UNIVERSITY OF TWENTE.

Discovering groups using short messages from social network profiles

Using hierarchically organized concepts to discover relations

Master Thesis of Tom Palsma
Computer Science,
Track Information Systems Engineering
Enschede, July 14, 2010

Supervisors University of Twente
Maurice van Keulen
Dolf Trieschnigg

Supervisors Topicus FinCare
Jasper Laagland
Wouter de Jong

Discovering groups using short messages from social network profiles

Using hierarchically organized concepts to discover relations

Tom Palsma

July 14, 2010

Samenvatting

In de afgelopen paar jaar zijn mensen het internet steeds meer gaan gebruiken voor het delen van foto's, gedachten en de activiteiten waar ze mee bezig zijn via foto-albums, gebruikersprofielen, blogs en korte tekstberichten in online sociale netwerksites, zoals Facebook en Hyves. Deze informatie kan erg interessant zijn voor (persoonlijke) marketing- en advertentiedoeleinden. Naast de groepen die gebruikers van netwerksites zelf vormen, bestaan er ook relaties tussen mensen gebaseerd op gemeenschappelijke interesses, die impliciet worden beschreven in korte tekstberichten.

Dit onderzoek richt zich op het ontdekken van groepen op basis van semantische relaties in korte tekstberichten van profielen, en het beschrijven van de kenmerken van de groepen en relaties. Omdat deze relaties door het vergelijken van gemeenschappelijke woorden in berichten vaak niet goed te ontdekken zijn, gebruiken we een hiërarchische conceptenstuctuur verkregen van het Wikipedia categoriesysteem om groepen en relaties op abstractere conceptuele niveaus te kunnen ontdekken op basis van korte tekstberichten uit Twitter-profielen.

We gebruiken een simpele classificatie methode die niet gebonden is aan een specifieke verzameling van concepten. Concepten (en daaraan gerelateerde hoger gelegen concepten) worden aan profielen gekoppeld als het concept, of de daaraan gerelateerde woorden, voorkomen in een kort bericht van het profiel. Handmatige evaluatie van deze manier toont aan dat 37.4 % van de koppelingen juist zijn (precisie), wat resulteert in een F-score van 0.54.

Om de precisie van het classificeren te verbeteren gebruiken we Support Vector Machines. Met behulp van statistieken over eigenschappen van de conceptenstructuur en de koppeling tussen het concept en het profiel kunnen de resultaten verbeterd worden met 14 % gemeten volgens de F-score (0.68).

Het groeperen bestaat uit het clusteren van gegevens gebaseerd op hoe vaak de concepten voorkomen in profielen. Interessante groepen zijn clusters van concepten die niet samen voorkomen in de originele Wikipedia categoriestructuur. Naast dit soort groepen laten de resultaten ook groepen zien met concepten die wel een semantische relatie hebben, waarbij die relatie niet voorkomt in de Wikipedia categoriestructuur. Deze informatie zou gebruikt kunnen worden om de structuur te verbeteren.

Het proces toont aan dat het gebruik van een hiërarchische conceptenstuctuur kan helpen om groepen te ontdekken op basis van semantische relaties op abstractere conceptuele niveaus. De selectie van concepten die worden toegewezen aan profielen kan helpen om de te ontdekken groepen naar bepaalde gewenste domeinen te leiden. Echter door classificatiefouten, veroorzaakt door onder andere meerdere betekenissen van concepten, zou een betere methode om concepten aan profielen te koppelen de kwaliteit van de ontdekte groepen nog wel kunnen verbeteren. Of de ontdekte groepen ook bruikbaar zijn voor marketing- en advertentiedoeleinden zal verder onderzoek moeten uitwijzen.

Summary

In the past few years people used the internet more and more for sharing their photos, thoughts and activities via photo albums, user profiles, blogs and short text messages on online social networking sites, like Facebook and MySpace. This information could be very useful for (personal) marketing and advertising. Besides groups formed by the users of the social network sites explicitly, people could have a relation based on similar interests that they implicitly leave in short text messages.

This research has a focus on discovering groups, taking into account semantic relations between user profiles and describing the characteristics of the groups and relations. Because it is hard to discover these types of relations at word level by matching similar words in messages, we introduce a hierarchical structure of concepts obtained from the Wikipedia category system to discover groups and (semantic) relations at more abstract conceptual levels between profiles with short text messages obtained from Twitter.

In order to provide a general approach that is not limited to a specific set of concepts we use a naive classification approach. Concepts (and their parent concepts) are assigned to profiles when concept (related) terms occur in a short message of the profile. Manual evaluation of this approach shows 37.4 % of the assignments is correct (the precision), which results in an F-score of 0.54.

To improve the precision of the classification results we use Support Vector Machines. Using features related to characteristics of the concept structure and the relations between concepts and profiles improves the classification results with 14 % according to the F-score (0.68).

The grouping process consists of clustering of statistical data of concept occurrences in user profiles. Interesting groups discovered based on the clustering results are groups of concepts that are not grouped together in the original Wikipedia category structure. Besides these types of groups the results also show groups of concepts that have a semantic relation, which is not reflected in the Wikipedia category structure. This information could be used to improve the Wikipedia category structure.

The overall process shows that the usage of hierarchical concepts and clustering helps to discover groups based on semantic relations on abstract conceptual levels. The selection of concepts and assigning them to user profiles could guide the grouping results to desired domains and the concepts help to describe the groups. However, due to problems with ambiguous meaning of concepts and characteristics of the messages, another approach of assigning concepts could improve the quality of the discovered groups. To know how useful the groups are for marketing and advertising requires more research.

Voorwoord

Voor u ligt de afstudeerscriptie van Tom Palsma. De afstudeeropdracht is ter afronding van de master Computer Science, aan de Universiteit Twente. Het onderzoek heeft plaatsgevonden bij Topicus FinCare te Deventer, waar ik met plezier aan dit onderzoek heb gewerkt.

Maurice en Dolf, de begeleiders namens de Universiteit Twente, bedankt voor jullie expertise, suggesties en feedback tijdens het onderzoek. Daar heb ik veel van geleerd.

Ook wil ik Jasper en Wouter, mijn begeleiders namens Topicus, bedanken voor het helpen opzetten van de opdracht en de begeleiding gedurende de hele opdracht. Daarnaast wil ik alle collega's van Topicus FinCare bedanken voor hun interesse en de leuke afstudeerperiode.

In het bijzonder bedank ik Rike en mijn ouders voor hun steun en vertrouwen. Tot slot bedank ik ook mijn vrienden waar ik de afgelopen jaren een ontzettend leuke studententijd mee heb beleefd.

Tom Palsma,
Enschede, juli 2010

Contents

Chapter 1

Introduction

In the past five years people used the internet more and more for sharing their photos, thoughts and activities via photo albums, user profiles, blogs and short text messages on online social networking sites, like Facebook and MySpace (Figure 1.1). Users of these online communities express themselves by publishing their interests and hobbies. This information could be very useful for (personal) marketing and advertising. In this chapter we introduce our research to the possibilities of discovering groups using short text messages from online social network profiles. The research is motivated in the context of Topicus and in the context of data mining and information retrieval. This is followed by the research goals and scope and the research questions.

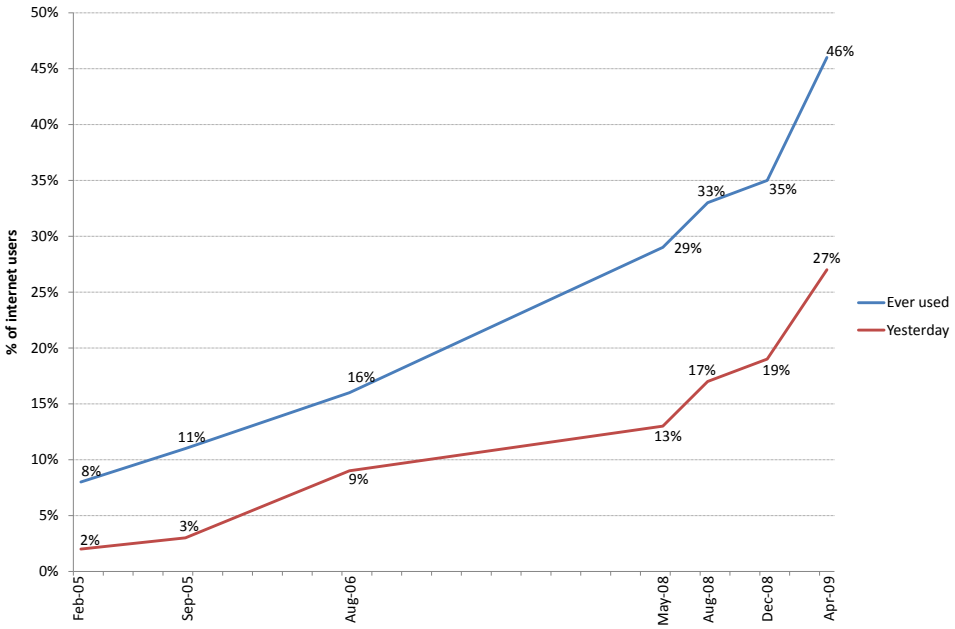

Figure 1.1: Growth in social networking sites used by adults (2005-2009) [35]

1.1 Motivation

1.1.1 Topicus

This research is carried out at Topicus FinCare B.V., a software company located in Deventer, The Netherlands. Customers of Topicus FinCare are health care insurance companies. Health care insurance companies often create several insurance policies with different coverage (product differentiation). The coverage is most of the time related to different age groups like students, parents with young children and seniors. For example, a policy for students has coverage for physical therapy and vaccines, but not for baby care. The insurance companies also sell collective policies, subscribed by one or more companies, associations or other groups that have something in common. For marketing purposes, discovering (new) target markets and target groups is an interesting subject for Topicus and their customers. Information available in online social networking sites could help to achieve this.

1.1.2 Information in online social networking sites

From the perspective of marketing, advertisement and product development it is interesting to discover groups of people that have something in common, for example age (range) or gender. For advertising campaigns and product development, discovering the appropriate target markets and audience is an important stage in the market(ing) research.

Finding (new) target groups and analyzing potential customers is hard without knowing these people. Market researchers often use statistical surveys, like questionnaires, to gain information about people, such as age, gender and interests. However, the quality of the results depends on the number of respondents and the completeness of the surveys. In addition, surveys with closed ended questions will result in statistical information that is limited to the pre-defined answers in the surveys. This will not discover new relation between concepts (e.g. hobbies, taste in music, gender) that do not occur in the questions and answers of the survey.

Nowadays, people share their interests, hobbies, marital status, religion, location and a lot of other information with other people on the internet on online social networking sites, such as Facebook, MySpace, Twitter, Hyves, etc. Users of these social networking sites build a profile of themselves and leave this information in predefined labeled text fields on their profile on the site. This personal information is visible for contacts in the network and often for other users on the site. However, there is no information about when the last update of the text field was or how important an interest is to the user. Studies showed that people are not always willing to explicitly specify or update their interests[38], which make sources where people leave this information implicit more useful. In text messages from the user to other users or the public, people leave this information implicitly, but that is not labeled. Besides the labeled fields, sharing photos, sending private messages, connecting to friends, most online social networking sites (OSNS's) also provide a communication platform via short messages. Twitter is an OSNS that focuses only on communication via short messages, also called microblogging. In these short messages people describe their current status, daily life activities, short news stories and other

interests [28, 24]. These messages are published from and to the web, mobile phones, instant messages and other OSNS's.

