translations – we simply link the word in one language to each of the translations, using training to set the appropriate weights between them.

In Figure 10.2 we display an example of two small associative networks, in this case an English and a Dutch one, being combined into a single associative network. In the first step we have two separate associative networks. In the second step, links are added between translated terms. In the final step, a new, larger associative network has been created that

combines both networks, and which is ready to be trained as a whole on a multilingual dataset.

Multilingual associative networks are a potential improvement in the capabilities of associative networks, allowing them to categorize documents written in multiple languages. However because of the way association concentration works (see Chapter 8), adding additional connections to nodes in the form of links between translations means the spread from one concept to a related concept becomes weaker. After all, the activation of the node now has to spread over more paths than before. Thus, it would seem at first glance that the quality of the categorizations made by the associative networks should decrease, seeing as how less activation is spread to closely related concepts than before. However, this is not necessarily the case. Having additional links through translated versions of words may lead to more feedback from a node to a related node. For example, in the original Dutch associative network in Figure 10.2 there is only one path from *wandelen* to *lopen*: the direct connection. However in the Dutch-English associative network, there are three paths: the direct connection, a connection through *walk* and a connection through *walk*, *run*, and *rennen*. This means that part of the activation going from *wandelen* to *walk* will loop back to *lopen*, which in turn means more activation spreads from *wandelen* to *lopen*.

10.1.3 Related Work

In Chapter 5, we explained a method of using Wikipedia to create an associative network. Besides providing a wealth of information and articles on a variety of topics, Wikipedia offers the additional feature that its articles are internally linked to different language versions of those articles. Ni et al. [2011] use Wikipedia's multilingual links to extract relationships between terms in different languages for multilingual text classification. Primarily they extract relevant concepts in texts and translate these concepts through Wikipedia. We use Wikipedia's links between articles in different languages for similar purposes, effectively constructing an associative network in each language and using the relations between articles in different languages in Wikipedia to connect the two.

It should be noted that our approach - combining two monolingual associative networks to create a larger bilingual one, is different from methods such as those used by Lee and

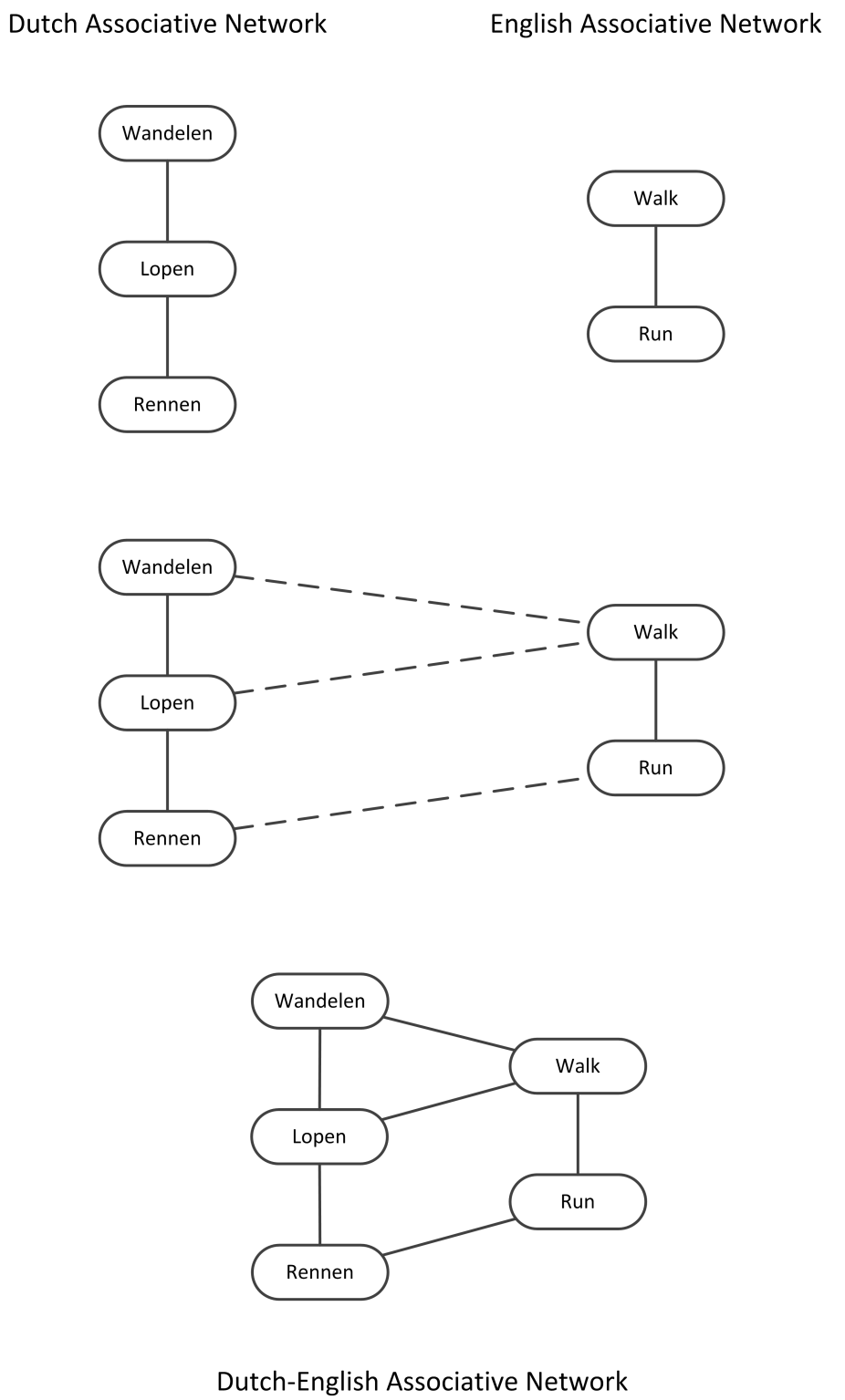

Figure 10.2: Combining an English and Dutch associative network

Yang [2009], who also create a concept-based multilingual classifier, but based on a single concept with forms in multiple languages. Such methods would not be able to capture the subtleties within different languages described above, as they are rooted in the assumption of a strong form of synonymy: a singular concept expressed by the corresponding words in all languages. Our method also differs from an ontology based approach such as used by de Melo and Siersdorfer [2007], who use ontologies expressing more information than associative networks, which only establish how closely two concepts are related.

Bel et al. [2003] were amongst the early pioneers examining cross-lingual text categorization. They used the Rocchio algorithm, a popular learning method based on relevance feedback, and the Winnow algorithm, a method for learning a linear classifier from labelled examples, to categorize documents in multiple languages. We compared associative networks to Rocchio amongst other classification algorithms in Chapter 5, and found associative networks performed significantly better than Rocchio. Rigutini et al. [2005] discussed extending automatic classification systems to include multiple languages, basing their work around the EM (Expected Maximization) algorithm, an iterative method for finding the maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. In their work, Rigutini et al. rely on translating the training data, a step that is not necessary with associative networks.

10.2 Description of the Experiment

In this experiment, we examine whether a multilingual associative network produces better or worse results in terms of accuracy than monolingual associative networks, which will also give us some indication of the effect of the aforementioned additional connections on the quality of the results.

We created three Wikipedia-based associative networks (using the method described in Chapter 5): one Dutch language associative network, one English language associative network and one multilingual Dutch and English associative network, as described in the previous section.

We constructed five datasets analogous to the experiments in Chapter 7 and Chapter 6, that is, datasets consisting of articles in five random subcategories of a manually selected

	Dutch A.N.	Dutch-English A.N.	English A.N.
Dutch Dataset Accuracy	88%	91%	-
English Dataset Accuracy	-	86%	81%

Table 10.1: Average results for the different associative networks

main category. However, to make our dataset multilingual, the datasets consisted of one hundred articles, rather than fifty, with fifty of those articles being in English and the other fifty in Dutch. It should be noted that the five subcategories within each pair were the same (or as similar as possible based on the available data) between each pairs of datasets. For example, the Dutch dataset might have articles in the subcategory *'Engelse Koningen'*[1] while the English dataset would have articles in the subcategory *'English Monarchs'*.

As in earlier experiments, the articles were matched to the five possible subcategories by comparing them with the Dutch main article for Dutch language articles and the English main article for English language articles for each of these subcategories, the category of the best main matching article being chosen as the correct sub-category.

As a Dutch language based associative network is of course unable to produce meaningful results on an English language article and vice versa, in the test for those monolingual associative network, only the corresponding language articles were used. Since associative networks look at single articles at a time anyway, the results of the Dutch-English associative network on each dataset might be compared, but it should be considered that different main articles (Dutch and English) were used between the two datasets within a pair, with an English main article for the English dataset and a Dutch main article for the Dutch dataset. For this reason, the data from these datasets cannot be simply added together for a final result.

10.3 Results

As can be seen in table 10.1, the Dutch-English associative network performed consistently better than its monolingual equivalents. Over the 500 documents analysed, the difference

[1]Translation: English Kings

in accuracy between the monolingual and multilingual associative networks is significant ($p < 0.05$ based on a Student's t-test).

This improvement of the multilingual associative network over the monolingual one is in line with works in cross-lingual search, such as Lavrenko et al. [2002], though other work in cross-lingual text categorization such as Bel et al. [2003] and Rigutini et al. [2005] did not find this improvement in their work, instead observing marginally lower or equal performance for their multilingual classifiers.

This difference might be explained by the fact that the classification schemes used by Bel et al. and Rigutini et al. use no inherent linguistic information in their algorithms, whereas Lavrenko et al. and associative networks both use information about relations between the words in the covered languages. Thus, the improvement made by using a multilingual associative network suggests that the additional connections in the associative network, as described in Section 5, help spread activation to related concepts, rather than simply smearing it out over more nodes.

It should be noted that the difference between the Dutch and the English test in terms of performance is related to the different articles as well as the different main articles for the categories in each of the two languages. For this reason, no direct comparison can be made in terms of the quality of the Dutch versus the English associative network, but the results of the monolingual and multilingual associative networks on the same dataset can be compared.

10.4 Conclusions

Multilingual associative networks provide an improvement similar to the one offered by Montessori training (described in Chapter 6) in terms of performance boost, but require little work in terms of creating the associative network. By basing our networks on Wikipedia, we were able to use language information already available so creating the multilingual associative network is relatively easy. As such, we think it to be a more effective way of increasing performance than Montessori training. Moreover, as Wikipedia is available in many different languages, it will be possible to make associative networks which handle different languages and even more than two languages.

A comparison to the state of the art was made in other chapters, notably in Chapter 11.

In this experiment we created a multilingual network by merging two existing associative networks, but it would also be possible to create a multilingual associative network from scratch. Trained on a multilingual data-set, this network would presumably be especially well suited for a multilingual environment, where documents of different languages were mixed. This may thus be further expanded upon in future work.

Chapter 11

Experiment 7: Comparison to State of the Art

"Be of one mind and one faith, that you may conquer your enemies and lead long and happy lives." - Genghis Kahn, as quoted by Prawdin et al. [1940]

In previous experiments we have been testing and comparing specific aspects of associative networks, improving individual steps of the greater whole. In this chapter we test our combined system and compare its performance against state-of-the-art methods by comparing our own results to those of the LSHTC3 from 2012.

11.1 Description of the LSHTC Challenge

The Third Pascal Challenge on Large Scale Hierarchical Text Classification (LSHTC3), organized in 2012, was opened to allow methods for the hierarchical classification of texts to compete and establish which ones were the most effective.

The dataset and evaluation system were made available to the public as well as the results produced and methods used by the competing teams. This allows us to establish how our system as a whole, including the various improvements we made in previous experiments, compares to the overall state of the art.

The organizers state on their website that research on large-scale classification has so far focussed on situations with large numbers of documents with a limited number of categories, highlighting the latter as something that is distinctly not the case for hierarchies, which in the problems presented by the challenge may have up to 325.000 categories.

11.1.1 Structure

The Challenge has been divided into three tracks, each with its own dataset and some with further variation.

The first track is named 'Large Scale Hierarchical Classification' and covers the primary goal of the competition. Two other tracks were present in the LSHTC Challenge, but these were not used in our experiment.

In the first track, two datasets are provided: a medium sized one based on 36.500 categories and a large sized one based on roughly 325.000 categories. The medium sized dataset holds a hierarchy of categories that forms an acyclic graph and is provided as either the original text data or as a pre-processed version, whereas the larger category hierarchy allows cycles and is made available only as a pre-processed dataset.

Pre-processing involved converting the text into a feature set where each word was stemmed and given a number, the count of the words with that stem being used as the value for that feature. The actual stem represented by the various numbers was not provided, which meant crucial information for the associative network was lost here, making the large set unusable for our experiment.