These text messages are unstructured and not labeled; however, the messages are published at a certain point in time. So you know how recent the information is and if multiple messages are about the same subject, it contains implicit information about a user's interests and how important they are to the user.

The problem is how to extract useful information, such as groups of people based on common interests, from the unstructured short messages in OSNS's. Topics in these short messages could be very specific and the messages of a user have many different topics. These topics could be related at a more generic conceptual level. For example, 'Britney Spears' and 'Christina Aguilera' are both 'American female singers' and 'pop singers'. Based on more generic concepts there could be relations between people, which will not be discovered when looking only to the content of the message. Using a hierarchical structure of concepts that contains concepts organized based on conceptual levels, relations between people could be discovered on the different conceptual levels. The user profiles of these people could be (automatically) grouped by similarity of these relations and the groups could be labeled with the concept names, which are closely related to human perceptions.

1.1.3 The use of groups

Discovering groups based on information in short messages published by people on the internet, is useful in the context of marketing and advertising. Traditionally companies do not advertise for make-up in a magazine about soccer, because most of the readers of these magazines are men, while women use make-up. In this case there is a relation between groups based on gender and the use of a product or reading a type of magazine may be quite obvious. We expect that incorporating a hierarchy of concepts by linking concepts to user profiles based on short text messages and creating groups based on the co-occurrences of concepts in user profiles could result in the discovery of new groups. By describing the properties of the groups using the existing (human defined) concepts from the hierarchy, it is possible to show what people in a group have in common and what distinguishes them from other groups.

An example of properties of a group could be that people who often listen to eighties music also read comic books, but are not interested in the Oprah Winfrey show. Companies could use this information when they want to create advertisement campaigns that are not focused on traditional groups based on for example gender or age. The properties of a group could be used to decide whether it is useful to broadcast a commercial for a product during a certain tv show, sending special offers in an e-mail from a webshop, place products in a store near each other, sell the products together or publish advertisements in specific groups or for specific users of the online social networking sites.

Discovered distinct groups are useful if a company wants to reach a wide range of people without reaching the same group of people many times. For example by broadcasting a commercial during the Oprah Winfrey show and during the eighties program on the music channel.

1.2 Terminology

User profile and short messages

In this research, we use terms related to online social networking sites. With a (*user*) *profile*, we mean the webpage or web pages that contain information related to a member of the networking site. A *user* is a person who owns a profile on the social networking site and leaves (personal) information by publishing short messages on his or her profile. These *short messages* are often limited in length and contain information about where a person is, what he or she is doing or what is on the mind of the profile owner. Short messages are also used for communication between users. We call the collection of all short messages published by a user, a *message stream*.

Text mining

Because this research is about transforming unstructured data in short text messages to meaningful information, we operate in the field of *text mining*. In the context of text mining the word *document* refers to the object that contains textual data that is analyzed in text mining tasks [14, 21]. In this research we consider the short messages as documents. A document often has one or more topics: a focus theme or subject what the text is about. A message stream is a collection of documents and (often) has multiple topics.

Categorization, classification and clustering

Categorization is the process of dividing objects into groups of entities whose members are in some way similar to each other [23]. In the context of text mining the objects are text documents and (text) *classification* and (text) *clustering* are typical tasks.

We use these tasks in the process of discovering groups based on short text messages. Classification is the task of assigning documents to one or more predefined categories (or labels). Clustering is the task of grouping objects based on the similarity between the objects. Similar objects will be organized in the same cluster. There are *supervised* and *unsupervised* versions of these tasks. The supervised versions rely on manual creation of training sets, for example by assigning the correct concepts to user profiles by a human.

Hierarchical structure of concepts

Discovering groups based on existing structures plays an important role in the research. We use the term *hierarchical structure of concepts* or *concept hierarchy* for a graph structure where concepts (e.g. categories or subjects) at a higher level have a broader meaning than concepts at lower levels. In Figure 1.2 the concepts 'ball (sports)' and 'precision (sports)' are related to 'sports', in this case it are subsets. Profiles could have relationships with these concepts on all levels of the hierarchy. The difference between concepts and topics is that concepts come from the hierarchy and topics not. A concept related to a document can be more general or abstract than the topic of the document. For example, the text 'I'm playing tennis.' had the topic 'tennis', but could have for example related concepts like 'tennis', 'sport', 'ball sport', 'individual sport' and 'racquet sport'.

Relations between concepts and user profiles

The short messages in user profiles have different topics: the main theme or subject of the message, profiles could have *relations* to multiple concepts. For example, Figure 1.2 shows that John is interested in 'soccer' and in 'curling'. The definition of a *relation* between a *user* and a *concept* based on a message is important. We defined that this relation exists if a user explicitly describes a positive relation between him and the concept once, e.g. 'I play soccer'. The message 'I hate to play soccer' or 'I don't like soccer' describe negative relations between the user and the concept 'soccer' and are not considered as a relation in the first place.

When a user refers to the concepts more than once. For example, when the user John writes two messages: 'Jane plays tennis' and 'I hate tennis', we consider this as a relation of the user John and the concept 'tennis'. This is based on the assumption that (writing) time is a good user implicit interest indicator [5] and that publishing more than one message is a good threshold to measure this.

Semantic relations between users

Based on their messages, users could have *semantic relations* on higher abstraction levels. Figure 1.2 shows John and Jane have relations to different concepts. However, on a higher abstraction levels the concepts are all related to the concept 'Sports', so there is a semantic relation between John and Jane. This is a semantic relation, because based on the meaning of the messages there is a relation, while the text in the message could contain different words or different lower level concepts.

Groups

The research is about discovering groups. With a *group* we mean a set of user profiles of people and the definition of properties that describes what distinguishes the group from other groups. These properties are based on occurrences of the concepts that are related to the user profiles. People can have a relation to the group based on their message stream and the concepts that they have in common with the groups. The *grouping process* consists of text mining tasks and uses short messages from Twitter and a concept hierarchy from Wikipedia as input. Figure 1.3 shows a global overview of this process. We discuss a more detailed overview of the grouping process in Chapter 2.

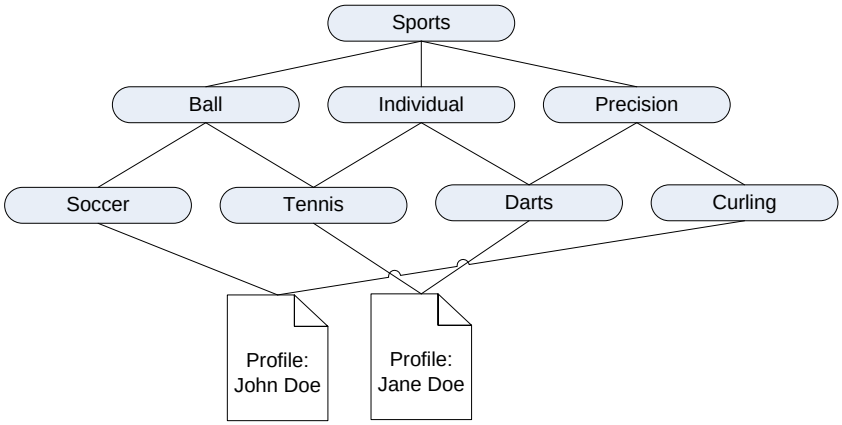

Figure 1.2: Categorization of profiles in concepts related to *sports*

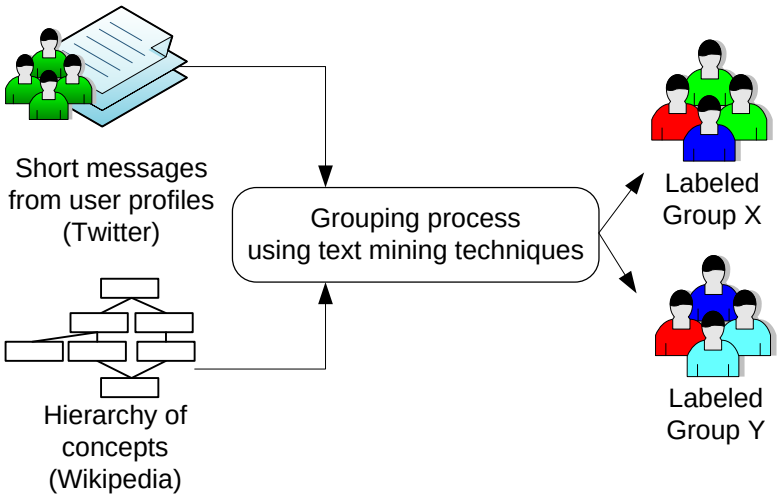

Figure 1.3: Global research overview

1.3 Research

1.3.1 Goal and Scope

The main goal of this research is discovering groups based on short messages published by users at online social network sites. We consider the grouping process as categorization problem of small text documents.

The short messages are often about only one specific topic (e.g. 'tennis') and the different message in a user profile contain multiple topics. In this research we focus on finding semantic relations between user profiles on higher abstraction levels, while the short messages often do not contain abstract concepts explicitly (e.g. 'sport'). To discover groups based on abstract relation between profiles, we use (hierarchical) concepts that are related to user profiles. We consider the assignments of concepts to user profiles based on short messages as a *classification task* and the creating of groups based on the co-occurrences of concepts in user profiles as a *clustering task*. Because we want to deal with large and variable concept structures, manual training is not an option and we focus in this research on an *unsupervised* process.

Concluding, this research focuses on automatic discovering groups (of users) based on short text messages from user profiles using existing hierarchical structure of concepts, to discover relations on different conceptual levels, in the domain of online social networking sites. Other subjects related to classification and clustering of user profiles, like privacy issues and presenting the results in a user-friendly way are beyond the scope of this research.

1.3.2 Research Questions

In the process of discovering information based on texts, also called text mining, different approaches could be applied. Figure 1.4 shows the functional architecture of a text mining system. The process in these kinds of systems is usually the same, while the implementation of the sub processes (pre processing, representation model, strategies, etc.) differ. The research questions address some of the sub processes. The main question is:

Can we automatically categorize and group user profiles from online social networking sites based on the semantic similarity of short text messages using an existing hierarchical structure of concepts?

The following sub questions relate to the main question:

- *Which existing types of hierarchical structures of concepts are according to literature useful for finding relevant groups, taking semantic relation into account?*
- *Which classification strategy is suitable for automatic assigning concepts from a hierarchical structure to user profiles based on short text messages?*
- *How to evaluate the quality of the assignment of concepts to user profiles?*
- *What is the quality of the assignment of concepts to user profiles?*
- *Which clustering strategy is suitable for discovering groups based on hierarchical concepts related to user profiles?*
- *What is the effect of using the hierarchical concepts on the discovered groups?*

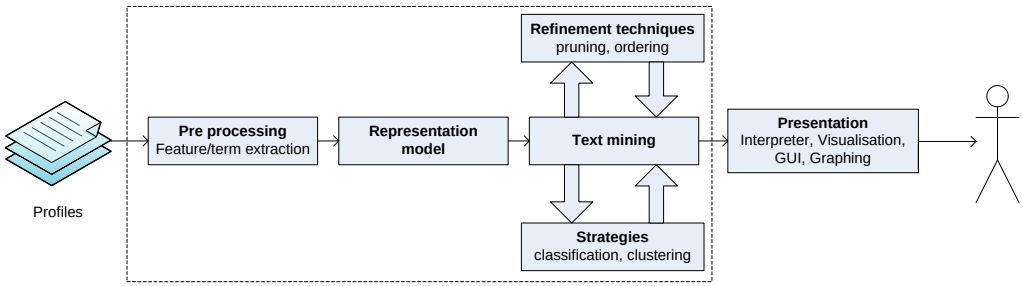

Figure 1.4: Functional architecture of a text mining system [14]

1.3.3 Research Approach

To answer the research questions we follow the steps in the text mining process [14] (Figure 1.4). First, we need to gather data: user profiles with short messages from online social networking sites. According to the existing text mining and information retrieval techniques in literature, we pre-process this data.

The next step is creating a representation model from the pre-processed data. This model should be suitable for storing information from the short messages related to a user profile. Because the focus of this research is on using a hierarchical structure of concepts to discover relations, the model should be capable of discovering relations between concepts and short messages, in order to link the concepts to the profile.

Literature describes a lot of text mining strategies and techniques to discover relations between text documents and how to group them. For example: similarity functions, clustering, classification and ordering algorithms. Based on literature we combine relevant strategies to achieve the goal: automatic discovering groups of users based on short text messages, where a profile can occur in multiple groups.

In the end, the quality of the process is important. Are the profiles enriched with correct hierarchical concepts and are the profiles assigned to relevant groups? In the information retrieval and text mining fields there are several performance measurements available to measure the validity of the result of the text mining process. We discuss which measurements are relevant in this case and apply them to the different results obtained by varying in text mining strategies.

To achieve the assignment of concepts to user profiles and the grouping of concepts and profiles based on these relations and the performance evaluation, we build a prototype. The prototype has four tasks:

- Creating a representation model from the user profiles with short messages.
- Enrich the user profile with information from a hierarchical structure of concepts.
- Evaluate the performance of this process.
- Using clustering strategies to discover (labeled) groups based on the data model containing user profiles enriched with hierarchical concepts.
- Presenting the properties of the discovered groups.
- Evaluate the effect of using the hierarchical structure of concepts on the discovered groups.

Chapter 2 gives a more detailed description of the tasks in the process in the context of text mining.

1.4 Overview

In the next chapter (Chapter 2) we give background information on text mining in the context of this research and the approach. The chapters after that are related to the elements of Figure 1.3. Each chapter contains a section that discusses related work on the elements of the research. First, we introduce the data source of user profiles with short messages from Twitter (Chapter 3) and the hierarchical structure of concepts from Wikipedia (Chapter 4). Chapter 5 describes the assignment of hierarchical concepts to user profiles to discover relations between users on higher conceptual levels in later stages of the process. For validation of this assignment process we use a manual evaluated collection (Chapter 6). This collection is also used to validate the generated classification models using Support Vector Machines and features based on user profile data described in Chapter 7. For the evaluation of the results of the classification of the assignments of concepts to profiles (Chapter 8) we use precision and recall to measure the quality of this process.

Chapter 9 describes the used clustering strategies to discover groups based on the user profile data with concepts and evaluates the effect of using the hierarchical concepts on the discovered groups.

Finally, we draw conclusions and do suggestions and recommendations for future research (Chapter 10).

Chapter 2

Approach

In the first chapter, we introduced the goal and scope of the research together with the approach that we take, to reach the goal and answer the research questions. The main part of the research consists of the grouping process using text mining tasks, as mentioned in Figure 1.3. We incorporate (hierarchical) concepts to discover groups of user profiles based on short text messages. Figure 2.1 gives a more detailed overview of the tasks in this grouping process.

We view the assignments of concepts to user profiles based on short messages as a (text) *classification task* and the creation of groups based on the co-occurrences of concepts in user profiles as a *clustering task*. In this chapter, we motivate in more detail why we use the approach of classification and clustering to discover groups based on short text messages (Section 2.1) and explain these task in more detail (Section 2.2).

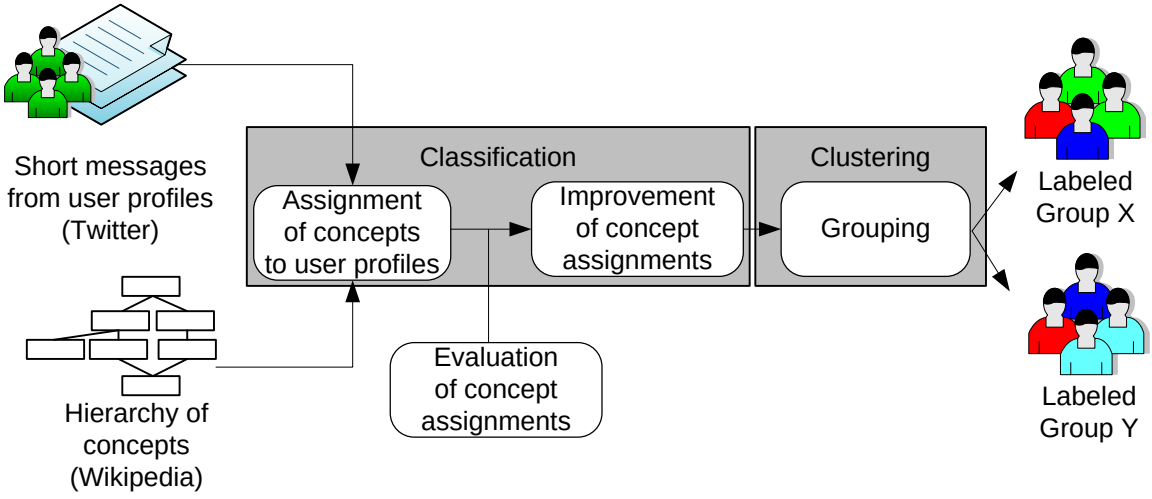

Figure 2.1: Process overview

2.1 Motivation and approach

In Section 1.2 we defined the basic terminology that is used in this research. In this section, we motivate our research approach, by looking in more detail to the characteristics of short messages, hierarchical concepts and text mining tasks.

2.1.1 The challenge of handling short messages

Compared to text documents, such as web pages, short messages contain very few terms. Our tests collections, shows an average of twelve terms per message (Chapter 3). Text mining techniques that are used to discover semantic relations between documents often rely on the assumption that there is similarity between other related terms in the document. However, short message do not contain many of these terms. This makes it hard to discover semantic relations between user profiles based on the short messages.

For example, there is a certain relation between 'soccer' and 'tennis', both are sports. Using word or letter similarity functions will not discover this relation, because the words are very different. In larger text documents these words are probably surrounded by related words, such as 'ball' and 'play', but the texts in short messages often mention only one specific concept.

2.1.2 User profile classification based on a concept hierarchy

To discover groups based on a relation between users at a more abstract conceptual level, we introduce the assignment of hierarchical concepts to user profiles. A concept hierarchy contains more abstract concepts on higher levels in the hierarchy. This information could be used to discover that there is a relation between 'soccer' and 'tennis'.

We use classification techniques to link concepts to user profiles based on short messages. Incorporating the hierarchical information in the concept structure will help to discover relations between the user profiles on more abstract conceptual levels.

2.1.3 Step 1: concept classification of user profiles

Classification techniques use a predefined set of classes and documents will be linked to the existing classes (or categories). Based on similarity between the document and (earlier defined) properties of the classes, documents are assigned to classes. We use the concept hierarchy as the predefined set of classes and the short messages are the documents. The similarity between those two is based on the occurrences of the concept terms in the short messages. Multiple concepts could have a relation to a profile. Besides the related concepts based on matching similar terms, our classification approach assigns the higher level concept in the concept hierarchy also to a profile. This could help to discover the relations between profiles on more abstract conceptual levels. In Section 2.2 we explain our classification approach in more detail.

2.1.4 Step 2: clustering to discover groups

Figure 2.1 shows that grouping is the last part in the text mining process. We use a clustering approach to discover groups of user profiles. A clustering task consists of grouping a collection of objects (documents, profiles, etc.) into meaningful clusters. The objects in a cluster are similar in some sense, however the clusters are not predefined. Clustering algorithms often generate labels from terms that occur frequently in the documents in the cluster to describe the clusters. The result is that these labels differ from real world concepts or categories, because there is does not have to be a (semantic) connection between these frequent terms. A collection of terms is also less descriptive than a predefined concept.

In our approach, the objects are the user profiles with (multiple) linked concepts and the clusters are the groups. Based on similarity of occurrences of the concepts in user profiles, a clustering algorithm could discover the groups. The group is labeled by the concepts that occur in the group. These concepts are more descriptive than a collection of terms that are not connected and intuitively organized in the Wikipedia category structure.

2.2 Text mining tasks in detail

We split up the classification task of assigning concepts to user profiles to discover semantic relations into two separate tasks (Figure 2.1), we refer to these tasks as naive classification (Subsection 2.2.1) and improved classification (Subsection 2.2.2). The grouping process is done by using a clustering algorithm applied to the information of concepts associated with hierarchical concepts (Subsection 2.2.3). This section gives a short introduction the approaches that are used.

2.2.1 Naive classification

Classification techniques often rely on a supervised learning process. During this learning process, training documents are manually assigned to their true class. During the classification process, new documents are assigned to the classes based on the similarity between the properties of these documents and the properties of documents that were already assigned to classes in the training phase.

In this research, the set of classes obtained from a concept hierarchy could be very large, the concepts could change and user profiles could have relations to multiple concepts on different levels in the hierarchy. This makes the process of manual creation of a training set, hard and time consuming.

To provide a generic mechanism for assigning concepts from a selected hierarchy to a user profiles based on short message, we use a classification method that is based on term matching and does not require manual training. When the terms (words) in a concept, or terms related to the concepts according to the hierarchical structure, occur in the message of user, the concept is assigned to the users. In addition, the parent and root concepts, that occur one level higher and on the first level of the hierarchical structure respectively, will be assigned to the user. We call this naive classification, because every match is

considered as an assignment of concept to a user. It does not use information like the frequency of assignments or level of the concepts. The naive classifier is the first block in the classification process in Figure 2.1. The assignment of concepts to user profiles is a bootstrapping process for linking the concepts to users and collecting statistical data that is used in the further classification process. Details about this classification algorithm and the implementations using information retrieval techniques are described in Chapter 5.

2.2.2 Improved classification

The simple approach of the unsupervised naive classification, leads to classification errors. These errors occur because concepts could have an ambiguous meaning or terms related to concepts could be very generic and occur very frequently in messages. The results of the evaluation of the naive classification (Chapter 6) show that only 37 % of the naive classifications are correct. Grouping the profiles based on these classification results will result in invalid groups and invalid user profiles in these groups.

To improve the results of the naive classification we take into account statistics, such as how often a concept was assigned to a user. Using Support Vector Machines, a machine learning method, models are trained based on statistics related to the concepts, the messages, the assignments of concepts to user profiles and the validity of the classification by the naive classifier. These statistics are not related to the content or meaning of the concept or the message in order to be able to use other concept structures and collections of user profiles. In Chapter 7 we present the methods and the types of statistical data that we use to improve the results of the naive classification approach.