We focus on the medium sized dataset from the first track, which covers the primary purpose of the competition and allows for the best comparison. The medium sized dataset is the most common one tackled by other researchers, listing 17 participants compared to 6 for the next most tackled task. However, the reason it is the focus of our work to the exclusion of the others is because it provides a raw text dataset, unlike the other tasks and tracks which only provide pre-processed data. While no doubt useful for probability-based classifiers, associative networks rely on knowledge of the language itself. This information was lost during pre-processing, with word stems being converted into numbers that are

meaningless to the associative network. Without a conversion to establish what stem is represented by what number, the data is not usable by associative networks.

11.1.2 Evaluation

In order to evaluate the performance of competitors, the LSHTC made an online evaluation system available to which results can be submitted. Rankings amongst participants based on different criteria were made available in real time during the challenge and the online evaluation system has been kept active after the competition, allowing results to continue to be compared based on the various qualifications. Specifically, the following metrics are calculated and used for ranking purposes:

- Accuracy
- Example based Precision, Recall and F-measure
- Label-based macro Precision, Recall and F-measure
- Label-based micro Precision, Recall and F-measure
- Hierarchical Precision, Recall and F-measure

The list of measures contains some standard measures, but some differ from the traditionally used metrics. The new hierarchical measures are based on the works of Kiritchenko [2006].

In her thesis, Kiritchenko notes that traditional error measures are insufficient to properly judge hierarchical categorization, a problem we ourselves ran into in earlier experiments as well in our experiments involving hierarchical classification in Chapter 8. Kiritchenko propose the hierarchical metrics mentioned above as alternatives.

To illustrate the problem, we depict an example hierarchy in Figure 11.1, each circle representing a category with child relations represented by arrows. The correct category for a document is circled, and the category identified by the system is marked as well. The three situations depicted in the image are clearly not the same in terms of accuracy, with the two leftmost examples being much closer to the correct answer than the right one. Traditional precision / recall measures do not take this into account.

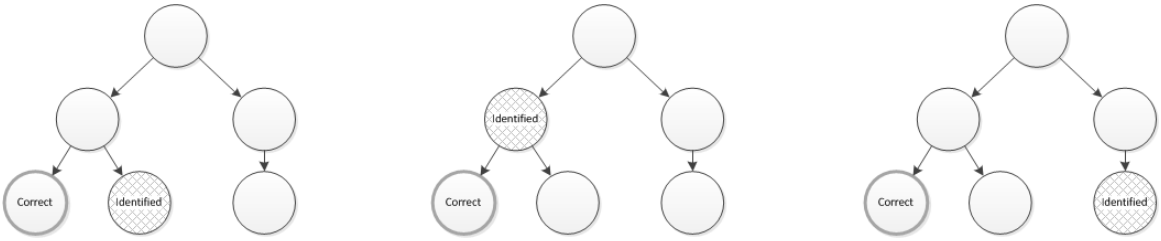

Figure 11.1: Weakness of traditional evaluation

To compensate for this problem, Kiritchenko argues that we can use the transitiveness of hierarchies (i.e. the fact that if a document belongs to a certain category, it also belongs to the parent category). By comparing the set of ancestors[1] of the correct category and of the category into which a document was actually sorted, a measure of how close the classification is can be determined - the closer the two sets match, the more accurate the categorization of the document is.

More specifically these values, which Kiritchenko has dubbed the hierarchical precision (hP) and hierarchical recall (hR), are calculated as described in Equations 11.1 and 11.2, where $C_{categorization}$ is the set of ancestors of the category into which a document was sorted (including that category) and $C_{correct}$ is the set of ancestors of the correct category (including that category):

$$hP = \frac{|C_{categorization} \cap C_{correct}|}{|C_{categorization}|} \tag{11.1}$$

$$hR = \frac{|C_{categorization} \cap C_{correct}|}{|C_{correct}|} \tag{11.2}$$

Thus, in the hierarchy depicted in Figure 11.2, if a document belongs in class E, and was categorized as belonging in class D, hP would be $1/2$ and hR would be $1/4$, as the node E has the ancestors A, B and C, thus $C_{correct}$ would be the set {A, B, C, E}, whereas $C_{categorization}$ would be the set {B, D} (the root node, which is shared by all categories, is ignored).

[1] Which in this case includes the category itself

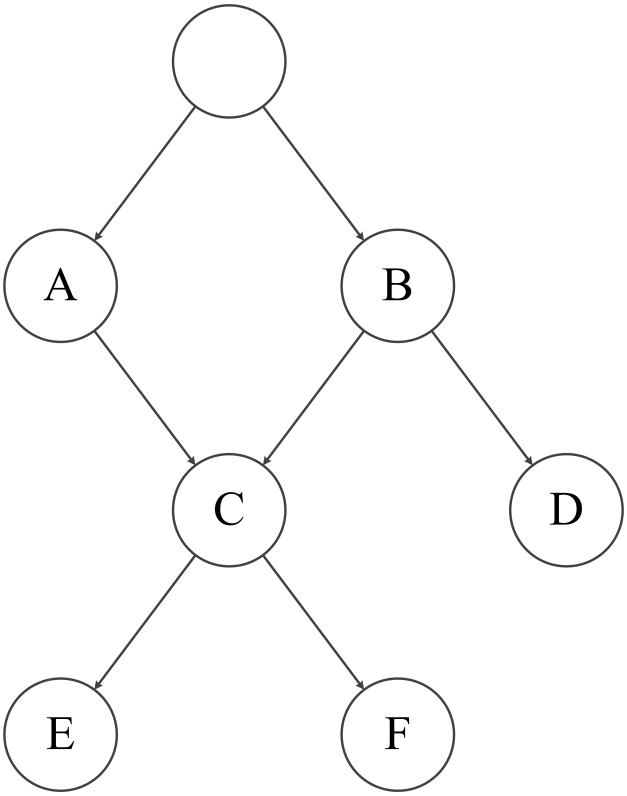

Figure 11.2: Example misclassification

For a set of multiple classifications, the top and bottom part of the equations are both summed, giving the hierarchical precision and hierarchical recall over the entire set, which can then be used to calculate an F-measure as normal.

This method thus allows for a more accurate analysis of how well a method is able to categorize documents in a hierarchy.

11.2 Other Competitors

In this Section we describe some of the methods used by the competitors in the challenge. Unfortunately, not all participants submitted a paper, so we focus on the top competitors for the medium sized Wikipedia dataset of the first track, which is the one in which we participated. In this problem, a system is expected to classify 81.262 test documents into 50.312 categories, based on 456.866 training documents.

Sasaki and Weissenbacher [2012] used a top-down method, by which a support vector machine was trained for each edge in the category hierarchy. The documents in the test set are then passed through the classifiers from the root note, moving on deeper into the hierarchy or not depending on the judgement of the support vector machine, to eventually stop in one or more leaf categories. Afterwards, a global pruning is carried out, in which unlikely categories are eliminated from amongst the results based on a confidence score.

Wang et al. [2012] also used a top-down approach but relied on a method which they have dubbed Meta-Top-Down. This method employs multiple top-down classifiers, which are used to classify data, and a system is then trained to pick the best classifier for each document. Thus, first a set of base classifiers is trained, and a meta-training set is constructed based on which classifications were or were not correctly estimated by those base classifiers. Next, a meta-classifier is trained on that training-set and the combined classifiers and meta-classifier are then used to assess the test set. This method improves slightly upon Sasaki and Weissenbacher, going on to score the highest in most measures.

Han et al. [2012]'s methods most closely resemble our own, as they compare test documents to the set of training documents to find the documents that most closely resemble it (k-NN). The overlap between the categories of these documents is then examined, weighing in documents within parent categories to find the best matching categories for a document. Their technique is similar to associative networks, though it relies on a TF-IDF measure to compare individual documents, while our own methods use associative networks for this task.

11.3 Description of the Experiment

In this section we describe the methodology we used to tackle this challenge.

11.3.1 Techniques Used

Rather than focus on a single part of the process as we did in earlier experiments, this time we utilize the entire process that we have developed for the use with associative networks as

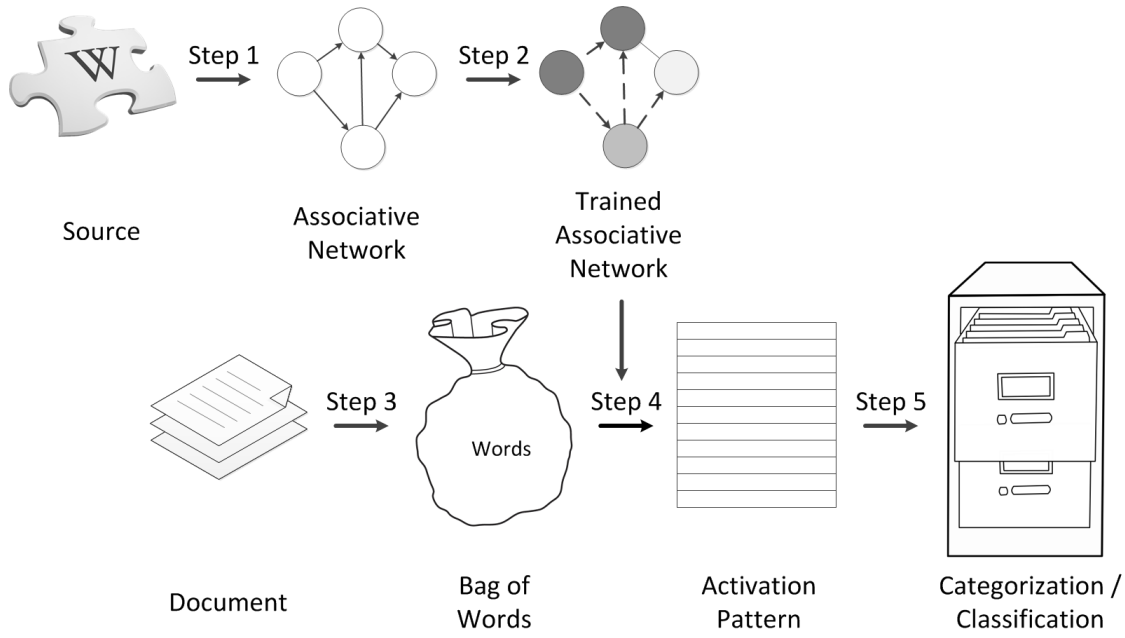

Figure 11.3: Process used in Experiment

depicted in Figure 11.3. Moreover, we utilize some of the techniques developed in earlier experiments.

Documents are first analysed using Natural Language Processing, more specifically using both part-of-speech tagging and parsing as described in Chapter 7. Part-of-speech tagging is used to create an accurate bag of words for each document (step 1). The Stanford Natural Language parser was additionally used to identify key words in the sentence, which are given additional weight in the bag.

We use a Wikipedia based associative network, as established in Chapter 5, which is the associative network that yielded the best results for us. Since the Associative Network is merely constructed based on the texts in Wikipedia, and does not actually contain any of them, we feel that this does not violate the spirit of the competition by using the same source as the one from which the training and test set were drawn (Wikipedia) for the construction of the associative network. It should not offer any advantage to the associative network beyond being seeded as a better network in the general sense with regards to this task.

Finally, we use the Flow network tested in Chapter 8 to establish the activation pattern (step 3). This method performed slightly better than the spreading activation method we first developed for use with associative networks.

11.3.2 Classification

To classify documents into a specific category, we wish to compare them to documents we know to be in each category. Rather than comparing all input to all known cases however, we instead compare each activation pattern to a combined pattern. This combined pattern is created by viewing all documents within a category as one large document consisting of the combined text of each document. The pattern most closely matching the documents pattern was then selected as the correct category.

While this method produces results, it requires a lot of comparisons for each document due to the large number of categories. A top-down classifier used by Wang et al. mentioned above would have been a more efficient approach to this.

11.4 Results

In Tables 11.1 and 11.2, we list the results of the other competitors in the LSHTC, with the various metrics by which their results were compared. This table is identical to the one presented by the LSHTC3 with our own results added for comparison.

11.5 Conclusions

Though our system was unable to defeat the highest scoring algorithms listed, associative networks performed very reasonably, with a solid spot in the middle in terms of most of the variables tested for. These results are in line with other comparisons to competing algorithms such as the one described in Chapter 5.