2.2.3 Clustering

Clustering makes it possible to discover groups based on the concepts that have a relation to user profiles. We use a hierarchical agglomerative clustering approach based on the co-occurrences of concepts in user profiles. Because of the hierarchical relation between the concepts that are assigned to the user profiles, these concepts occur frequently together in the user profiles. This results in groups that have similarities with the structure of the concept hierarchy. We focus on the groups that have concepts that diverge from the original concept structure, which could be interesting. The discovered relations and groups are declared by analyzing the occurrences of the concepts in the profiles. The effect of the hierarchical concept structure to discover relations on more abstract conceptual levels is discussed in Chapter 9 by comparing results of clustering with and without incorporating the hierarchical information.

Chapter 3

Data sources

To achieve grouping of user profiles based on hierarchical concepts we need an evaluated collection of short messages (Figure 3.1) related to users and associated concepts. A data set with these properties did not exist. However, there are other researches that deal with documents that are associated with (hierarchical) concepts. Section 3.1 gives a short overview of these kinds of data sets that were used in other researches. In Section 3.2 we give background information about the availability and the context of short messages in online social networking sites. How we gathered a collection of short messages is described in Section 3.3, and Section 3.4 gives the characteristics of the short messages in this collection.

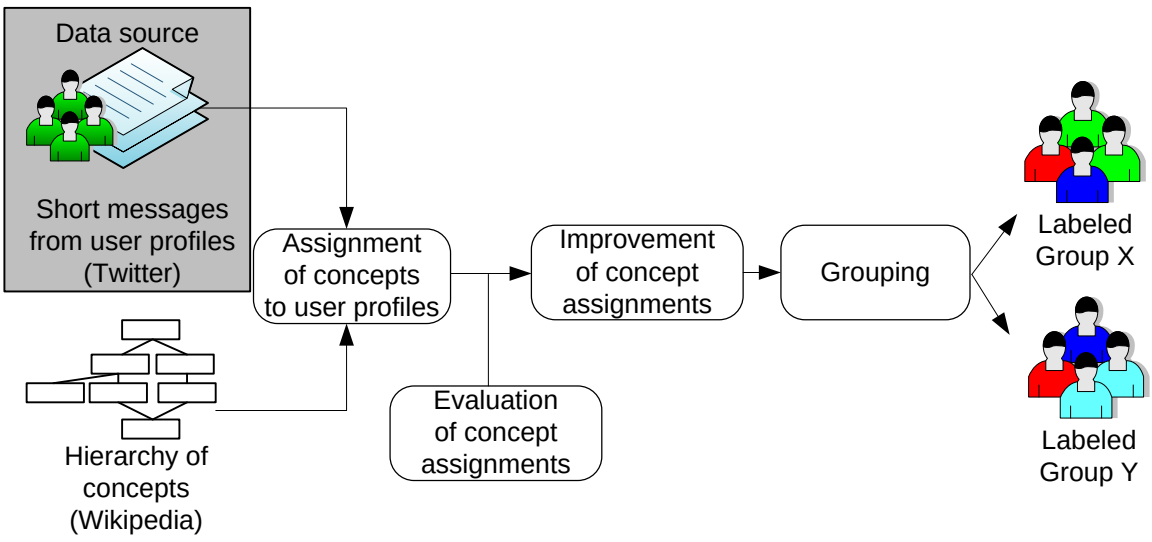

Figure 3.1: The data source in the text mining process

3.1 Related work

Other researches [6, 39, 18, 46] with a focus on categorizing and grouping of text data often use data sets where concepts (also called categories or topics) often do not have hierarchical relations and documents in the set are not related to the domain of online social networks. Table 3.1 shows these often used collections for evaluation of classification and clustering processes and their characteristics: the domain of the text documents, the length of the documents, the number of levels in the concept hierarchy, the total number of documents in the set, the total number of different categories assigned to the documents and the average number of categories assigned to each document.

We use a collection of short text messages in the domain of online social networking sites that were linked to concepts from a hierarchical structure with a depth of five (last row in Table 3.1).

The next sections compare short messages in online social networking sites and the properties of the Twitter data set that we use. Chapter 4 describes how the concept hierarchies with categories was selected and obtained. Chapter 5 and Chapter 6 show how the collection, with associations to the hierarchical concepts was made and evaluated.

3.2 Online social networking sites with short messages

There are many online social networks sites on the internet where users can build a profile and leave (personal) information at different components of the site. Table 3.2 shows six popular online social networking sites with the components where users can leave short messages, the question the user was asked and the maximum length of the messages. In these cases, people leave a message explicitly, however there are also sources like the photo albums on these sites or photo sharing sites like Flickr where people fill in a title or description of a photo. These descriptions could also be considered as short messages.

Twitter is an online social networking site focused only on short messages and has a very open character compared to the other sites. On other sites, users often protect their profile by allowing only friends to view the information. Figure 3.2[1] shows that less than 7 % of the accounts on Twitter is protected. Protected profiles are only visible for users approved by the profile owner. Besides that, Twitter provides an API[2] that is able to access profiles and messages without scraping HTML pages. There is also a data set of 900,000 Twitter messages available at on the website of CAW2.0[3]. Because of the public access and the easy way of accessing the data, we use Twitter messages as source for short text messages.

[1] Source: http://www.techcrunch.com/2009/10/05/twitter-data-analysis-an-investors-perspective/
[2] http://apiwiki.twitter.com/
[3] http://caw2.barcelonamedia.org/node/7

Data set	Domain	Doc. length [4]	Concept levels	# Doc.	Associated categories	Avg. # of categories per doc.
Case+ALR+JLR[6]	Law	Long	-	951	11	-
50-Topics-2HNs[6]	Law	Short (36.4)	-	533	50	1.57
50-Topics-3HNs[6]	Law	Short (43.2)	-	756	50	2.44
LRC (Law Firm)[6]	Law	Long	-	4,517	39	1.25
Reuters-21578[5][39, 8]	News	Long	3	[6] 11,367	135	1.26
20 Newsgroups [18, 17]	Misc.[7]	Medium	3	20,000	[8] 20	[9] ~1.04
RCV1 subset [46]	News	Medium	4	6,588	23	3.50
PubMed [46]	Medical	Long	-	3,687	15	3.20
CaseLaw [46]	Law	Long	-	2,550	20	4.82
CICLing-2002 [13]	Computational Linguistics	Medium	-	48	4	1.00
Twitter profiles and Wikipedia concepts (this research)	OSNS	Short (12.5)	5	1,503	210	12.51

Table 3.1: Characteristics of evaluated collections with documents and concepts

[3]Based on average number of terms in document: Short 0-50, Medium 50-300, Long > 300
[4]Statistics from http://www.cs.umb.edu/ smimarog/textmining/datasets/index.html
[5]Documents with at least one category.
[6]Discussions of people related to: Computers, Sports, Science, Religion, Politics
[7]Categories (newsgroups) have a hierarchical relation to five mid-level categories.
[8]Approximately 4 % of the articles are crossposted.

Site	Component	Question / label	Max. # char.
Facebook [9]	Wall	What's on your mind?	420
Hyves[10]	Who, What, Where?	Tell your friends where you are and what you are doing!	>1000
LinkedIn[11]	Activity	What are you working on now?	140
MySpace[12]	Status and Mood	What are you doing right now?	140
Orkut[13]	Updates	Set your status here	140
Twitter[14]		What are you doing?	140

Table 3.2: Online social networking sites and their short messages

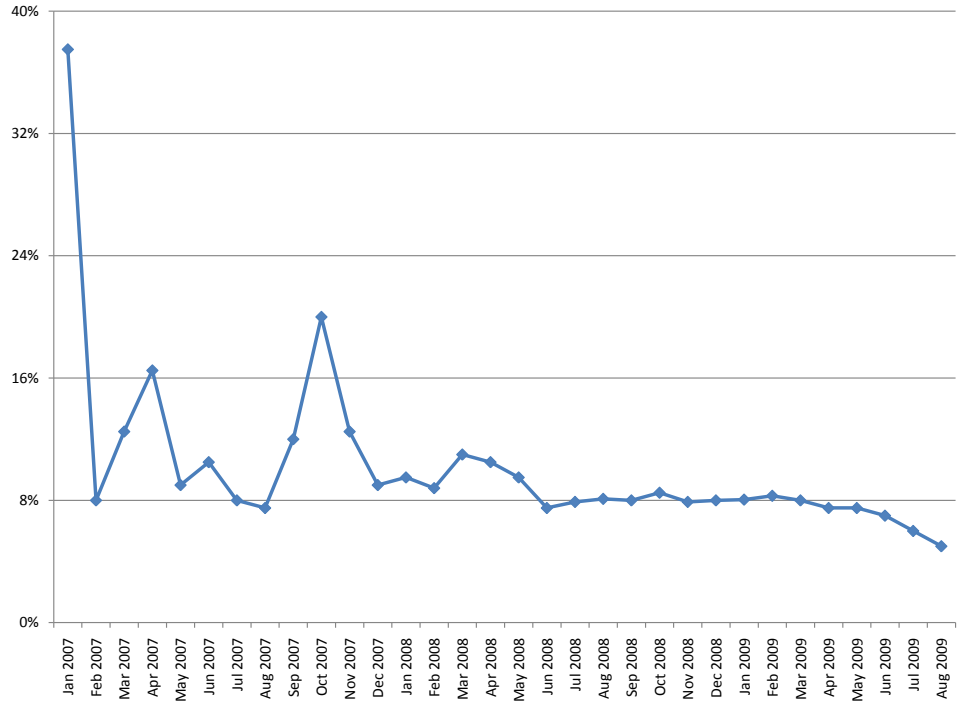

Figure 3.2: Trend of protected Twitter accounts

[9] http://www.facebook.com
[10] http://www.hyves.nl
[11] http://www.linkedin.com
[12] http://www.myspace.com
[13] http://www.orkut.com
[14] http://www.twitter.com

3.3 Gathering Twitter messages

To retrieve messages from Twitter by using the API we build a Java program that is able to fetch these messages. This program uses the library Twitter4J[15] to make easy use of the Twitter API from the program code. The tool we built takes a user as input and fetches the followers (people interested in this user) and the latest 100 messages of these followers. We are not interested in a collection of messages in different languages, for this research we focus only on English messages. However, there is no language field in the profile of a Twitter user to determine if it is an English-speaking user or not. We use the available time zone field of the user and select only users in time zones ending with the string '(United States & Canada)' to increase the chance of selecting English profiles.

Due to the rate limiter of Twitter, the number of requests, using the API, is limited to 150 per hour. Fetching a list of 100 usernames and their profiles is one request and fetching a list of 100 messages is one request. We fetched a maximum of 100 messages of 19,261 profiles with at least 10 messages in October 2009 and stored them in an XML file, with the same DTD as specified by CAW2.0, except that we added the date the profile was fetched. Together with the CAW2.0 profiles we got a set of 46,390 profiles with a total of 2,136,285 messages.

3.4 Our short text messages collection

For our research, we use a selected set of 1,503 user profiles. Chapter 5 describes how the profiles were selected and how the messages were processed and stored for further usage in the text mining process. Of course, all messages together could be considered as one large document, however in that case you lose information about how often a user talks about a specific topic. The indicator of the number of times a topic occurs in a document of all messages differs from how many times a message with a topic occur. For example, the message 'Tennis is a great sport, that is why I play tennis. I love tennis.' is about 'tennis'. The word occurs three times in the document (term frequency), but it could be the only message about tennis in the user's profile. The message frequency is in that case one.

Figure 3.3 shows the characteristics of the number of words in the messages in the full Twitter collection of 2,136,285. The collection has an average of 12.45 words per message and 70.55 characters per message, which is very short compared to other existing test collections.