We expect that these results can be further improved upon in the future by using more extensive training. Using a multilingual approach, despite the dataset being in a single

		Example-based			Hierarchical		
Name	Accuracy	F1	Prec.	Recall	F1	Prec.	Recall
TTI	0.420	0.477	0.505	0.508	0.692	0.724	0.722
chrishan	0.412	0.477	0.518	0.512	0.677	0.709	0.717
arthur	0.438	0.494	0.552	0.496	0.709	0.764	0.714
szarak	0.371	0.437	0.466	0.486	0.645	0.664	0.706
coolvegpuff	0.429	0.482	0.550	0.476	0.689	0.762	0.678
anttip	0.408	0.446	0.504	0.433	0.680	0.761	0.660
dhlee	0.385	0.435	0.494	0.426	0.666	0.733	0.656
brouardc	0.354	0.418	0.479	0.432	0.643	0.694	0.668
Knn Baseline	0.249	0.318	0.283	0.416	0.561	0.512	0.691
SSir	0.327	0.387	0.434	0.398	0.639	0.700	0.640
KULeuven	0.298	0.341	0.371	0.367	0.549	0.580	0.594
daq	0.353	0.390	0.454	0.368	0.633	0.710	0.615
glouppe	0.047	0.085	0.050	0.342	0.668	0.811	0.642
hautcs	0.320	0.347	0.402	0.329	0.603	0.702	0.571
TUD_KE	0.245	0.284	0.323	0.295	0.537	0.613	0.540
Peaceguard	0.250	0.292	0.405	0.250	0.592	0.721	0.541
dicaro	0.063	0.080	0.083	0.106	0.345	0.365	0.403
A.N.	*0.326*	*0.379*	*0.380*	*0.343*	*0.611*	*0.617*	*0.612*

Table 11.1: Accuracy, Example-based and Hierarchical results

	Label-based Macro			Label-based Micro		
Name	F1	Prec.	Recall	F1	Prec.	Recall
TTI	0.284	0.506	0.307	0.473	0.477	0.468
chrishan	0.245	0.426	0.300	0.419	0.394	0.447
arthur	0.267	0.573	0.288	0.494	0.566	0.438
szarak	0.219	0.352	0.276	0.377	0.339	0.424
coolvegpuff	0.251	0.526	0.257	0.478	0.552	0.421
anttip	0.239	0.489	0.245	0.431	0.511	0.372
dhlee	0.282	0.476	0.318	0.421	0.485	0.372
brouardc	0.240	0.358	0.265	0.374	0.379	0.369
Knn Baseline	0.176	0.252	0.235	0.298	0.251	0.367
SSir	0.168	0.461	0.161	0.388	0.441	0.347
KULeuven	0.183	0.279	0.202	0.305	0.279	0.335
daq	0.193	0.400	0.203	0.365	0.434	0.316
glouppe	0.176	0.324	0.160	0.097	0.057	0.314
hautcs	0.104	0.540	0.107	0.327	0.433	0.263
TUD_KE	0.130	0.250	0.131	0.276	0.295	0.260
Peaceguard	0.055	0.470	0.054	0.275	0.405	0.208
dicaro	0.021	0.143	0.025	0.072	0.057	0.097
A.N.	*0.116*	*0.322*	*0.119*	*0.329*	*0.394*	*0.295*

Table 11.2: Label-based Macro and Micro results

language, may also boost results, as suggested by our work in Chapter 10, though we did not use this technique due to it requiring additional training data in alternative languages.

Even if associative networks did not beat some of the state-of-the-art algorithms, they have many additional advantages over these techniques in terms of usability and computational efficiency, which we describe in Chapter 13.

We can summarize all the experiments described in this part of the thesis as follows. In the first five experiments, we examined each of the steps of our process as described in Chapter 2, attempting different ways to improve upon those steps and thereby improve the quality of the outcome of our process. Additionally, in our Experiment 6 (Chapter 10) we have extended associative networks to be able to handle multilingual document sets. Finally, in this chapter we have integrated all techniques that were found to be improvements, and proved that the proposed process can measure up with many state-of-the-art techniques.

In the next chapter we will describe how the approach towards defining and implementing the process of creating associative networks that we have presented in Parts I and II has been the basis of the implementation of a real world application.

Part III

Applications and Findings

Chapter 12

Associative Networks in Practice

> *"Through observation, sensation and experience shall the truth of the multiverse be revealed."* - Fall-from-Grace [1999]

As part of our research project, we developed a system for Pagelink to help organize their digital library using associative networks. This document categorization system has been dubbed the '*Pagelink Kenniscentrum'* or Pagelink Knowledge Centre[1]. In this application, the company collects articles with diverse topics such as descriptions of the products offered by the company, technical problems users may encounter (and the way to solve them) and general background articles concerning software products.

The envisaged users consist on the one hand of the employees of the company that are in charge of content management, and on the other hand of the users who want to consult the document library, which we refer to as visitors. This group consists of Pagelink employees and customers. The Pagelink Knowledge Centre library consists of approximately 10.000 documents, written by the employees themselves or copied from sources such as Wikipedia and the Microsoft Knowledge Base. All the collected articles are made available online through the company website. Davenport and Prusak [1998] describe three aims for knowledge management projects: to make knowledge visible, to develop a knowledge-intensive culture and to build a knowledge infrastructure. The first of these is the primary goal of the Pagelink Knowledge Centre: the articles in the Pagelink Knowledge Centre are used

[1] `http://www.perrit.nl/kenniscentrum/`

to draw traffic to the company website and to demonstrate what products the company has expertise with. The third goal described by Davenport et al. is a secondary consideration: the Pagelink Knowledge Centre provides an easy source of solutions for common technical problems with the offered products, reducing the workload of the help-desk.

The Pagelink Knowledge Centre is composed of three main components: a front-end which is integrated into the company website, a back-end content management system through which content managers can add, remove or edit articles, and an associative network which services both of these components. We will examine the three parts separately, and then examine some of the practical problems encountered while during the development of the Pagelink Knowledge Centre. The chapter will conclude with addressing research question 7, posited in Chapter 1: what lessons can be learned by applying associative networks in a real-life knowledge management platform?

12.1 Pagelink Knowledge Centre Front-End and Usage

Figure 12.1 shows the front end of the Pagelink Knowledge Centre, displaying how articles appear on the company website. As the Pagelink Knowledge Centre contains thousands of articles, it is crucial to provide search and navigation functionality to help visitors find the information they are looking for. To help facilitate navigation, articles have been ordered into categories, have been enriched with tags and have links to similar articles. Generally speaking, categories cover a specific product and articles in that category therefore concern that product. Tags describe other, more generic topics that the article covers. All three types of navigation (tags, categories and linked articles) were selected based on the three most likely usage scenarios envisioned when the Pagelink Knowledge Centre was created.

The categories in the Pagelink Knowledge Centre are ordered into a hierarchy of categories and sub-categories. This hierarchy of categories is displayed as a menu that the visitor can browse, on the left side of the screen in Figure 12.1. Categories generally are named after specific products or groups of similar products (although this is not required by the system), and are intended specifically for visitors who know what they are looking for, or who need specific information on a product. This might also include visitors who are interested in purchasing that product from the company.

As said, articles are enriched with tags describing the general topics the article covers. Tags serve as a second dimension by which articles can be found in addition to categories. For example, a tag might cover the topic of '*special characters'* which might include articles from the category '*Mozilla Thunderbird'* describing how to use special characters in e-mail, as well as articles in the category '*Microsoft Office*' which describe how to fix broken special characters caused by a conversion from one format to another. Tags are displayed at the bottom of the page below the article, as depicted in Figure 12.1, and link to a list of links to all articles with that tag, in other words all articles covering that topic. Tags are intended for visitors who have a general interest in a specific topic and they allow users to find other articles about that topic.

Tags and categories both express the topic of articles and choosing whether to make a new category or a new tag for a topic can be difficult. The general guideline for article description used in the Pagelink Knowledge Centre is that products are used as categories. However there is some overlap in the system and this guideline is not strictly enforced, with some products represented by both a category and a tag, for example.

Articles also link to the top five most similar articles, based on their content. This list of related articles is displayed at the bottom of the page next to the tags, also depicted in Figure 12.1. The list of related articles is intended for visitors who have specific technical problems. Visitors with specific technical problems will have a general idea of what is wrong but they may not know the exact nature of the problem. By providing a list of articles that are similar, the visitor may be able to find the solution to their specific problem after having found a solution to a similar problem.

Articles in the Pagelink Knowledge Centre may have versions in multiple languages. By default, the Pagelink Knowledge Centre supports both Dutch and English articles, with options in place to implement additional languages. Due to the fact that visitors may only understand one of those languages, the list of related articles includes only articles that are in the same language as the article being viewed. Thus, the Dutch version of the article could be listed as a related article for a Dutch article, while the English version could be listed for a related English article. This filtering by language is done only for displaying related articles. English and Dutch articles are both found when browsing using categories and tags as browsing in this way displays all matches (rather than just the top five).

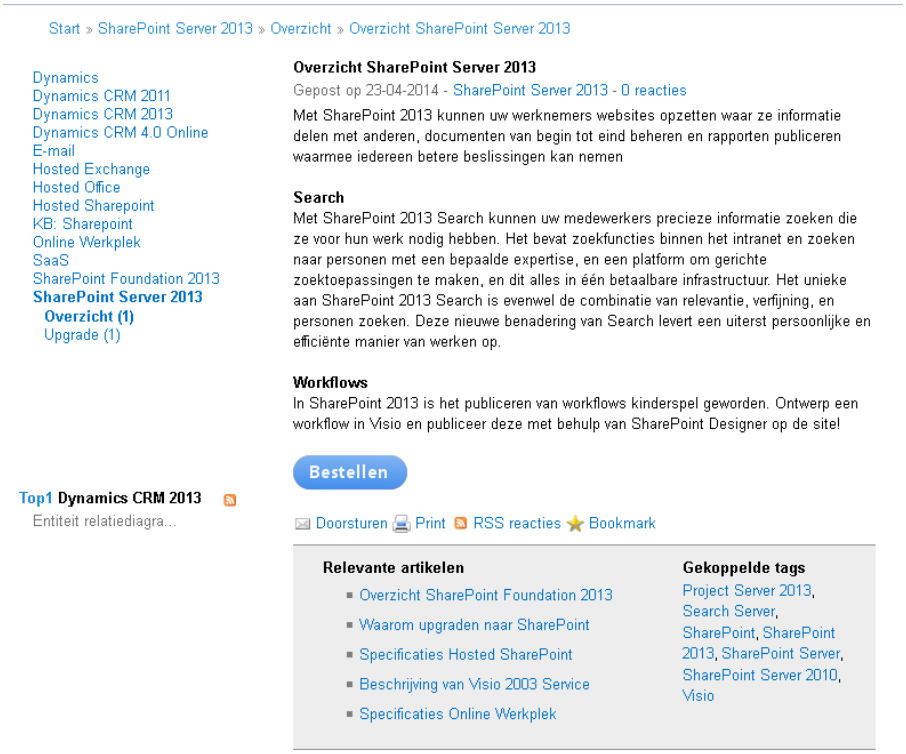

Figure 12.1: Pagelink Knowledge Centre: article presentation

12.2 Pagelink Knowledge Centre Content Management

The content of the Pagelink Knowledge Centre is controlled by the company itself. Articles are extracted from public sources, from knowledge bases of partners and in some cases written by the company's employees. Figure 12.2 displays the general interface of the content management system. At the top there are tabs to manage articles ("Artikelen"), the category structure ("Categorieën") and Tags ("Tags").

Through the Articles tab, content managers can add, remove or edit articles, as well as add translations in different languages for various articles. As many of the articles are added from external sources, the Pagelink Knowledge Centre has an API to facilitate automatic importing of articles, and an implementation of this API has been included in the application which allows the loading of articles directly from Wikipedia. The Pagelink Knowledge Centre automatically manages different versions of articles, which allows edits to be saved and reviewed before they are published, while keeping the older version active on the website.

The Category structures tab allows content managers to manage the category structure used for navigation by visitors. Categories can be nested up to any level, though content managers are advised to not go beyond two or three levels to prevent the user interface from becoming cramped. For similar reasons, the names of categories are kept short. The associative network is used to recommend a category for new articles, which is discussed in more detail in the next section.

The Tags tab allows content managers to add new tags to be used by the Pagelink Knowledge Centre, which can then be coupled with articles. The associative network is used to recommend tags for new articles, as can be seen in Figure 12.3. This is discussed in more detail in the next section.

Besides these three tabs, there are also tabs to manage the banners displayed with various articles ("Banners"), manage any maintenance messages to inform users of things such as server downtime ("Mededelingen") as well as a general settings tab ("Instellingen"). The final tab ("Namen") allows the content manager to add product names to the associative network, discussed further in the Section 12.4.