[15] http://yusuke.homeip.net/twitter4j/en/index.html

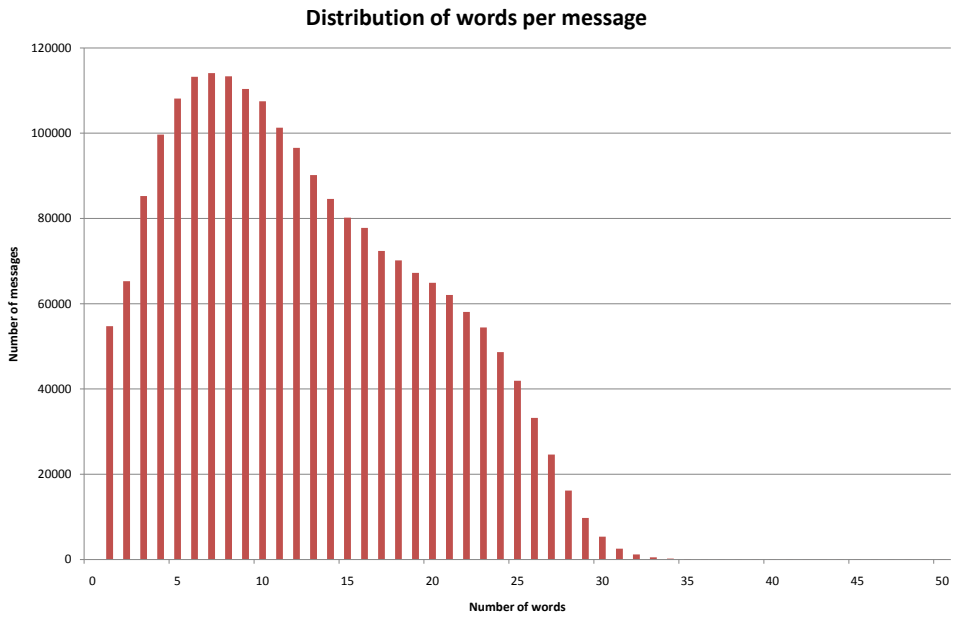

Figure 3.3: Number of words per message

3.5 Summary

For evaluation in this research, a collection with the following properties is required:

- Short documents;
- related to users;
- from an online social networking site;
- related with concepts (or categories);
- the concepts have a hierarchical relation.

A collection with all these properties did not exist. In this chapter we presented a collection of short messages with a maximum of 160 characters and an average of 12.45 words per message from Twitter users, that meets the first three requirements. Concepts from Wikipedia (Chapter 4) are attached to the user profiles in the collection (Chapter 5) to meet the last two requirements. The collection is evaluated (Chapter 6) in order to use it for validation of improving the classification task in this research.

Chapter 4

Concept hierarchies

A concept hierarchy contains a large number of concepts organized into multiple levels in such way that concepts at a higher level have a broader meaning than those at lower levels[47, 37], often called taxonomies. Organizing information in a hierarchical structure of concepts provides the opportunity to take advantage of the (semantic) relations between concepts in the structure. In this research, we use this strategy to discover (semantic) relations between user profiles on different hierarchical levels, as described in Subsection 1.3.1. This requires a concept hierarchy that covers concepts that occur in the short messages of users in online social networking sites. Section 4.1 gives an overview of concept hierarchies used in other researchers compared to the requirements of the type of hierarchy that we need. Section 4.2 describes the process of obtaining a hierarchical concept structure and the properties of the structure that is used in our research.

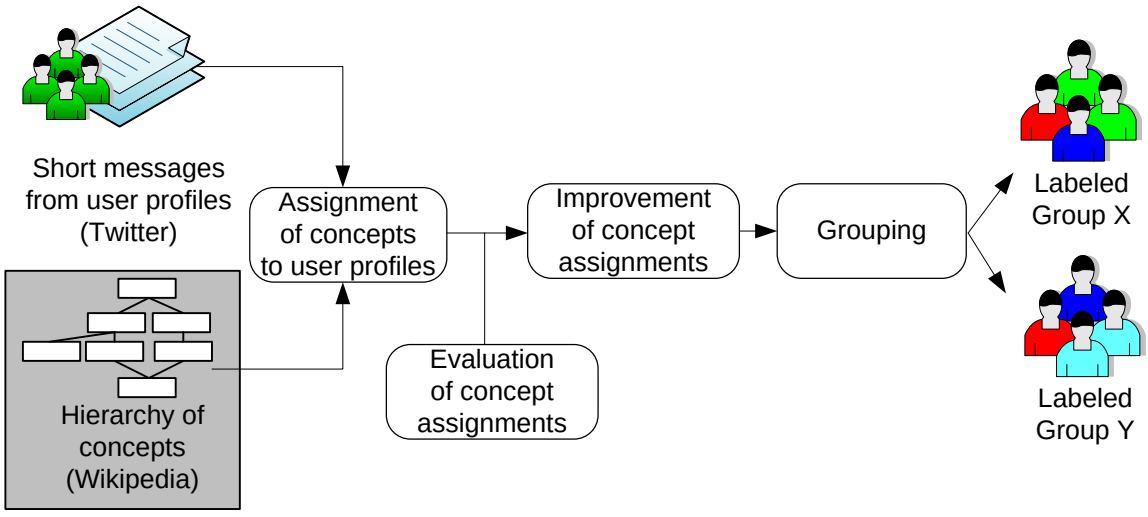

Figure 4.1: The hierarchy of concepts in the text mining process

4.1 Related work

In this research, it is relevant that the concept hierarchy is not related to a specific domain, because the concepts that are mentioned in the short messages on OSNS's are not domain specific. The most used domain independent concept hierarchies in text mining research are WordNet, the Open Directory Project and the online encyclopedia Wikipedia.

WordNet

WordNet is a lexical database organized in synsets consisting of synonyms and description of semantic relations between these synsets[25]. WordNet contains hypernyms of words, which describe a hierarchical relation, where the hypernym has a more generic meaning than another word. For example, *color* is a hypernym of *red* and *vehicle* of *car*. Hierarchical relations between words could be discovered to look up if two words have the same hypernym. WordNet describes the relations in a very structured and complete way. It contains not only *is-a relations*, but also *has-part*, *is-made-of* and *is-attribute-of* relations [34]. However, WordNet contains 120,000 synsets and contains no domain specific information [41]. For example, it does not contain many named entities (names of persons, movies, books, etc.), which are common topics people write about on online social networking sites. Another issue is that WordNet is English-only, so hierarchies from WordNet could not be applied to data sets with non-English documents.

Open Directory Project

The Open Directory Project (ODP) is a collection of links to web pages stored in a hierarchical structure called the directory [30]. It contains links to 4,5234,425 sites organized in over 590,000 categories including different languages [33]. Compared to WordNet, ODP contains named entities, however there is no description of the type of relations between the different categories and between the categories and web pages. Another difference is that ODP is a directory with a tree structure: a category or web page could not occur in multiple other categories. Besides that, it does not contain more abstract concepts like *red* and *color* [17].

Wikipedia

Wikipedia has many properties that WordNet and ODP do not have. Wikipedia is available in many languages and has relations between those different languages and sister projects with news articles or dictionary definitions and is up-to-date. The English Wikipedia contains 6,833,928 articles categorized in 510,674 categories[49]. An article can occur in multiple categories. The categories and articles contain named entities and abstract concepts. The coverage of domains is very broad and up to date [41]. Unfortunately, the structure and description of relations is less rich than in WordNet and does not form a taxonomy with only 'is-a' relations in the hierarchy [37]. The Wikipedia categorization system is a thematically organized thesaurus, concepts on higher levels in the structure often do have a broader meaning, but that is not always the case. In Section 4.3 we discuss other types of relations that occur in the structure. Schönhofen [41] concludes that representing documents using Wikipedia

categories result in equal or better results than using their full text, and that is without exploiting the hierarchy.

4.2 Gathering a Wikipedia structure

Because the short messages in online social networks are often about recent subjects, named entities as well as abstract concepts, we will use the Wikipedia category hierarchy to discover relations in user profiles with short messages.

Wikipedia data is available as a database dump. We used the dump of base per-page data (page.sql) and category membership link records (36,739,993 records in categorylinks.sql) of 29th September 2009 [49]. The page data consists of metadata about all types of pages: user pages, talk pages, help pages, articles and categories. We are only interested in the last two. Figure 4.2 shows the parts of the database we use to obtain the hierarchical structure. The *namespace* stores the type of the page, using this attribute we can select only articles and categories. The links between categories and articles is stored in the *catagorylinks* table. The attribute *from* refers to an *id* of a *page* and the attribute *to* contains a string with the title of the category the article occurs in. In our research, we use a subgraph of the Wikipedia category and article hierarchy. Because the data set of user profiles with concepts from the hierarchy has to be manually evaluated (Chapter 6), the number of concepts in the hierarchy was limited in order to reduce the number of relations between user profiles and categories. Topicus FinCare B.V. has customers in the health care domain. For them it is interesting to discover groups of people in this domain. That is why the category 'Health' and the four underlying categories 'Disability', 'Hygiene', 'Health effectors' and 'Diseases and disorders' are used as start categories to build the concept hierarchy.

When using the 'Health' concept hierarchy, profiles are grouped based on concepts in this structure, which are in the health domain. Gathering different concept hierarchies makes (multi-)domain-driven grouping of profiles possible. Of course, it is possible to take a very generic category (e.g. 'Main topic classifications') as root and gather all sub categories, which cover many different domains.

The algorithm to retrieve the category structure consists of the following steps:

- Lookup the title of the category in the *to* field of the *categorylinks* table. The related *from* fields contain the *id*'s of pages or categories that occur in the category.
- Lookup the *title* in the *page* table using the *from* value of the *categorylinks* entry.
- Check whether the category or article according to the *title* should be filtered or not (see Subsection 4.2.1).
- Add the information (title and depth) to the structure.
- If the found title is a category (depending on the *namespace* value), this process could be repeated from the first step until you reach a specified depth.

4.2.1 Pruning the category structure

The structure of titles linked to the 'Health' concept to a depth of 5, becomes very large and contains concepts that do not have a very strict relation to the health domain. In order to be able to create a manually evaluated collection, we reduce the concepts in the 'Health' structure by filtering branches with words we are not interested in. We assumed that they are not relevant for assigning these concepts to user profiles to discover groups. Examples of these type of concepts that occur in the structure are 'Films involving disabilities', 'People with disabilities', 'Blind people', 'Blind animals' and 'Fictional diseases'. We filtered out branches with category titles containing the following words: history, animal, fiction, organizations, music and films. In addition, when a category title starts with 'Lists of' it is left out, because these type of categories contain only pages with a title that starts with 'List of'. These titles are not useful to use them as concepts.

4.3 Characteristics of the Wikipedia structure

Graph with multiple paths to concepts

For the concept hierarchy with 'Health' as root and a depth of 5 levels, this resulted in 535 unique concepts. Due to the graph structure of the Wikipedia category system, categories can be found multiple times via different paths. In this case, 792 categories and 29,798 page titles were visited and stored. For example, the category 'Aphasias' was visited via 10 different paths from the root node with a maximum distance of 5. Subsection 5.3.1 describes how the structure is stored. Figure 4.3 shows how many categories where visited multiple times via different paths.