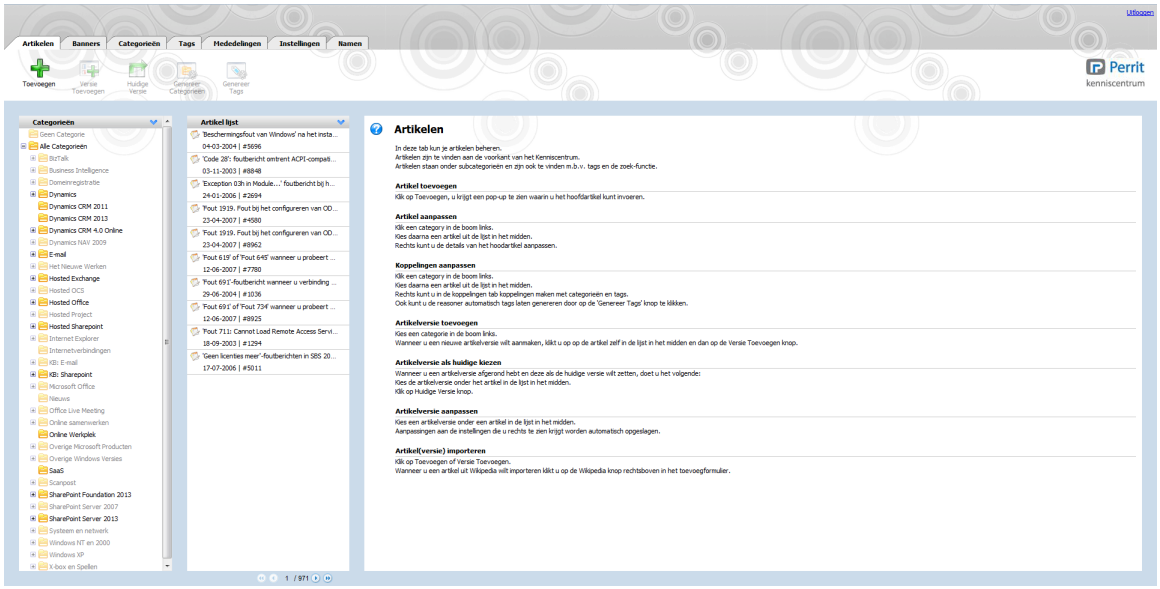

Figure 12.2: Pagelink Knowledge Centre: content manager interface

12.3 Associative Network and the Pagelink Knowledge Centre

So far we have explained how the Pagelink Knowledge Centre works. In this section we describe what role associative networks play in this platform.

In the front-end, the list of related articles is generated directly by an associative network, thus the associative network provides suggestions to the user for other content they may be interested in, based on similarity between articles. Visitors looking for solutions to technical problems may browse through articles that are relevant to their problem. When it turns out the first article they find is not relevant, a list of most closely related articles can help users to find articles that are more relevant for solving their problem.

For content managers, the associative network offers some additional features, such as suggesting categories and tags for new articles. This not only saves time, but additionally allows users not familiar with the available tags and categories to easily add this information.

When a new article is added, the associative network is used to offer suggestions for which category the article might fit into, which the user adding the article can either reject

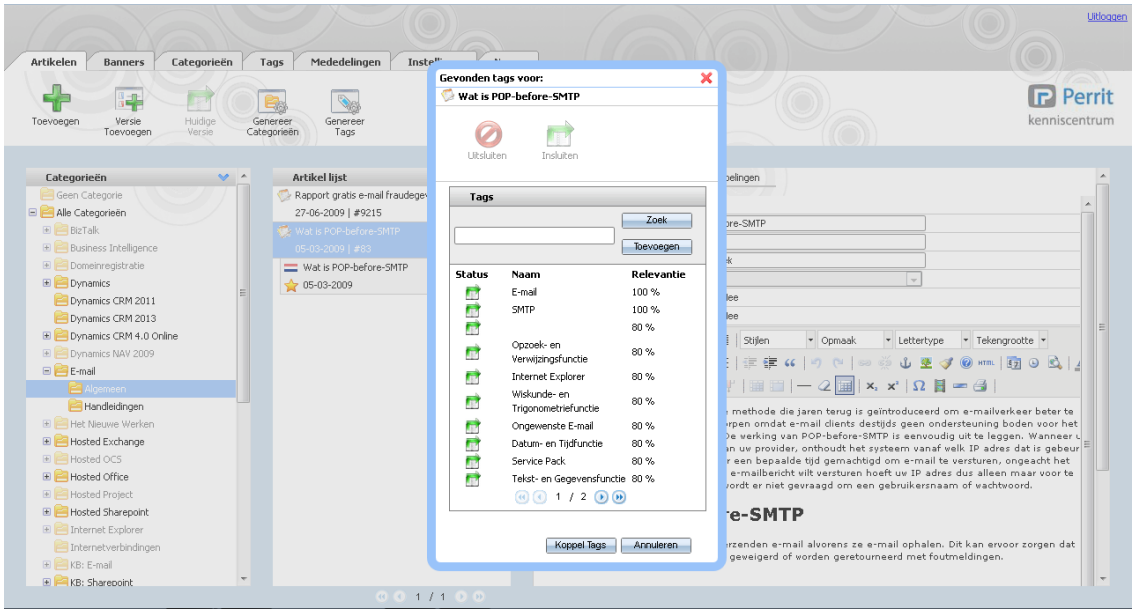

Figure 12.3: Pagelink Knowledge Centre: content manager interface to add tags

or accept. The suggestion is made based on the content of the newly written article, which is compared to all articles in all existing categories. The category with the lowest average distance is suggested as the category for the new article. Empty categories are ignored. Thus, the associative network uses classification, rather than categorization, as described in Chapter 2. The reason for using this less powerful method of grouping is that the role of the Pagelink Knowledge Centre is also to advertise company's expertise in certain fields. By controlling the categories and sub-categories that are used, the company can display expertise with the products represented by those categories.

When matching tags, the name of the tag itself is used as input for the associative network (rather than using the existing articles that have that specific tag, as with categories). The tag name is used to create an activation pattern, as if the tag name were a miniature article, and the most closely matching "tag articles" are then listed as suggestions.

Using tags as miniature articles also fits well with the difference between tags and categories described earlier, with tags describing generic topics. It is easier to generate an activation pattern for tags named after general topics than it is for tags named after specific product names as is common with categories, because the specific product is usually not

listed in the associative network, while general topics usually are. Figure 12.3 depicts the interface with suggestions for tags being automatically selected in the list for a new article that is being added.

We have not done an extensive assessment of the quality of the tag suggestion function. A general glance suggests that most suggested matches are according to expectation. As future work, an assessment of the quality of tag suggestions could be made by comparing the tags suggested for articles to a golden standard provided by a panel of experts.

All three functions - categorizing articles, suggesting tags and finding closely related articles - are done by a single associative network that performs each of these tasks. The ability to perform all these tasks without requiring additional training illustrates the versatile nature of associative networks, which we will discuss in more detail in the next chapter.

12.4 Practical Problems

During the creation and use of the Pagelink Knowledge Centre, two practical problems arose which negatively impacted the usability of the system. The first and most obvious one was the multilingual nature of the dataset. In early versions, the Pagelink Knowledge Centre used a monolingual associative network, based on the fact that most articles in the Pagelink Knowledge Centre at the time were in Dutch. However, as the number of articles in the Pagelink Knowledge Centre grew, so too did the number of English articles. Most people in the Netherlands speak both Dutch and English[2], so for most users the fact that articles might be available only in English is not an issue, but as the associative network did not support articles in different languages, it produced some rather unpredictable results.

This problem was one of the reasons for developing multilingual associative networks, which are discussed in more detail in Chapter 10. The associative network used in the Pagelink Knowledge Centre is Wikipedia-based and can handle both English and Dutch, similar to the one we developed in Chapter 10. A remaining limitation is that while it is easy to allow articles to have versions in additional languages beyond Dutch and English, adding such versions will not automatically update the associative network, which means that adding articles in, for example, German would not be very beneficial.

[2]Up to 90% of all Dutch people speak English [European Commission, 2012]

The other problem encountered was much more dataset-specific. The Pagelink Knowledge Centre features many articles concerning specific products such as Microsoft Office or SharePoint. For one, many of these products have different versions and editions. The official names for these different versions are not always used correctly and some names may be abbreviated. For example, Microsoft Office XP might also be referred to as Microsoft Office, Office XP or just Office. Likewise, products may have common abbreviations such as Win7 for Windows 7. In many cases, the associative network will not know how to relate these terms to one another.

One possible solution is to make use of Wikipedia redirects as alternative names for the same product. This leaves the name variations and abbreviations as a different surface form expressing the same concept, just like we do for many other concepts. However, not all abbreviations can be matched in this way. For example, the term "Office" is simply too generic to match in this manner and some products may not be notable enough for Wikipedia to have an entry at all. As the Pagelink Knowledge Centre is continuously updated as new products become available, however, redirects might not even exist in the associative network, and it would not be desirable to have to rebuild the associative network each time Wikipedia is updated.

Instead, we added a special additional method by which content managers may interact with the associative network, in the form of a "Names" tab in the back-end interface (see Figure 12.2). Here, content managers can add an entry for a product, and list alternative names and abbreviations. These products are added automatically to the associative network as independent nodes, not otherwise connected to the associative network. While this means they are not fully integrated into the system, it at least allows the associative network to add these nodes to the activation pattern and to map different variations and abbreviations of the name to a single concept.

We have not done an assessment of the impact of this method on the quality of the results produced by the associative network, though as with tag suggestion, informal inspection suggests that this works as intended. Future work on this topic may provide more insights, and may offer ways to link the product more firmly to the rest of the associative network, allowing it to receive activation from and spread activation through the rest of the network. As most of the new products are notable enough to be covered by Wikipedia

once they are released, and as Wikipedia usually offers automatic redirects for common abbreviations, it may be possible to automatically update a Wikipedia-based associative network automatically as new articles are added to Wikipedia. For products that are not notable enough, a semi-structured corporate wiki, such as the one developed by Wanders and Te Brinke [2014] might be used instead.

12.5 Conclusion

The Pagelink Knowledge Centre places associative networks firmly in the real world as a practical application. The versatile nature of associative networks allows the formalism to be used for various system features that support both content managers and visitors to the site, and versatility is further highlighted by the fact that only a single associative network performs all these tasks, which additionally simplifies the design (and thereby the cost of maintenance and updating) of the system significantly.

Coming back to our research question about what lessons can be learned by applying associative networks in this real-life knowledge management platform, we find that multilingualism is a key feature in practice. Additionally, though potentially dataset-specific, dealing with new terms as well as abbreviations can be a potential pitfall. Some of these issues have since been fully resolved, while for others options have been identified for how they can be solved in the future.

Commercially, the Pagelink Knowledge Centre is just the beginning of the practical implementation of associative networks and Pagelink intends to create more products based on associative networks. Amongst other future possibilities, the company is exploring the creation of a system to reduce the workload for help-desks by providing answers to questions by customers beyond what current search technology can do. Such a system would use associative networks to find articles with relevant information based on a question or description of the problem provided by the customer. Pagelink is also examining the use of associative networks to find case precedents for legal disputes. For all these future steps, the versatile nature of associative networks could be a key feature to speed up development while producing results that are on a par with other state-of-the-art methods.

Chapter 13

Discussion

> *"Just make the whole thing you know, cooler... It needs to be about 20% cooler."* - Rainbow Dash [2011]

The previous parts of this thesis examined various aspects of the performance of associative networks for document grouping. We have explored the relationship between natural language processing and associative networks and have examined the way in which associative networks can be created, trained and used most effectively.

In this chapter we will discuss some of our findings in more detail, specifically looking at insights gained about associative networks as well as their strengths and weaknesses.

13.1 General Insights

During our experiments, many new insights were gained into the way associative networks operate. This section discusses insights which are not directly related to our research questions, as the latter are discussed in detail in Chapter 14. Here we list insights that go beyond these particular questions.

- Wikipedia is an excellent source for creating associative networks
- More generally, sources useful for creating associative networks need not be very structured

In our first experiment (see Chapter 5), we examined different sources to feed the initial structures and values when constructing the associative network, specifically looking at synonymy-based, WordNet-based and Wikipedia-based associative networks. We found that Wikipedia-based networks give the best performance in terms of accuracy.

A key factor here was that amongst these three types of associative networks, only the Wikipedia-based associative networks have access to *a priori* information about the relationship between concepts, which can be extracted from the Wikipedia data but not from WordNet or synonym lists. Thus, unlike synonymy- and WordNet-based associative networks, Wikipedia-based associative networks will start out with relatively accurate weights, thus requiring less training. Another difference is that Wikipedia, being an encyclopaedia, has a larger list of concepts than WordNet, which more closely resembles a dictionary. Thus it is more likely that the name of a person or location is present in Wikipedia than in WordNet.

Both these differences – or more accurately, the better performance of Wikipedia-based associative networks that results from them – illustrate the importance of the data used to construct and train the associative networks: the higher the quality of the information used to construct the associative network, the better the results. Of course this is not surprising, and holds for most problems in the field of artificial intelligence. In our particular case however, we have shown that while external information needs to be present and preferably of high quality, it need not necessarily be highly structured, and we can thus easily use many existing sources of information to build associative networks. As we have shown by building associative networks using Wikipedia, it is not necessary to create a detailed ontology to describe reality such as attempted by Lenat [1995].

- Associative networks are tolerant of errors in the input data
- The ability to compensate for errors stems from the use of association concentration

Training the network (as described in Chapter 6) can eliminate errors in the associative network itself without a need to manually check if the value of each relationship between concepts makes sense (which would be impractical given the hundreds of thousands of concepts and millions of relations in the average associative network).

During classification, associative networks are also quite tolerant when it comes to errors within the documents themselves. Gamon [2004] noted, in his work on sentiment based classification of documents, that using a large number of features was beneficial for classification performance. Cohen and Singer [1999] found that their context sensitive classifiers produced better results than context insensitive methods, and were especially efficient on noisy corpora. Associative networks both use a large number of features and are context sensitive (but not in the same way as the methods examined by Cohen et al.).

Though tolerance of inaccurate data is not a unique feature of associative networks, it is a more generic benefit within the model of associative networks: in Experiment 2 (Chapter 6) we noted that the improvements made by using Montessori Training were not as great as we had expected. In Montessori Training, training data was provided that contains only key features, eliminating superfluous data as much as possible to allow the network to learn only the key connections. The ability of associative networks to be forgiving of inaccurate data is a likely cause of the low improvement: associative networks do not have much trouble with learning from noisy data, so the improvement made by removing the noise is minimal. However, more accurate data will still bring better results. This holds both during training as was established in Experiment 2, and with regular data processing as seen in Experiment 3 (Chapter 7).

Associative networks closely resemble humans in terms of strategy when dealing with information. Similar to associative networks, humans use a lot of background knowledge. The more people know about a topic the better they are able to resolve problems, but they do not require their knowledge to be structured. Also, humans are highly capable of pattern recognition, even in the most noisy of data, which is in fact a strength that is used to distinguish man from machine such as in captchas, the decoding tasks users have to perform to register for internet sites (see Figure 13.1). Similarly, associative networks are able to deal with noisy data with relative ease.

The key to this ability may be found in the way association concentration works. Through association concentration, activation spreads out across the network from many points of observation (words in our use of associative networks for text classification). That

Figure 13.1: Distinguishing man from machine

activation concentrates in nodes which are closely connected to the many different observations. Even if some of those observations are incorrect, or do not match perfectly, their combined activation will still concentrate in nodes related to the document.

Based on the results of Experiment 4 (Chapter 8), where two different methods to calculate the association concentration were compared, it was concluded that the method used is not a major factor in getting better results.

- Additional, meaningful connections within the network improve the quality of the output

Association concentration relies on finding multiple connections between concepts, some more direct than others. By spreading activation from multiple sources, and finding multiple paths from each source, a vast number of connections can be found between various concepts.

Improving the number of meaningful connections appears to improve the quality of the categorizations made by associative networks. When switching from WordNet to Wikipedia as a source, as in Experiment 1 (Chapter 5), increasing the number of nodes and connections by using a different, more interconnected source, improved the accuracy of the associative networks. In Experiment 6 (Chapter 10) a multilingual associative network was used, with new possible connections via the additional languages, again improving categorization results.

In both cases, adding more connections also meant more paths exist between concepts. As activation spreads, more paths may be found between related concepts, leading to more activation being concentrated towards related concepts. Of course, more connections also means activation might spread more easily in the wrong direction, but association concentration means that activation still concentrates in the correct nodes.

The way additional paths lead to a better overall result - even if some paths are wrong - resembles an emergent wisdom of the crowd [Surowiecki, 2005]. A whole 'crowd' of paths is generated by the spreading activation, each path suggesting that a certain concept is related to a document. Some of those paths correctly spread activation to concepts related to the document, some activate semi-related concepts and some go way off to topics that are not related to the document at all. Together these paths on average send the highest activation to closely related concepts, lower activation to semi-related concepts and none or almost no activation to distant concepts. This results in an emergent wisdom of the crowds - even with noise, overall the correct answer still emerges from the data.

In Experiment 6 which covered multilingual associative networks, we increased the number of ways in which concepts can be sensibly linked by adding a second language through which paths could connect concepts, as they can now go through related concepts in both languages. This means that there are more possible paths between concepts. More possible paths mean a bigger 'crowd', and in turn the emergent wisdom of the crowd improves in accuracy. Having additional connections from using a different source for the network, as in Experiment 1, follows the same pattern.

The trend that additional, meaningful connections improve the quality of the associative network was not investigated in great detail, but it is a notable feature which may warrant a closer look in the future. The word 'meaningful' is crucial here of course, as it is not the case that simply adding more connections will lead to better results.

13.2 Strengths of Associative Networks

Associative networks have a wide range of possible applications. Our main research has focussed on various forms of document grouping, and for this task associative networks

have proven to be a versatile tool with good results. In various experiments, we have taken on increasingly complicated methods of document grouping:

Classification – In Experiment 3 (Chapter 7), associative networks were used for basic classification, matching the activation patterns of documents to the activation pattern of a document exemplifying that category.

Hierarchical Classification – In Experiment 7 (Chapter 11), documents were instead sorted into a hierarchical classification. In this structure, classes are still provided, but they form a hierarchy, rather than being independent. In the hierarchy, there is no single document exemplifying each category, so instead we created a combined pattern made up of the patterns of all documents within the hierarchy.

Categorization – Experiment 4 (Chapter 8) went one step further still, from classification to categorization (see also Figure 2.3 and Figure 2.2). In this experiment, we no longer compared the activation patterns of documents to category patterns, but rather directly to other documents, using a clustering algorithm to group related documents together.

The fact that with very little adjustment associative networks can be used to classify, classify hierarchically or categorize documents illustrates the wide usability of associative networks within the field of document grouping.

Beyond document grouping, it is easy to envision associative networks being used for other purposes. Search is one possible use, and can be done by matching the activation patterns of a search query to those of individual documents. Moreover, through association concentration, the activation patterns of documents shows which topics are central to each text, as key concepts are highlighted by a high activation in the activation pattern. This can in turn be used to add tags to documents for concept-based browsing, topic extraction or, as we did in Experiment 4, to identify the topic shared between a set of documents.

All those different tasks and benefits can be achieved with a singular associative network. It is thus possible to train an associative network once and use it for a multitude of tasks. We discussed some different sources to construct associative networks in Chapter 5, and different methods of training and activation in Chapters 6 and 8. Little or no training is required to adjust an existing associative network to a new dataset. In fact, to meet the specific requirements for a scenario of use, primarily the pre- and post-processing needs to be adjusted, not the associative network itself.

Another important benefit of associative networks established in Experiment 6 (Chapter 10) is that associative networks are capable of handling texts in multiple languages. Associative networks are easily constructed using Wikipedia, as established in Experiment 1 (Chapter 5), and we can use the multilingual nature of Wikipedia to create a multilingual associative network that can group documents in different languages together.

Very little adjustment or extra work is required to make an associative network multilingual, and being multilingual even improves performance of the network overall. As a consequence, the potential for applying associative networks is even wider.

Associative networks are quite easy to train. The method of back-propagation, described in Chapter 6, only requires labelled textual data, and once that is provided, training can be done automatically. Moreover, associative networks can even be trained while they are in use, learning in between queries and thereby adjusting to changes in the dataset on the fly, as long as some sort of feedback mechanism is provided. While the method of Montessori training explored in Experiment 2 (Chapter 6) requires more time and effort to implement, the improvement over simple to implement training methods based on back-propagation is low. This argues for mostly automated training of the associative network.

In Experiment 4 and 7, we applied associative networks to very large libraries, consisting of hundreds of thousands of documents. Even so, associative networks remain very fast, in part due to the very low computational complexity of the algorithm (O(V)). As discussed in Chapter 3, associative networks benefit a lot from the sparse nature of their design, which makes the algorithm very fast despite the limited memory requirement.

We have put the associative networks up to the test against several state-of-the-art algorithms such as support vector machines and k-nearest neighbours, first in Experiment 1 and later in Experiment 7, finding that associative networks perform on a par with the state of the art. These methods however do not have the enormous versatility, easy training and low computational complexity of associative networks.

13.3 Limitations of Associative Networks

Associative networks are not without limitations. By converting documents into a bag of words, as explained in Chapter 2, we can significantly simplify the documents in a library. This method has been used successfully with many classification methods [Joachims, 1998, McCallum and Nigam, 1998], but it also removes information from the data that is expressed by the grammar and word ordering.

That loss of information means the associative network has less (and less accurate) data to work with than is actually made available in the text it processes. In Experiment 3 (Chapter 7), we looked at methods to improve on the bag of words method and retain some of the information lost in creating the bag. We used part-of-speech tagging to better match between the text in the document and the words that go into the bag based on the grammatical structure, while natural language parsing was used to identify which parts of a sentence are more important.

Despite these improvements, the associative network still infers the general topic based on the words in the documents, rather than on a detailed understanding of what has been written. For classification and categorization purposes, determining the general topic of a document is usually sufficient, but it is insufficient for tasks such as question answering, for example.

We expect it may at some point be possible to do tasks like question answering by using associative networks, but it will mean making a significant deviation from the bag of words method currently used, swapping it for a model that can more accurately represent the sentence structure itself. Moreover, this model should be able to represent sentence structure information into the associative network itself, not just process it beforehand and then converting it into a bag of words as in Experiment 3. That step goes far beyond the scope of this research.

Another limitation of associative networks is their limited ability to deal with numerals. Computers are naturally very capable of handling numbers and more generally the language of mathematics, but the absolute world of numbers is problematic for associative networks, as the distance within an associative network between the numbers four and six may be the same as the distance between six and seven, for example – the actual value represented

by the numbers has little impact on this as they are just another word in the network. The same problem crops up when dealing with terms referring to dates and durations. In the future, we may look at solutions similar to part-of-speech tagging as in Experiment 3, or Montessori training, as in Experiment 2 (Chapter 6) to better deal with numbers, dates and durations, but for the moment they remain difficult to express accurately using association and cannot be used as easily as when using mathematics-based solutions.

A final limitation applies for documents that cover multiple topics. Associative networks are quite capable of extracting the core topics of a document, even if the document gives a broad overview of that topic. Thus, a document describing a country, covering both its history and geography would still be identified as being a document about that country by an associative network. But as texts grow longer and cover a wider range of topics, the associative network will eventually be overwhelmed by the data and create less and less accurate representations of the article as a whole, as dominant topics become harder and harder to identify. This makes sense as by definition there will be less dominance of the primary topics when a wide range of side topics is covered by a document.

When handling large documents – rather than large collections of smaller documents – this should be taken into account. One solution might be topic segmentation, where larger documents are divided up into smaller sections, such as extracting a pattern for each chapter, thereby using the divisions already put in place by the person or people who created the document. Reynar [1998] offers an overview of various algorithms used for topic segmentation, and though we have not done any research in this area, it might be interesting to see how associative networks fare with these when organizing literal libraries, where each document is an entire book.

Chapter 14

Conclusion

"It's the job that's never started as takes longest to finish." - Tolkien [1954]

In this chapter, we will examine the research questions that we raised in Chapter 1. We will re-examine the paradoxes discussed there and we will explore the future for associative networks.

14.1 Research Questions

In Chapter 1, we raised seven questions that we hoped to answer. In this section, we will discuss the answers that we found to each of these questions.

1. *How can an associative network be created such that it does not require a large amount of manual configuration?*

In Experiment 1 (Chapter 5), we examined several methods by which associative networks can be created. These methods use existing data sources to build associative networks. The best performing sources that we examined were WordNet and Wikipedia, both of which are publicly and freely available, and we found that Wikipedia gave the highest performance in terms of accuracy between the two.

By using synsets in WordNet or articles in Wikipedia as nodes in the associative network, and using the links already defined by these sources as edges, almost all the work

of constructing the network can be done without manual configuration. Moreover, as these data-sources are updated, new associative networks can easily be created. The same methodologies to construct associative networks from WordNet or Wikipedia can also be used with other sources at the basis.

2. *How can the connections in an associative network be trained to accurately represent the associations between the concepts modelled in the network?*

In Experiment 2 (Chapter 6), we examined the use of back-propagation to train associative networks. Based on the same technique used in neural networks, we can boost connections in the associative network that have led to correct classifications while weakening connections that have led to incorrect classifications.

Back-propagation requires very little effort on the part of the trainer, as it is only necessary to confirm or deny that a certain result is indeed correct. Beyond this, the associative network needs no feedback or manual configuration.

Additionally, we have examined Montessori learning, a method we developed inspired by the works of Maria Montessori [1909]. This method requires more effort on the part of the trainer in preparing training samples and produces slightly better results.