Parent-child relations

As mentioned in Section 4.1, the relation between child and parent categories and articles in the Wikipedia structure are not always of the type 'is-a' or 'is-a-subset-of', in case of a collection. In our Wikipedia structure, 'Sleep apnea' is a child of 'Sleep disorder', which is an 'is-a' relation. However, there are many other types of relations that occur in the structure, for example 'is-part-of' (*Cleaning is part of Hygiene*), 'is-a-symptom-of' (*Impulsivity is a symptom of ADHD*) and 'is-used-for' (*Iron is used for Laundry*). Examples of more complex relations to describe are the categories 'Braille' and 'Blind people' that both occur in the category 'Blindness'.

Due to the fact that Wikipedia is an open collaborative system that enables users to categorize the content of the articles, parent concepts do not always have a broader meaning than child concepts. For example, one of the parents of 'Sleep disorder' is 'Sleep medicine', which looks like an error in the structure. A more logical structure would be that 'Sleep medicine' is the child concepts, because it is used against a 'Sleep disorder'.

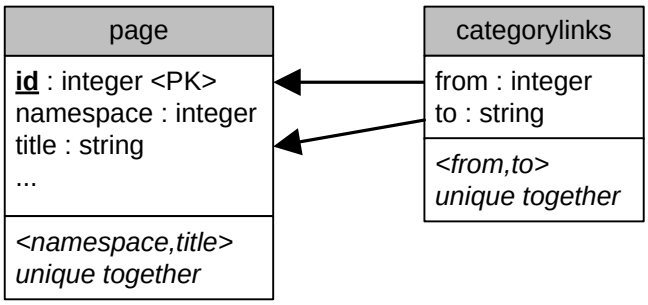

Figure 4.2: Wikipedia database layout related to pages and categories

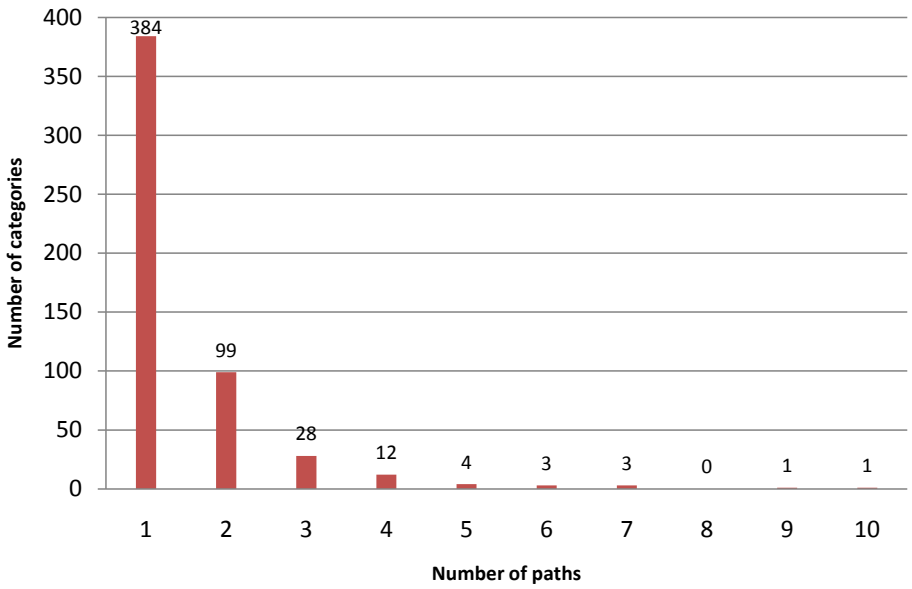

Figure 4.3: Distribution of categories and the number of different paths

4.4 Summary

Characteristics of the topics of short messages are that they are not specific to one domain, could be recent or are names of people, books, movies, etc. To discover relations between user profiles on higher conceptual levels we use a hierarchical structure of concepts. To map concepts to user profiles based on short text messages, there should be an overlap in the set of concepts and the topics in the content of the published messages. In this chapter, we introduced Wikipedia that has the required properties to match topics of short text messages:

- Domain independent
- Up to date
- Named entities as article titles
- Hierarchical structure due to the category system

From the Wikipedia category and article structure, we use a set of 535 hierarchical related concepts of 5 levels deep. This concept structure is limited to the health domain. However, the approach using Wikipedia as source for a structure of hierarchical concepts is not limited to this domain.

Chapter 5

Naive classification of user profiles

To link concepts from a hierarchical structure of concepts to user profiles, we use a naive classification approach (Figure 5.1). In Section 5.1 we discuss related research in the context of classifying user profiles and hierarchical concepts and underlying data representation models. Our approach for the classification of user profiles using concepts from a Wikipedia category structure and the explanation of the consequences of this approach is described in Section 5.2. Section 5.3 shows the process of applying the approach of naive classification with the data set of Twitter messages and the concept structure from Wikipedia (Figure 5.3). The naive classifier will be evaluated in the next chapter. Chapter 7 presents how the results of the naive classifier could be improved by using additional information and classification algorithms based on machine learning.

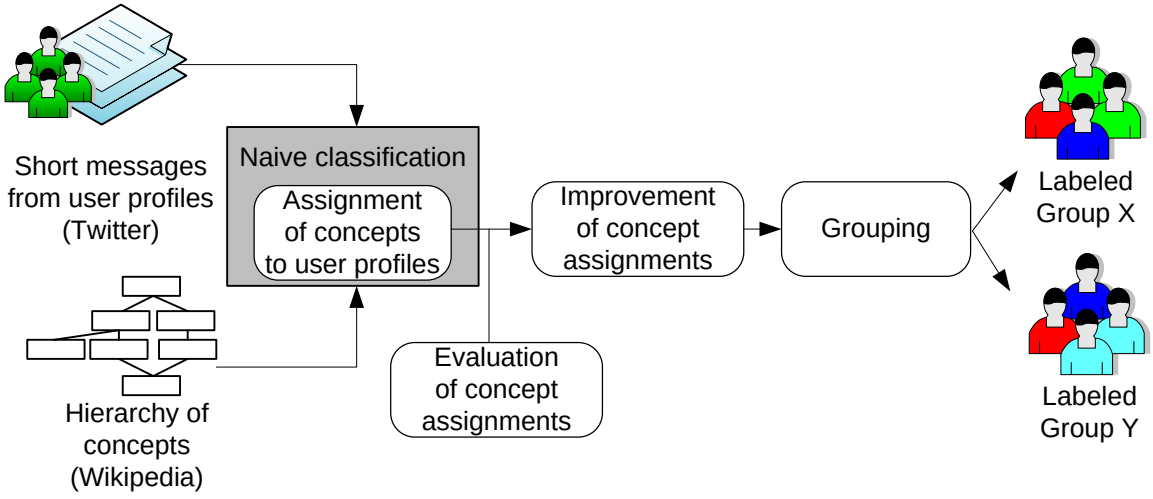

Figure 5.1: The naive classifier in the text mining process

5.1 Related work

The representation model and the techniques for the classification of profiles into hierarchical concepts should be suitable for the properties of a message stream and should meet the requirement that the classification of user profiles could happen without supervised training on a specific hierarchy of concepts. In this research it is important that this is an unsupervised process, because the number of concepts in the structure could be very large, which makes it very time consuming to assign these concepts to user profiles manually. In addition, changing (the domain) of the hierarchy requires that this time consuming process should be repeated with the new concepts in the hierarchy.

In Subsection 5.1.1 we introduce two representation models that are used in the context of text mining and discuss what is the best model for discovering relations between short messages (documents) and the related users. Subsection 5.1.2 discusses the related work to classification of user profile data in relation to hierarchical concepts.

5.1.1 Representation models

5.1.1.1 Vector Space Model

Text representation models serve as an intermediate step between the raw text data and the analysis using text mining strategies like classification and clustering. Most of the existing text mining methods rely on use of the Vector Space Model (VSM), also called the bag-of-words model, known from information retrieval [44, 4, 25, 17]. The Vector Space Model introduced by Salton [40] is one of the oldest, most widely used and most extensively studied models for text mining [44, 1] . In this model, documents are represented as an unordered collection of words. Each document is described by a vector which dimension values are related to a word (term). The values in the document vectors could be used by analysis functions to discover similarities between documents or similarity between a query and documents. One of the disadvantages of the VSM is that it loses semantic information, because it does not preserve the word order. This means that a document with the words 'alarm' and 'clock' at different positions could have the same representation as a document where these words occur together. Other representation models used in text mining are often extensions of the VSM. There are also models that focus more on the (semantic) relations between words or on phrases, such as Latent Semantic Indexing.

5.1.1.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a model that has more focus on discovering semantic relations. Based on the vector space, LSI uses Singular Value Decomposition to compute a smaller semantic subspace. This subspace consists of less noise and redundancy and problems with ambiguous words are reduced. LSI is based on statistics and assumes that words that often occur together in a document have a semantic relationship. In documents about 'influenza' the words 'flu' and 'illness' probably occur more often. Based on the smaller subspace, patterns in the relationships between terms in the documents could be discovered.

LSI could discover relations between concepts based co-occurrence of words, however in small documents like the messages (~12 words) there are not many words in the same context. The alternative is considering all messages in a profile as one documents. However, this would result in a document with many different topics which often do not have a semantic relation. This makes LSI less suitable for usage in the context of short messages.

Another disadvantage is that discovered concepts (related words) do not rely on concepts identified and described by humans and do not have a hierarchical relation, while Wikipedia categories have both of these properties [17].

5.1.1.3 Vector Space Model with extensions

Because discovering relations on higher concepts levels using an existing structure of hierarchical concepts is part of this research, Latent Semantic Indexing is in this research not a good method to store data from short messages. In this research we use a model based on the Vector Space Model, that also stores the word order in the documents, to preserve this semantic information.

5.1.2 Hierarchical concepts and classification

In this research, it is relevant that hierarchical concepts could be linked to the user profiles based on the short text messages in the profiles, in order to be able to discover relations between users on higher conceptual levels.

Ramanathan et al. [38] present a method for constructing user profiles using Wikipedia concepts for describing the user interests. The purpose of constructing the user profiles is to use these for personalized information retrieval and personalized web applications. The approach they take consists of mapping visited web pages of a user (documents) to Wikipedia concepts by querying an index of Wikipedia pages with a query generated from the documents. The titles of the Wikipedia pages that are returned as the result of the query are used for building the user profile. Feeding the entire document content as (fuzzy) query to the user profile results in a poor precision, also called the long query problem. They try to overcome this problem by selecting relevant words from the document, building a query of these words and selecting the top 20 results of Wikipedia pages returned by the query. The result of the matching process is stored in an index with the documents together with the related Wikipedia titles that matched the document. This technique is based on the approach taken by Gabrilovich and Markovitch [17], where document texts were matched to Wikipedia articles, both stored using the Vector Space Model.

Sieg et al. [43] present a framework for integrating user profiles and concept hierarchies. Their data source is a set of documents in which the user has shown interest. The documents are stored as term vectors and a clustering algorithm creates a concept hierarchy based on term frequencies in the term vectors. This generated concept hierarchy is used to assist the user in the creation of effective queries for a search task. The created (domain-specific) concept hierarchy provides a semantic context for (initial) queries in the information retrieval system.

Dumais and Chen [10] explore the possibilities of exploiting a hierarchical structure for classifying web content. Their goal is to automatically classify web search results into an existing hierarchy. A model is trained by using Support

Vector Machines (SVM) and a set of documents that have been manually classified into the concepts of the hierarchy. New documents (visited web pages) are classified by this model to the concepts in the structure.

Kim and Chan [27] present an algorithm to build a user interest hierarchy (UIH), organizing user's general to specific interests. They created a divisive hierarchical clustering algorithm that recursively divides clusters in child clusters based on words in visited web pages of a user. The information (e.g. frequent words) in the UIH provides statistics to rank result from a search engine according to the interests of a user.

These researches create implicit (hierarchical) user profiles based on documents (visited web pages) related to users for personalization of information retrieval. Different text mining techniques, such as classification and clustering are used to build a hierarchy that represents interests of a user. A concept hierarchy obtained from unstructured user data could contain relations between concepts that do not have a semantic relation. That makes it less useful for discovering groups of profiles based on semantic relations. By using an external concept hierarchy, like Wikipedia categories, information will be organized in a human defined structure based on semantic relations.

Schönhofen [41] identifies topics of documents using the Wikipedia category structure. An algorithm relates Wikipedia categories to documents by matching article titles with words of the documents. Categories are then weighted by different factors, such as words shared between the document and the Wikipedia article title, strength of the match and properties of the Wikipedia article. Only the titles and categories of Wikipedia articles are used by the algorithm to identify the topics of documents. The actual text of the articles and the hierarchy of the categories are not exploited.

In this research, the documents are very small text messages that are not domain specific. We focus on discovering relations between users using an existing structure of hierarchical concept. To achieve this, our naive classification approach assigns concepts from the Wikipedia category structure to user profiles based on short messages. This naive approach is based on the query-based method of Ramanathan et al. [38] for incorporating the hierarchical information and the method of Schönhofen [41] to link categories to documents based on the titles and the content of the documents.

5.2 Naive classification approach

In Chapter 2 we mentioned that we incorporate a hierarchy of concepts to discover semantic relation between profiles and that linking the concepts to user profiles is considered as a classification problem. For the classification, we use the short text messages in the user profile as documents and the concepts as labels. Subsection 5.2.1 defines the terms that we use for describing the approach and the classification algorithm.

In this section we discuss our requirements for the approach (Subsection 5.2.2) and our assumptions about the short messages and the concepts (Subsection 5.2.3). The existing classification approaches of Ramanathan et al. [38] and Schönhofen [41], which also use Wikipedia concepts for classification, are discussed in relation to the classification problems and the assumptions about short messages and concepts. In Subsection 5.2.4 we present our naive classification al-

gorithm and discuss how it meets the requirements of the classification process. Subsection 5.2.5 shows the problems in text classification in the context of this research that will not be solved.

5.2.1 Definitions

Term	Meaning
Concept	A string obtained from a category title in the Wikipedia structure.
Category title	The title of a category in the Wikipedia structure, used as a concept.
Page title	The title of an article in Wikipedia that occurs in a category and has a relation to concepts, based on the categories the article occurs in.
Terms	Words obtained by splitting string (category title, page title, short message) by whitespaces and lowercasing the tokens.
Parent concept	A concept that occurs one level higher (than a concept) in the Wikipedia structure.
Root concept	The concept that occurs at the first level on the path to a concept in the Wikipedia graph structure of categories.

Table 5.1: Definition of terms

5.2.2 Requirements of the approach

Unsupervised classification

The main requirement of the classification process is that it is an unsupervised process. The process should require no input of a user that directs the classification process. For example by manually labeling a small set of user profiles or short messages to the correct concepts, so that the classification algorithm could link new profiles to concepts based on the similarity with previously classified profiles.

Linking concepts to profiles (documents)

Ramanathan et al. [38] and Schönhofen [41] both use an approach that requires no user input. The classification is only based on matching terms and the similarity of data representation of the concepts (sometimes together with related data like the Wikipedia article content) with the data representation of the document that needs to be classified.

To measure the similarity between a document and a concept, Ramanathan et al. [38] use a query-based matching algorithm that matches important words in the document with the words in the concept of the Wikipedia articles using a search engine. Important words from documents are words with a greater or equal length than the average word length in the document and a greater or equal word frequency than the average word frequency in the document. The top 20 Wikipedia articles (the concepts) that are similar to these important words based on the ranking of the search engine, are linked to the document.

The algorithm of Schönhofen [41] identifies and ranks all Wikipedia categories supposedly related to the document by matching Wikipedia article titles with words of the document. Top 20 ranked categories are linked to the documents. The pages in the categories or the content of the articles are not used in the matching process.

Because the approach of classification by matching terms is an unsupervised process and we do not incorporate the article content, our algorithm is based on the strategy of Schönhofen [41].

Hierarchical relations

Another problem of the classification process is that we want to use the hierarchical concepts to discover relations between profiles on more abstract conceptual levels. Ramanathan et al. [38] and Schönhofen [41] do not incorporate the hierarchical information directly. Their sets of concepts contain a hierarchical relations, however if a concept is not assigned during the classification process based on term matching, then the classified document does not contain the higher-level concepts.

In our algorithm, we will use concepts at higher levels even if they do not have a relation with the document based on matching terms.

5.2.3 Assumptions about short messages and Wikipedia concepts

The text classification approaches discussed in Section 5.1 use different kinds of document types, such as web pages. These types of documents have different characteristics than the short messages from online social networking sites. In this section, we discuss the assumptions that we made about short messages and the structure of Wikipedia concepts in relation to the classification approach based on matching terms.

Short messages are not a good source for selection of important words

Ramanathan et al. [38] use a selection of important words for selecting relevant Wikipedia concepts during the classification process. For a short message, this process could result in the selection of words that do not cover the topic of the message, which results in the selection of irrelevant Wikipedia concepts. Or when the important words are very generic, for example 'tennis', this results in related articles, such as 'tennis', 'wheelchair tennis', 'tennis court' and 'US Open (tennis)'. However, the word 'tennis' in a message does not mean that the user has a relation to concepts like 'disabled sports' or 'tennis tournaments in the United States'.

A message stream is not a good source for selection of important words

A message stream contains a lot of different topics. This variation in topics is reflected in the results of the selection of important words. For example, when the words 'wheelchair', 'olympic', 'games' and 'ball' occur frequently in different messages, this results in the selection of articles that cover all these words, like 'basketball', 'softball', 'football', 'wheelchair football', 'wheelchair basketball', 'tennis' and 'women's basketball'. While the individual words occur in different messages and do not have a relation, articles where these words

occur together will be used as relevant concepts. The number of Wikipedia concepts that possibly have a relation with the user (the recall) would be high, while the number of concepts that actually do have a relation with the user (the precision) is low.

When all words of a Wikipedia title occur in a short message the related concept is relevant to the user

Schönhofen [41] selects Wikipedia categories if some of the words in a title occur in a document and the weighting score is high enough. This could result in a low precision, because the title 'wheelchair basketball', should not be matched to users that have only the word 'basketball' in a message. We assume that if all words in the Wikipedia category or page title occur in a message, there is a high probability that the relation between this Wikipedia concept and the user that published the short message exists.

When words from Wikipedia page titles occur separately in a short message the related concept is not relevant to the user

This is based on the assumption that page titles are often single words or multi-word expressions. The words in the page title together have a more specific meaning than separately. For example when the titles 'European Union', 'junk food' and 'alarm clock' are chucked into pieces they have a different meaning [4].

5.2.4 Algorithm

Based on the assumptions about the characteristics of the short messages and the concept structure described in the previous section, we use our own algorithm that matches terms from concepts with terms in short messages. This algorithm incorporates the hierarchical structure of the concepts.

Ramanathan et al. [38] and Schönhofen [41] do not consider all matches as a relation between a concept and the document. These methods select concepts based on a retrieval and similarity model. In our naive approach, we do not use the similarity information for selecting the relevant concepts. We consider all matches as a relation between a concept and a user profile. The similarity information is used for refinement of the naive classification approach (Chapter 7).

Our matching approach is based on the approach of Schönhofen [41], because we consider matches of terms of the page or category title) in a short message (document) as a relation. However, taking into account the last assumption, we make a difference between the matching of terms from page titles and terms from category titles.

Our classification algorithm assigns a concept to a user profile based on one of the following criteria:

- All terms of a page title related to the concept occur *exactly* in the same order and together in a short message of the user profile.
- All terms of a concept (category title) occur in a short message of the user profile.

When a concept is assigned to a user profile based on one of these criteria, the related parent and root concepts will also be assigned. This assignment process

Algorithm 5.1 Classification algorithm in pseudo code

```
Foreach [concept] in [concept structure]:
        Foreach [short message] in [short messages]:
                Foreach [page title] relatedTo [concept]:
                        If [short message] contains [page title] exactly:
                                Assign [concept] to [short message].user
                                Assign ParentOf([concept]) to [short message].user
                                Assign RootOf([concept]) to [short message].user
                If [short message] contains all terms in [concept];
                        Assign [concept] to [short message].user
                        Assign ParentOf([concept]) to [short message].user
                        Assign RootOf([concept]) to [short message].user
```

is repeated for every concept in the hierarchical structure, which results in the classification algorithm described in Algorithm 5.1.

5.2.5 Unsolved problems

Other problems that often occur in text classification methods could also occur when using our algorithm based on term matching. We give a short overview of these problems in the context of this research that are not solved.

Missed concepts by misspellings and synonyms in page and category titles

A short message could have a relation to a concept while the words of the Wikipedia category title or the phrase of the page title does not occur in the message. This could also happen when a user misspells a word. For example, when the user writes 'weelchair basketball', while he probably means 'wheelchair basketball'. We assume that when a user has a relation to the concept (interest, activity, etc.) and this is an important concept for the user, that he publishes more messages that are related to this concept. This will increase the probability that the concept eventually is covered by a Wikipedia concept.

However, concepts could still be not assigned when there is no match between terms in the titles and terms in the messages, while there is a relation between those two. This could happen when a message contains a synonym of a concept in the message. For example, the message 'I'm sick' does not match the term of the concept 'ill'.

Incorrect assignments of concepts by polysemous words in page and category titles

Polysemous words are words with multiple meanings [4]. For example the word 'iron', could mean an appliance for removing wrinkles from fabric or the metallic chemical element or a type of golf club. Based on terms there could be a match between the concept 'Laundry' and the message 'Iron Man is a good movie'.

Article titles in Wikipedia that occur in the article collection with different meanings are often marked with a suffix to distinguish the meanings of the articles titles. We measure the effect of this property on the results and use this property for improving the classifier.

Incorrect assignments of parent concepts due to the Wikipedia structure

Not all concepts in higher levels of in the Wikipedia category structure have a broader meaning. For example the message 'I've got a jet lag' will be matched by the page title 'jet lag', that is related to the concept 'Sleep disorder'. Parent concepts of 'Sleep disorder' are 'Sleep' and 'Sleep medicine'. While 'Sleep' is a more generic concept of 'Sleep disorder', 'Sleep medicine' is not. This results in a classification error caused by the Wikipedia structure.

5.3 Assigning concepts to user profiles

In order to perform the matching between concepts and user profiles we use the algorithm that is discussed in Section 5.2. The algorithm is based on matching terms. For the implementation we use an information retrieval system that matches queries generated from concepts with terms in short messages from users. The queries generated from the gathered Wikipedia page titles and category titles are used to retrieve messages of users that (probably) are related to the concept. There are already search engine libraries available that are able to index text and retrieve text documents. In this research we use the indexing and retrieval mechanism of the Apache Lucene[1] text search engine library. This section describes the implementation of the algorithm: how the data is stored using a storage model based on the Vector Space model (Subsection 5.3.1), how the queries are generated from the Wikipedia concepts (Subsection 5.3.2) and the underlying matching model that is used to match these queries to short messages (Subsection 5.3.3).