3. *How can the input used by an associative network be improved by using Natural Language Processing?*

Associative networks take as input a bag of words extracted from the documents being categorized. In Experiment 3 (Chapter 7) we examined the use of Natural Language Processing (NLP) to help improve the quality of that bag of words.

We used part-of-speech tagging [Toutanova et al., 2003] to more accurately identify which surface form belongs to which lemma and natural language parsing [Klein and Manning, 2003] to give additional weight to key words in the sentence, both of which improved the accuracy of our results. These techniques thus created a better match between the actual text and the activation of nodes in the associative network, eliminating terms that were not related to the document but that had homonyms in the actual text and modifying the input to place more weight on words that were relevant than on words that were not instead of counting them equally.

As discussed in chapter 7, we have used only these two rudimentary NLP techniques to create these improvements, and further research may reveal additional ways in which NLP and associative networks can be used together.

4. *How can a model of associative networks be used to automatically group documents?*

We propose a method to use associative networks for document grouping, described in detail in Chapter 2. In our method, we extract a bag of words from documents which can then be used to find associated concepts in an associative network. In Experiment 4, we examined different methods to make such association patterns. Regardless of the method used to create the patterns, we can compare the patterns to find more accurate connections between articles than we would with just the bag of words. In turn, the patterns can be used to accurately classify or categorize those documents.

5. *Which methods can improve the groupings created by associative networks, and can those methods provide additional insights into the structure of associative networks?*

Associative networks can be used to find how closely two documents are related. When comparing a large number of documents, patterns become apparent, with clusters of documents being closely related. We can extract these patterns and use them to categorize these documents, as we did in Experiment 4. In Experiment 5 (Chapter 9) , we used Power Graph Analysis, a technique originally developed to better understand protein networks by identifying clusters of proteins that interact in a similar way, to find clusters of documents that relate to similar topics, thereby producing more accurate categorizations. To be able to do this, we expanded Power Graph Analysis so that the technique is capable of handling weighted graphs.

We additionally applied Power Graph Analysis to the concepts in the associative network itself. By finding clusters of concepts within the associative network, we were able to create a simpler visualisation, similar to what Royer et al. [2008] did with protein networks. This simpler visualization gave a clearer insight into the structure of the associative network, allowing us to make a quick scan to determine the generic quality of an associative network as it is being created or trained.

6. *How can an associative network be expanded to handle document collections with documents in multiple languages?*

We created a multilingual associative network as part of Experiment 6 (Chapter 10), which allowed the associative network to directly compare English and Dutch documents, by combining Wikipedia-based associative networks in both languages. In our experiment, we found that multilingual associative networks were easy to create, and they produced results that were better than those of a monolingual network.

As we used connections between Wikipedia articles in multiple languages to accomplish this goal, the technique is very easy to reproduce and expand to other languages, especially if those languages have a relatively good Wikipedia version (and most languages do). It requires comparatively little effort (or in fact, understanding of the language) on the part of the person creating the associative network.

7. *What lessons can be learned by applying associative networks in a real-life knowledge management platform?*

In Chapter 12, we examined a practical application of associative networks in the Pagelink Knowledge Centre and we described the many benefits offered for it by the method of associative networks presented in this thesis.

The method offers a new way by which many text processing tasks can be completed, including but not limited to document categorization, hierarchical classification and clustering. Our method can easily be implemented for any language for which a data source is available and it has a low computational complexity. In Experiment 7 (Chapter 11) we compared the results of our method for document grouping to the state-of-the-art techniques in terms of the accuracy of the results, finding that associative networks performed on a par with the top methods.

14.2 Future Work

There are still many additional improvements that can be made to the method of using associative networks, and future avenues to explore:

- *Seeding:* new and better sources for associative networks may be or may become available with which to seed the associative network. The Google Knowledge Graph

[Singhal, 2012] might be one such source, and Word2Vec [Mikolov et al., 2013] and ConceptNet [Havasi et al., 2007] might others. It might also be possible to integrate multiple sources together to form a single network, for example by using the generic probabilistic approach to combine data from multiple sources proposed by Wanders et al. [2014].

- *Training:* Our proposal of using Montessori training produced slightly better results at the cost of greater effort. The cost of that effort might be reduced in the future by using existing sources to create training samples. As we noted in Chapter 6 the dictionary definition was used as a guideline, but perhaps a dictionary with slightly more extended definitions might be usable as a direct source of training examples. Alternatively, new methods of training may be developed that allow associative networks to be trained with better weights, possibly borrowed from fields such as neural networks as we did with back-propagation.

- *Natural Language Processing:* In our experiments, we only used basic, off-the-shelf methods of natural language processing to improve the input. A lot might be gained by using different part-of-speech tagging techniques or parsers to further improve the input for the associative network. Named entity recognition might also offer benefits as it will likely lead to more accurate matching between the words in the document and the concepts intended by the author.

- *Association Concentration:* We tested two algorithms for association concentration, but additional methods might be viable and may even produce better results. One possibility might be to spread activation successively, that is, by activating each term in the bag of words individually, allowing it to spread through the network, then adding the spread of the next term until all terms have been activated. This could allow activation to concentrate in certain nodes more easily, which might improve the quality of the results. Successive activation might also enable associative networks to abandon the bag-of-words approach in favour of one where word order is preserved, to counteract the loss of information resulting from using a bag of words (see Chapter 13). Such a model lowers the activation of nodes as new words are entered

in succession, while keeping track of nodes that - at one point during the process - have had a high activation.

- *Clustering:* Estivill-Castro [2002] argues that there are many different ways to define what such a cluster is supposed to be and as a result there are many different algorithms to find clusters. Based on this, there might be other ways of clustering documents than the method of power graph analysis which we propose which are better suited for document grouping.

- *Multilingualism:* We have examined a bilingual associative network for Dutch and English, but other language pairs - especially languages which are conceptually more different from English such as Chinese - might offer different results. Additionally, it may be interesting to see what results an associative network with more than two languages could produce. All of these techniques might further improve the results of our method, even beyond the current state of the art.

- *Resolving current limitations:* In Chapter 13, we mentioned several limitations of associative networks, notably the ability to deal with numerals and the ability to handle documents covering multiple topics. The ability to deal with numerals might be improved by using part-of-speech taggers to distinguish dates and numbers as well as using Montessori training focussed on numbers to more accurately set the weights in the associative network for them. Documents covering multiple topics might be split into parts, with activation patterns for each part being combined to find the generic topic of the greater document.

- *Other tasks:* We have applied our method to the task of document grouping, but we have already mentioned various other possible applications throughout earlier chapters. We have used associative networks for tag suggestion, but this was not explored further. Various other text processing tasks might benefit from associative networks. The applications of the technique are not limited to text processing, but might include any number of problems that people can do well, and which computers cannot, such as pattern recognition in visual data.

- *Self-learning network:* Associative networks currently need to be created from a source. A self-learning associative network, or one which updates automatically based on changes in the source as proposed in Chapter 12, would be able to integrate new concepts on the fly. This might offer significant benefits, and would at the least fix a problem in finding associations for new terms, such as names of new products, which we encountered during our practical application in the Pagelink Knowledge Centre.

All in all, many new avenues of exploration are possible. We believe our work will form a solid basis for these future developments.

14.3 Paradoxes Revisited

In the first chapter of this thesis, we discussed several classical paradoxes, most prominently covering the Ship of Theseus and the river of Heraclitus. Each of these paradoxes, we posited, arises from the divide between the way language is naturally understood and the language of mathematics.

We described informally the way in which the contrast between the language of the nervous system, that is the way the human brain naturally represents information, and the language of mathematics brought about these paradoxes, and through our research we have developed a method by which these paradoxes can be understood in terms of associative networks.

14.3.1 The Ship of Theseus Paradox

In Chapter 1, we described the contrast between the languages of the mind using the Ship of Theseus paradox, which asks if a ship remains the same when one replaces the planks, ropes and other parts one by one. In parallel, we wondered if a text remains the same if a word in that text is exchanged with a similar word.

At some level, one might argue that two words would only very rarely mean the very same thing, always carrying a different connotation or undertone. This view, based on a strict definition of synonymy often attributed to Leibniz, would mean that almost any

word replacement would change the meaning of the text. Quine [1951] even provides a detailed discussion of the logical requirements that a word must meet in order to be truly synonymous with another word, including the truth value of statements including the word.

Quine touches on some of the shortcomings in these requirements, but in their essence they remain an expression of absolutes that follows the language of mathematics. Just as the Ship of Theseus is fundamentally changed by replacing a plank, so too is a text fundamentally changed by almost any word replacement.

Miller et al. [1990] in building WordNet which relies heavily on synonymy relations between words, ran into this argument as well and chose a more relaxed definition for synonymy which was more practical for dealing with language as it is normally used. They stated (fitting very well with our Ship of Theseus example) that replacing the word *plank* with the word *board* in a text – provided it was in the context of carpentry and not, for example, in the sense of a *board of directors* – would be perfectly acceptable without changing the topic of the text.

This less strict definition, which has a wider application in practice as Miller et al. observed, follows the principles of the language of the nervous system much more closely, as, through its associative nature, it cares less for the exact details. What is most notable, however, is the requirement of context which Miller et al. raise, as context is exactly what associative networks use when inferring information about a text.

In the language of the nervous system, the Ship of Theseus is not changed because a plank is replaced, because the context of the ship is unaltered. Not only are the other parts of the ship unchanged, but the association with Theseus and his heroics, and the use of the ship by the Athenians remains the same despite the replaced plank. Its context is unaltered, even if a detail is changed. Similarly, the content of a text would be unchanged if a word is replaced by its synonym, and as such its categorization (to stick more closely to our practical application of associative networks) would be unaffected. The idea of the ship and of the category both hold against minor changes or noise in the data, as the general features overrule the minor details.

That is not to say that the Ship of Theseus is immutable: if its generic form is significantly altered, or if the reverence by the Athenians for Theseus and the use of his ship changes, eventually it will become just another ship – a change in its context can alter

the ship just as much as changes to the ship itself. Likewise, fundamental changes to a document may push it into a different category, but changes to the category system itself may also change the category of a document. This is especially relevant when the category system is itself founded on documents within the system, such as when classifying documents based on other documents in the categories or when categorizing an entire library: in this case, other documents added, altered or removed may impact the category of existing documents, even if the document itself is unchanged.

The barrier between categories is based not just on the conceptual understanding of the category and the documents within that category, but it is also defined by all other categories and the documents within those categories. In other words, the Ship of Theseus is defined as much by our understanding of SHIP as it is by our understanding of THESEUS and even our understanding of PLANK. The entire context, including our own understanding of that context, must be considered.

14.3.2 Bridging Heraclitus' River

When Heraclitus said that one could never step into the same river twice, he was describing the ever changing nature of the river, which is nearly impossible to describe in the language of mathematics. Despite this constant change, we can understand the river, and its properties within the associative language of the nervous system. Using associative networks, computers can capture concepts like a river in the same intuitive way.

Modelling associative thinking is obviously not enough to resolve all problems in this field. Many more challenges await - in the previous chapter we raised some of the weaknesses of associative networks that still need to be addressed. These include their weak ability to handle numbers and dates, finding better ways to extract information from text than the bag-of-words method we have used and finding better ways to deal with large documents. Beyond document categorization, there are many other fields in which associative networks may be of use. We believe that fields in which humans traditionally outperform computers by using their associative thinking, such as understanding speech and recognising images, could benefit significantly from using an associative model.

When we started out discussing these paradoxes, we mentioned the challenge of building a bridge between the associative language of the nervous system that humans rely on and the logical language of mathematics by which computers operate. Despite the steps that still need to be taken, we have successfully modelled a first version of the language of the nervous system in the language of mathematics, using the insights gained from the paradoxes mentioned earlier. With this, we believe that our research has brought us one step closer to our bridge building goal. The other side of the river is now well within sight.

Appendix A

Glossary

Terms in this glossary are described as they are used in the context of this work.

Activation Pattern The output of an associative network. An activation pattern contains concepts and their activation values, with high values representing a close relationship between a document and that concept. It is created through *Association Concentration* from the bag of words extracted from the document.

Aristotle's Universals Aristotle interpreted concepts as consisting of the experiences individuals have with specific instances of those concepts [Scaltsas, 1994]. This contrasts with *Plato's Universals*, a view in which concepts exist independent of their instances in the real world. The two interpretations represent opposing approaches to the *Problem of Universals*.

Association Concentration Association concentration works by activating an associative network with a bag of words as input. This activation spreads through the network in such a way that the concepts that are most related to the input document receive the highest activation value. Two algorithms to calculate association concentration are *Flow Activation* and *Spreading Activation*. They are described in Chapter 3 and examined further in Chapter 6.

Associative Network A network of concepts represented by lemmas in which each concept is linked to concepts that are semantically similar to it [Jackoway, 1984].

Back-Propagation A method of training associative networks based on the technique by the same name that is normally used for neural networks [Rumelhart et al., 1988]. This training method is described in more detail in Chapter 6.

Bag of Words The representation of a document as an unordered set of words (disregarding word order and grammar) which indicates how frequently each word occurs in that document [Harris, 1954].

Categorization A specific type of *Document Grouping*. Categorization does not require predetermined classes. Instead, a structure for the documents has to be created from scratch, with documents that cover similar topics being grouped together.

Classification A specific type of *Document Grouping*. Document classes are generated as part of the grouping task.

Concept The mental construct corresponding to an object, action or idea that is expressed by a word. For clarity, concepts are marked with SMALL CAPS.

Conceptual Distance A value expressing the semantic distance between concepts. For example, the conceptual distance between the concept CHAIR and the concept BENCH is smaller than the conceptual distance between HAPPINESS and CARDBOARD. Conceptual distance is represented in associative networks as the value of the edge between two terms. If this value is set in such a way that the associative network produces the best results for the given task, it is called the *Most Effective Conceptual Distance*.

Document A text in a single language concerning a certain topic. Examples of documents are articles in Wikipedia, or files in an information system.

Document Distance The distance between two documents, calculated based on the *Activation Patterns* for each document. The method by which this distance is calculated is described in Chapter 3.

Document Grouping The process of organizing a collection of documents in such a way that documents that cover similar topics are grouped together. Grouping can be done through *Categorization* or *Classification* and the groups created may or may not have a hierarchical structure.

Document Pre-processing The process of extracting the raw text from a document, removing any meta-data and formatting information to most accurately represent the actual text in the document. This process, described in Chapter 2, takes place at the very start of the process of grouping documents using associative networks.

Dynamic Data Libraries of documents in which documents are frequently added, removed or changed. This dynamic would potentially affect the output of document grouping for the library.

Flow Activation A method for calculating *Association Concentration* in which activation is reduced at each node from which it spreads. The details of the algorithm are described in Chapter 3 and it has been tested against *Spreading Activation* in Chapter 6.

Heraclitus' River Heraclitus, a Greek philosopher, famously stated that *"No man ever steps in the same river twice"* [Graham, 2011]. He argued that the river is ever changing and never remains the same, therefore it is never the same river.

Hierarchical Grouping A method of *Document Grouping* in which the groups are structured into a hierarchy, with groups containing other groups.

Language of Mathematics This term originates from the citation *"When we talk mathematics, we may be discussing a secondary language built on the primary language of the nervous system."* by John von Neumann, as quoted by Oxtoby et al. [1958]. It refers to

the logical structure behind mathematics, and the way in which computers process information based upon that structure. It is contrasted to the associative *Language of the Nervous System*.

Language of the Nervous System This term originates from the citation *"When we talk mathematics, we may be discussing a secondary language built on the primary language of the nervous system."* by John von Neumann, as quoted by Oxtoby et al. [1958]. It refers to the associative way of reasoning by humans. It is contrasted to the logic-based *Language of Mathematics*.

Lemma The canonical or dictionary form of a word.

Library A collection of documents that may contain documents in a single language or documents in multiple different languages (see Chapter 10).

Montessori Training A method of training associative networks inspired by the ideas of Maria Montessori for teaching children [Montessori, 1909]. It involves specifically prepared training data (rather than sample documents) and is described in more detail in Chapter 6.

Most Effective Conceptual Distance See *Conceptual Distance*.

Natural Language Parser A software program that determines the grammatical structure of sentences. In our research (see Chapter 7) we use the Stanford Natural Language Parser [Klein and Manning, 2003].

Out-of-Vocabulary Words Words which are present in a document, but which have no equivalent representation within the associative network analysing that document. In Chapter 3, we describe how we deal with out-of-vocabulary words in our experiments.

Pagelink A full-service software house from Hengelo (The Netherlands) that designs and implements applications, and creates strategic Internet solutions for businesses. Pagelink has kindly sponsored the research reported in this thesis.

Pagelink Knowledge Centre A system developed for *Pagelink* that supports the organisation of document collection using associative networks. Known in Dutch as the *'Kenniscentrum'*.

Part-of-Speech Tagger A program that marks words in a text as corresponding to a particular lexical class such as Noun, Verb, or Adjective. In our research (see Chapter 7) the Stanford Part-of-Speech tagger has been used [Toutanova et al., 2003].

Plato's Universals Plato interpreted concepts as existing independent of the actual objects that exhibit them and believed that objects could be described in an absolute form by their pure properties [Klima, 2013]. This contrasts with *Aristotle's Universals*, a view in which the concepts consist of experiences individuals have with specific instances of those concepts. The two interpretations represent opposing approaches to the *Problem of Universals*.

Power Graph Analysis A method of simplifying complex graphs originally introduced to help analyse protein networks [Royer et al., 2008].

Problem of Universals The question whether properties of concepts, such as BROWN or WARM, actually exist, and if so, what they are. Two possible answers to the problem of universals are *Aristotle's Universals* and *Plato's Universals*. The Problem of Universals is discussed in Chapters 1 and 14.

Quick Scan A method for quickly examining an associative network manually, and for determining if the values and relationships stored within the network are conceptually sound. The goal of such an examination is not to verify the validity of the entire network, but rather to take one or two samples to get a generic impression of the quality and

check whether a source or training method is producing sensible results. We describe how to perform a quick scan in Chapter 9.

Ship of Theseus Paradox This paradox considers a ship, once owned by Theseus, which over time has its individual planks replaced until no original part remains, asking whether the ship remains the Ship of Theseus despite sharing no part in common with the original vessel, or if it does not remain the Ship of Theseus, at which point it changed from one thing into another [Plutarch, 75].

Sorites Paradox The sorites paradox is a paradox that arises from vague predicates. The paradox describes a heap of sand, and posits that one can remove a single grain of sand without changing it from a heap of sand into something else. The logical consequence is that if all but one of the grains of sand in the heap are removed, the final singular grain would still form a heap.

Spreading Activation A method for calculating *Association Concentration* in which spreading continues from a certain node only if it has a minimum activation value. The details of the algorithm are described in Chapter 3 and it has been tested against *Flow Activation* in Chapter 6.

Surface Form In the context of this thesis, surface form is referring to a combination of letters representing a *Lemma* or *Word* as it is found in a text. A word may have multiple surface forms (such as the various conjugations of a verb) and a specific combination of letters may be a surface form for more than one lemma (homography).

Word A combination of letters representing a *Concept*. For clarity, instances of words are marked with *italics*. Words may have multiple *Surface Forms*.

WordNet A lexical database of the English language [Miller, 1995, Fellbaum, 1998], used as a source for creating associative networks (see Chapter 5).

Bibliography

D. Adams and M. Carwardine. *Last Chance To See*. William Heinemann Ltd., 1990.

R. Ahuja, T. Magnanti, and J. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

J. Algeo and T. Pyles. *The Origins and Development of the English Language*. Wadsworth Cengage Learning, 2009.

B. Andreopoulos, A. An, X. Wang, M. Faloutsos, and M. Schroeder. Clustering by common friends finds locally significant proteins mediating modules. *Bioinformatics*, 23, 2007.

C. Apte, F. Damerau, and S. Weiss. Towards language independent automated learning of text categorisation models. In *Research and Development in Information Retrieval*, 1994.

C. Apte, F. Damerau, and S. Weiss. Text mining with decision trees and decision rules. In *Proceedings of the Conference on Automated Learning and Discorery, Workshop 6: Learning from Text and the Web*, 1998.

Aristotle. Physics ii, 350 BCE.

C. Babbage. *Passages from the life of a philosopher*. Longman, Green, Longman, Roberts, & Green, 1864.

G. Bader and C. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4, 2003.

S. Bang, J. Yang, and H. Yang. Hierarchical document categorization with k-NN and concept-based thesauri. *Information Processing and Management*, 42, 2006.

S. Bassnett. *Translation Studies*. Methuen, 1980.

B. Basu. The Google story. *South Asian Journal of Management*, 14, 2007.

W. Bechtel. Connectionism and the philosophy of mind: an overview. *The Southern Journal of Philosophy*, 26, 1988.

R. Bekkerman and J. Allan. Using bigrams in text categorization. *Department of Computer Science, University of Massachusetts, Amherst*, 1003, 2004.

N. Bel, C. Koster, and M. Villegas. Cross-lingual text categorization. In *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, 2003.

Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2, 2009.

BIPM. International vocabulary of metrology: basic and general concepts and associated terms, 2008.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of machine learning research*, 3, 2003.

N. Bloom. Applying power graph analysis to weighted graphs. In *Proceedings of the 34th European conference on Advances in Information Retrieval*. Springer Berlin Heidelberg, 2012a.

N. Bloom. Using natural language processing to improve document categorization with associative networks. In *Proceedings of the 17th international conference on Applications of Natural Language Processing and Information Systems*, 2012b.

N. Bloom, M. Theune, and F. de Jong. Hierarchical document categorization using associative networks. In *Proceedings of the 12th IASTED International Conference on Artificial Intelligence and Applications*, 2013a.

N. Bloom, M. Theune, and F. de Jong. Using Wikipedia with associative networks for document classification. In *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013b.

N. Bloom, M. Theune, and F. de Jong. Document categorization using multilingual associative networks based on Wikipedia. In *Proceedings of the 1st International Workshop on Multilingual Web Access, WWW 2015 Companion*. ACM, 2015.

S. Bochner and J. Jones. *Child language development: learning to talk*. Wiley, 2008.

F. Bond and P. Kyonghee. A survey of wordnets and their licenses. In *Proceedings of the 6th Global WordNet Conference*, 2012.

G. H. Bower. Mood and memory. *American psychologist*, 36, 1981.

E. Bray. Will the real Sugababes please stand up?, August 2012. URL `http://www.independent.co.uk/arts-entertainment/music/features/will-the-real-sugababes-please-stand-up-8001732.html`.

R. Browne. *Objects of special devotion: fetishism in popular culture*. Popular Press, 1982.

A. Bryson and Y. Ho. *Applied optimal control: optimization, estimation, and control*. Xerox College Publishing, 1969.

L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on information and knowledge management*, 2004.

A. Calaprice and A. Einstein. *The New Quotable Einstein*. Princeton University Press, 2005.

S. Chakraborti, R. Lothian, N. Wiratunga, A. Orecchioni, and S. Watt. Fast case retrieval nets for textual data. In *Advances in Case-Based Reasoning*. Springer Berlin Heidelberg, 2006.

Chancellor Gorkon. In *Star Trek VI: The Undiscovered Country*, 2293.

C. Chen, H. Lee, and C. Hwang. A hierarchical neural network document classifier with linguistic feature selection. *Applied Intelligence*, 23, 2005.

P. Churchland. *Plato's camera: how the physical brain captures a landscape of abstract universals*. MIT Press, 2012.

R. Cignoli, I. D'Ottaviano, and D. Mundici. *Algebraic foundations of many-valued reasoning*. Kluwer Academic Dordrecht, 2000.

W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17, 1999.

M. Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, 1996.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20, 1995.

F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11, 1997.

W. Daelemans and A. van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. Van den Bosch. Timbl: Tilburg memory-based learner. *Introduction of Linguistic Knowledge*, 10, 2003.

S. D'Alessio, K. Murray, R. Schiaffino, and A. Kershenbaum. Category levels in hierarchical text categorization. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, 1998.

T. H. Davenport and L. Prusak. *Working knowledge: how organizations manage what they know*. Harvard Business Press, 1998.

G. de Melo and S. Siersdorfer. Multilingual text classification using ontologies. In *Advances in Information Retrieval*. Springer Berlin Heidelberg, 2007.

F. Debole and F. Sebastiani. An analysis of the relative hardness of Reuters-21578 subsets. *Journal of the American society for information science and technology*, 56, 2004.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Sciences*, 41, 1990.

E. Dijkstra. How do we tell truths that might hurt? In *Selected Writings on Computing: A personal Perspective*. Springer New York, 1982.

C. Ding and X. He. Cluster merging and splitting in hierarchical clustering algorithms. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, 2002.

X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.

A. Dragland. *Big data - for better or worse*. SINTEF Press Release, 2013.

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on information and knowledge management*, 1998.

M. Ehrig. *Ontology alignment: bridging the semantic gap*. Springer, 2006.

J. Ekedahl. Danish natural language processing in automated categorization, 2008.

V. Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 2002.

European Commission. Europeans and their languages. *Special Eurobarometer 386*, 2012.

P. B. Faber and R. M. Usón. *Constructing a lexicon of English verbs*. Walter de Gruyter, 1999.

Fall-from-Grace. In *Planescape: Torment*, 1999.

C. Fellbaum. *WordNet: an electronic lexical database (language, speech, and communication)*. The MIT Press, 1998.

S. Ferilli, M. Biba, T. Basile, and F. Esposito. Using explicit word co-occurrences to improve term-based text retrieval. In *Digital Libraries*, 2010.

N. V. Findler. *Associative networks: representation and use of knowledge by computers*. Academic Press, 1979.

L. Ford and D. Fulkerson. *Flows in networks*. Princeton University Press, 1962.

G. Forman and H. Suermondt. Hierarchical categorization method and system with automatic local selection of classifiers, March 2008.

C. Fox. A stop list for general text. *SIGIR Forum*, 24, 1989.

F. Fukumoto and Y. Suzuki. Cluster labelling based on concepts in a machine-readable dictionary. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011.

M. Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on computational linguistics*, 2004.

J. Goguen. The logic of inexact concepts. *Synthese*, 19, 1969.

S. Gottwald. *A treatise on many-valued logics*. Research Studies Press Baldock, 2001.

D. W. Graham. Heraclitus. In *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, 2011.

J. S. Gruber. *Studies in lexical relations*. PhD thesis, Massachusetts Institute of Technology, 1965.

S. Gupta, G. Kaiser, M. Grimm, P.and Chiang, and J. Starren. Automating content extraction of html documents. *World Wide Web*, 8, 2005.

B. Habert, G. Adda, M. Adda-Decker, P. B. de Marëuil, S. Ferrari, O. Ferret, G. Illouz, and P. Paroubek. Towards tokenization evaluation. In *The first international conference on Language Resources and Evaluation*, 1998.

M. Haines. *Distributed runtime support for task and data management*. PhD thesis, Colorado State University, 1993.

P. Hájek. *Metamathematics of fuzzy logic*. Springer, 1998.

M. Halliday. *Spoken and written language*. Oxford University Press, 1989.

P. Halmos. The legend of John von Neumann. *American Mathematical Monthly*, 4, 1973.

X. Han, S. Li, and Z. Shen. A k-NN method for large scale hierarchical text classification at LSHTC3. In *Large Scale Hierarchical Text Classification Challenge 3*, 2012.

D. Harper. Online etymology dictionary. September 2013. URL `http://dictionary.reference.com/browse/schadenfreude`.

Z. Harris. Distributional structure. In *Papers in Structural and Transformational Linguistics*. D. Reidel Publishing Company, 1954.

C. Havasi, R. Speer, and J. Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing*, 2007.

F. Herbert. *Dune*. Chilton Books, 1965.

D. Hiemstra. *Using language models for information retrieval*. Taaluitgeverij Neslia Paniculata, 2001.

H. Hodson. Google's fact-checking bot builds vast knowledge bank. *New Scientist*, 2983, August 2014. URL `http://www.newscientist.com/article/mg22329832.700-googles-factchecking-bots-build-vast-knowledge-bank.html`.

J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79, 1982.

G. Jackoway. *Associative Networks on a Massively Parallel Computer*. Duke University, 1984.

K. Jayanthi, S. Chakraborti, and S. Massie. Introspective knowledge revision in textual case-based reasoning. In *Case-Based Reasoning. Research and Development*. Springer Berlin Heidelberg, 2010.

G. Jeschke and M. Lalmas. Hierarchical text categorisation based on neural networks and Dempster-Shafer theory of evidence. *EUROFUSE Workshop on Information Systems*, 2002.

T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, 1998.

D. Jurafsky and J. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, 2009.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20, 1998.

J. Keller, M. Gray, and J. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15, 1985.

H. Kimoto and T. Iwadera. Construction of a dynamic thesaurus and its use for associated information retrieval. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, 1990.

S. Kiritchenko. *Hierarchical Text Categorization and its Applications to Bio-informatics*. PhD thesis, University of Ottowa, Canada, 2006.

D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, 2003.

G. Klima. The medieval Problem of Universals. In *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, 2013.

V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

LazyTown's Stephanie. Step by step song. In *Episode 7: 'Hero for a day'*, LazyTown, 2004.

C. Lee and H. Yang. Construction of supervised and unsupervised learning systems for multilingual text categorization. *Expert Systems With Applications*, 36, 2009.

D. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38, 1995.

D. Lewis. Naive (Bayes) at forty: the independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, 1998.

D. Lewis. Reuters-21578 text categorization test collection, distribution 1.0, readme file (v1.3), May 2004. URL `http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt`.

T. Li, S. Zhu, and M. Ogihara. Hierarchical document classification using automatically generated hierarchy. *Journal of Intelligent Information Systems*, 29, 2007.

W. Lidwell, K. Holden, and J. Butler. *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions,*. Rockport Publishers, 2010.

N. Machiavelli. *The Prince*. Antonio Blado d'Asola, 1532.

P. Malo, P. Siitari, and A. Sinha. Automated query learning with wikipedia and genetic programming. *Artificial Intelligence*, 194, 2010.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *AAAI-98 workshop on learning for text categorization*, 1998.

Merriam-Webster. *association*. 2014. URL `http://www.merriam-webster.com/dictionary/association`.

R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *North American Chapter of the Association for Computational Linguistics*, 2007.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ArXiv e-prints*, 2013.

G. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38, 1995.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3, 1990.

R. Mitkov. *The Oxford Handbook of Computational Linguistics*. OUP Oxford, 2005.

M. Montessori. *The Montessori method: scientific pedagogy as applied to child education in 'the children's houses' with additions and revisions by the author*. Frederick A Stokes Company, New York, 1909.

R. Munroe. Useless. *xkcd.org*, January 2006. URL `http://xkcd.com/55/`.

P. Nakov, E. Valchanova, and G. Angelova. Towards deeper understanding of the lsa performance. In *Proceedings of recent advances in natural language processing*, 2003.

N. Nanas and A. Roeck. Autopoiesis, the immune system, and adaptive information filtering. *Natural Computing*, 8, 2009.

V. Nastase and M. Strube. Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194, 2013.

F. Nes, T. Abma, H. Jonsson, and D. Deeg. Language differences in qualitative research: is meaning lost in translation? *European Journal of Ageing*, 7, 2010.

I. Newton. *The Mathematical Principles of Natural Philosophy*. Edmond Halley, 1687.

C. Ni, J. Sun, J. Hu, and Z. Chen. Cross lingual text classification by mining multilingual topics from Wikipedia. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011.

V. Novák, I. Perfilieva, and J. Močkoř. *Mathematical Principles of Fuzzy Logic*. Kluwer, Boston, 1999.

J. Oxtoby, B. Pettis, and G. Price. *John Von Neumann, 1903-1957*. American Mathematical Society, 1958.

R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131, 2003.

P. Pirolli, J. Pitkow, and B. Ramana. System for predicting documents relevant to focus documents by spreading activation through network representations of a linked collection of documents, November 1998.

Plato. *Cratylus*. 350 BCE.

L. Plutarch. Theseus, 75.

M. Prawdin, E. Paul, and C. Paul. *The Mongol empire: its rise and legacy*. G. Allen and Unwin, Limited, 1940.

R. Quillian. Semantic memory. In *Semantic Information Processing*, 1968.

W. Quine. Main trends in recent philosophy: Two dogmas of empiricism. *The philosophical review*, 60, 1951.

W. Quine. *Word and Object*. MIT Press, 1964.

W. Quine. What price bivalence? *The Journal of Philosophy*, 78, 1981.

J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

R. Quirk. *A Comprehensive grammar of the English language*. Longman, 1985.

Rainbow Dash. Art of the dress song. In *Episode 14: 'Suited for Success'*, My Little Pony: Friendship is Magic, 2011.

Ravel Puzzlewell. In *Planescape: Torment*, 1999.

Y. Ravin and C. Leacock, editors. *Polysemy: theoretical and computational approaches*. OUP Oxford, 2000.

J. Reynar. Topic segmentation: algorithms and applications. *IRCS Technical Reports Series*, 1998.

L. Rigutini, M. Maggini, and B. L. An EM based training algorithm for cross-language text categorization. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 2005.

J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. Prentice Hall, 1971.

L. Rohde. A case study in literary translation. *Translatio*, 2, 2011.

L. Royer, M. Reimann, B. Andreopoulos, and M. Schroeder. Unraveling protein networks with power graph analysis. *PLoS Computational Biology*, 4, 2008.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.

B. Russell. Vagueness. *The Australasian Journal of Psychology and Philosophy*, 1, 1923.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24, 1988.

Y. Sasaki and D. Weissenbacher. TTI's system for the LSHTC3 challenge. In *Large Scale Hierarchical Text Classification Challenge 3*, 2012.

T. Scaltsas. *Substances and Universals in Aristotle's Metaphysics*. Cornell University Press, 1994.

R. C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, 1982.

R. C. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Association, 1977.

E. Schubert. Enjoyment of negative emotions in music: an associative network explanation. *Psychology of music*, 24, 1996.

R. W. Schvaneveldt. *Pathfinder associative networks: studies in knowledge organization.* Ablex Publishing, 1990.

F. Sebastiani. Text categorization. *Text Mining and its Applications*, 2005.

S. Shehata. *Concept Mining: A Conceptual Understanding based Approach*. PhD thesis, University of Waterloo, 2009.

T. Sider. *Four-dimensionalism: An Ontology of Persistence and Time*. Clarendon Press, 2003.

A. Singhal. Introducing the knowledge graph: things, not strings. *Official Google Blog*, 2012. URL `http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html`.

P. Soucy and G. Mineau. Beyond TFIDF weighting for text categorization in the vector space model. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005.

R. Stanfel. Volkswagen of America, inc.: Fahrvergnügen campaign. In *Encyclopedia of Major Marketing Campaigns*, 2000.

C. Stifter, S. Anzman-Frasca, L. Birch, and K. Voegtline. Parent use of food to soothe infant/toddler distress and child weight status. an exploratory study. *Appetite*, 57, 2011.

Sun Tzu. The art of war, 500 BCE.

J. Surowiecki. *The Wisdom of Crowds*. Knopf Doubleday Publishing Group, 2005.

I. Swartjes, J. Vromen, and N. Bloom. Narrative inspiration: using case based problem solving for emergent story generation. In *Proceedings of the Fourth International Joint Workshop on Computational Creativity*, 2007.

D. Tikk, J. Yang, and S. Bang. Hierarchical text categorization using fuzzy relational thesaurus. *Kybernetika*, 39, 2003.

J. Tolkien. *The lord of the rings: the fellowship of the ring*. George Allen & Unwin, 1954.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003.

J. Trier, A. van der Lee, and O. Reichmann. *Aufsätze und Vorträge zur Wortfeldtheorie*. De Gruyter, 1973.

S. Turner. *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum Associates, 1994.

E. Turunen. *Mathematics Behind Fuzzy Logic*. Physica-Verlag Heidelberg, 1999.

A. Vallabhaneni, T. Wang, and B. He. Brain-computer interface. In *Neural engineering*. Springer, 2005.

C. Von Clausewitz. *Vom Kriege*. Ferdinand Dummler, 1832.

P. Vossen, K. Hofmann, M. de Rijke, E. Sang, and K. Deschacht. The Cornetto database: architecture and user-scenarios. In *Proceedings of 7th Dutch-Belgian Information Retrieval Workshop*, 2007.

L. Vygotsky, E. Hanfmann, G. Vakar, and A. Kozulin. *Thought and Language*. MIT Press, 2012.

B. Wanders and S. Te Brinke. Strata: typed semi-structured data in dokuwiki. In *Proceedings of the 10th International Symposium on Open Collaboration*, 2014.

B. Wanders, M. Van Keulen, and P. E. Van der Vet. Uncertain groupings: probabilistic combination of grouping data. Technical report, 2014.

X. Wang, H. Zhao, and B. Lu. A meta-top-down method for large-scale hierarchical classification. In *Large Scale Hierarchical Text Classification Challenge 3*, 2012.

C. Wartena and R. Brussee. Topic detection by clustering keywords. In *19th International Workshop on Database and Expert Systems Application*, 2008.

Z. Weber and M. Colyvan. A topological sorites. *The Journal of Philosophy*, 107, 2011.

E. W. Weisstein. Cube. *MathWorld – A Wolfram Web Resource*, February 2015. URL `http://mathworld.wolfram.com/Cube.html`.

A. Wichert. Hierarchical categorization. In *Ninth Midwest Artificial Intelligence and Cognitive Science Conference*, 1998.

P. Wiemer-Hastings. Latent semantic analysis. In *Encyclopedia of Language and Linguistics*. Elsevier, 2004.

Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1, 1999.

K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18, 1989.

C. Zhong, D. Miao, and P. Franti. Minimum spanning tree based split-and-merge: a hierarchical clustering method. *Information Sciences*, 181, 2011.