5.3.1 Storage

A Lucene index consists of documents and a document is a collection of fields. Fields contain the data and semantic information about how the data is stored. We distinguish two types of fields: keyword fields and text fields. During the indexing process the data for the keyword fields is stored without any modifications or tokenizing. Data in text fields is stored in their original form and as term vector after processing the text by a tokenizer. The tokenizer changes the text to a lower case string, removes special characters and breaks up the string in terms by the whitespace characters. This term vector is similar to the term vector of the Vector Space Model, however Lucene also stores the word order to support phrase queries.

Table 5.2 shows how the hierarchical information from the Wikipedia concepts is stored in the Lucene index. Only the page titles in a category, the parent category and the related category on the first level of the graph are stored, to reduce the complexity in the process of assigning the concepts to user profiles. Figure 5.2 shows a sub-graph of the concepts that exist in the gathered data set. Every concept is stored as a document in the index. Table 5.3 gives an example of how the categories 'Migraine' and 'Pain' are stored in the index. Page titles are considered as one token (e.g. 'phantom pain' and 'alarm clock') because separately these words have different meanings. Wikipedia article titles

[1] http://lucene.apache.org/

Field	Type	Content
Title	Text	The concept: the title of the Wikipedia category.
Parent	Text	The name of the parent concept: the title of the category in which this category occurs.
Root	Text	The name of the root concept.
Pages	Text	List of pages, each page title that occurs in the category is a token.
Level	Keyword	An integer value representing the distance from the root node in the graph structure.

Table 5.2: Fields in a document representation of a concept from Wikipedia

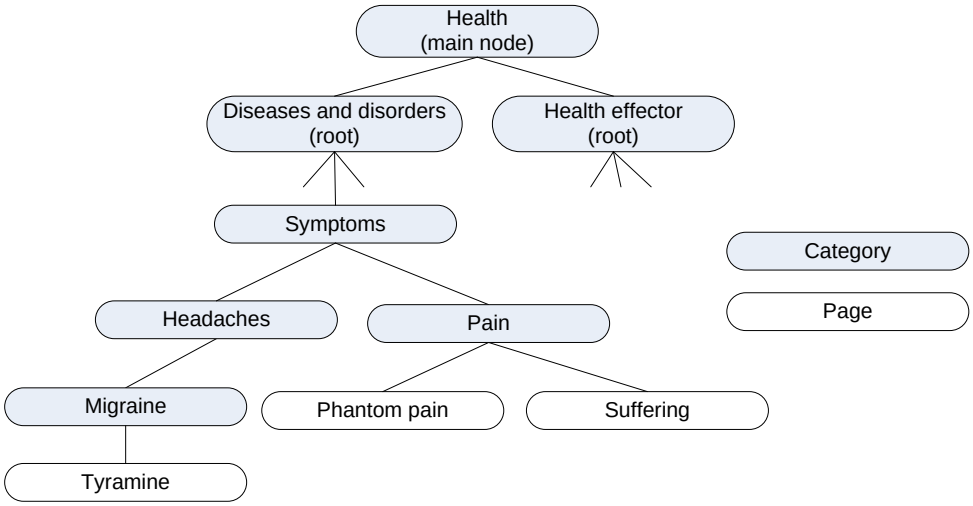

Figure 5.2: Part of the Wikipedia graph structure

sometimes contain a suffix between brackets like 'Sponge (material)' to distinguish the cleaning product from the animal. We remove this suffix, because we assume that the probability that both words occur in the same message is low.

Table 5.4 shows the fields of the documents representations that are used in order to store messages in the indexes. The text message is stored as a Lucene term vector and the username, which is unique for a Twitter user, is stored in the same document.

5.3.2 Retrieval and assignment process

To map the hierarchical concepts to the users, the naive classification, we use queries created from concepts to retrieve related messages from the index. Lucene implements a similarity function to retrieve and score documents in relation to the query (see Subsection 5.3.3). We use a tool that iterates over the concepts, generates queries from the data, retrieves all messages related to the query and assigns the concepts to the users that published the retrieved messages (see Figure 5.3).

Field	Migraine	Pain
Title	migraine	pain
Parent	headaches	symptoms
Root	diseases|and|disorders	diseases|and|disorders
Pages	tyramine	phantom pain|suffering
Level	4	3

Table 5.3: Example of categories stored in the index

Field	Type	Content
User	Keyword	The username of the user who published the message.
Message	Text	The short message.

Table 5.4: Fields in a document representation a short message of a user

5.3.2.1 Query generation

Every category and related data (Table 5.2) is fetched from the Wikipedia index. If the title of the category consists of more than one term, all these terms should occur in the message. However, it is allowed that the terms occur in different places in the message to assign this title to the user who published the messages. The query that will be used is like '+term1 +term2'.

When there category has underlying pages, queries will be created from the page titles to retrieve messages that are related to the category. If a page title consists of more than one term, all these terms should occur in the message in the same order, which is represented as a query formatted like '+"term1 term2"'.

5.3.2.2 Assigning concepts to users

For each retrieved message, the tool assigns the current concept (a category) to the user that published it. In addition, the hierarchical related parent concept and the root concept are assigned to the user as well. A match by a query generated from a page title also results in assigning the category in which this page occurs to the user. So, only concepts created from categories are assigned to users. Table 5.5 shows three concepts of Figure 5.2, the generated queries based on the data in the index and the concepts that will be assigned to the user in case of a match. We distinguish four types of assignments: page, base, parent and root. A page assignment is assignment using a query generated from a page title, while base assignments come from category titles. The parent and root assignments are related to the parent category and root category of the page or base assignment. Therefore, a single query results in one, two or three concept assignments to the users from the retrieved messages.

5.3.2.3 Gathering additional information

Besides the collection of combinations of users and concepts, extra information could be stored, like the type of assignment, the number of times a concept was assigned to a user, how many paths lead to the concept in the hierarchical structure and the retrieval score (Subsection 5.3.3) and term frequencies of certain words. This information could be used in later stages (Chapter 7) to improve

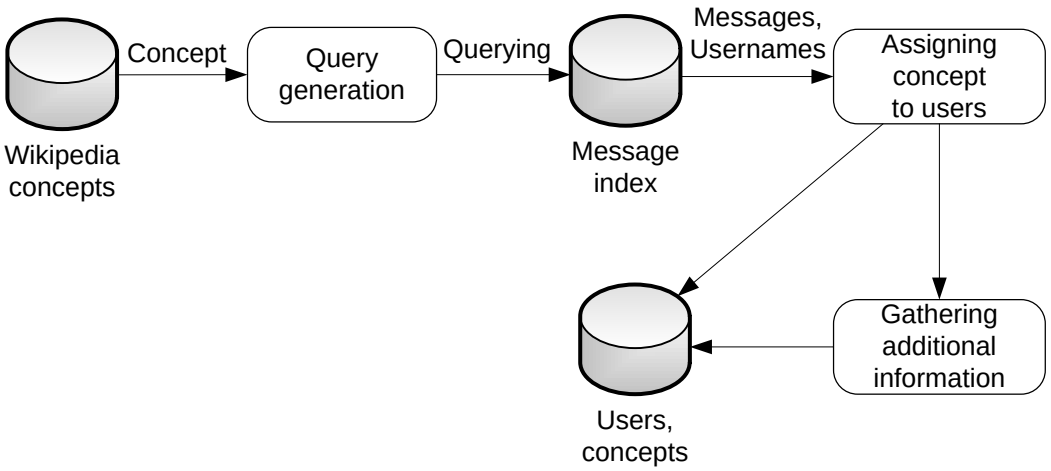

Figure 5.3: The naive classifier: concept to user mapping process

Concept	Query	Assigned concepts on match (assignment type)
Migraine	+"migraine"	Migraine (base), Headaches (parent), Diseases and disorders (root)
Migraine	+"tyramine"	Migraine (page), Headaches (parent), Diseases and disorders (root)
Pain	+"pain"	Pain (base), Symptoms (parent), Diseases and disorders (root)
Pain	+"phantom pain"	Pain (page), Symptoms (parent), Diseases and disorders (root)
Health effector	+"health" +"effector"	Health effector (base)

Table 5.5: Assigning concepts to users

the quality of classifications of concepts to users, without training on concept related data.

5.3.3 Retrieval model

The queries generated from Wikipedia categories and page titles are matched against the documents in the index of short messages. The matching and scorings model that is used relies on the default similarity model of Lucene [2]. During the matching stage of the retrieval process, the query is broken up in to terms and operators. These operators rely on the Boolean matching model. In this research we only use queries with the AND (+) operator, which results in that all terms in the query should occur in the document in order to match to the query. There are two types of terms in queries: *single term* and *phrased*. A single term is a single word, like 'clock'. In order to find a match between the document and the query, the documents (short messages) should have this term in the term vector. A *phrase* is a group of words surrounded by double quotes in the query, such as "alarm clock". Only documents that contain these

words in the same order after each other in document, match the phrase part of the query. All matches of short messages found by a query result in assigning a concept (origin of the query) to a user profile (publisher of the short message). However, based on the number of occurrences of terms in the documents and in the whole collection, the relevance level of the match could be calculated. To measure this relevance the following scoring function, that correlates to the cosine-distance between document and query vectors in a VSM, is used:

$$score(q,d) = queryNorm(q) \cdot \sum_{t \text{ in } q} (\, tf(t \text{ in } d) \cdot idf(t)^2 \,)$$

Document d is a short message related to a user and q is a query generated from a Wikipedia concept, like the queries in Table 5.4.

The main part of the scoring function is based on *tfidf* weighting. For every term t in the query q this weighting is done. The sum of all these weights is normalized using the query normalization which results in the score. The following term frequency (tr) and inverse document frequency (idf) functions are used:

$$tf(t \text{ in } q) = \sqrt{frequency}$$

$$idf(t) = 1 + \log(\frac{number\ of\ documents}{document\ frequency + 1})$$

Where *frequency* is the number of times the term t occurs in the document d, *number of documents* is equal to the total number of short messages in the collection and the *document frequency* is equal to the number of documents in which the term t occurs. Due to the idf factor, the term frequencies of terms that occur less often in the message collection get a boost.

To make the scores of different queries comparable, a query normalization factor is used that does not affect the ranking. This makes it also possible to use the score in a later stage of the process where the assignment of concepts to users is classified using a classification model. The query normalization function normalizes the score of all queries using the following function:

$$queryNorm(q) = \frac{1}{\sqrt{\sum_{t \text{ in } q} idf(t)^2}}$$

In the later stage of the process the score could be used to decide whether the relation between a concept and a user profile is correct or not.

Chapter 6

Evaluation of the naive classifier

The previous chapter described how the query-based classification mechanism assigns categories from the Wikipedia structure to the users. To be able to validate the performance of this process and later processes, an evaluated collection of the users with a message stream and correct assigned categories from the Wikipedia structure is needed. Section 6.1 gives background information about evaluation of collections used for classification and the method that is applied in this research. The guidelines for the evaluation process and the reproducibility of this process are discussed in Section 6.2 and Section 6.3. Section 6.4 presents the characteristics of the collections and analyzes the results of the evaluation.

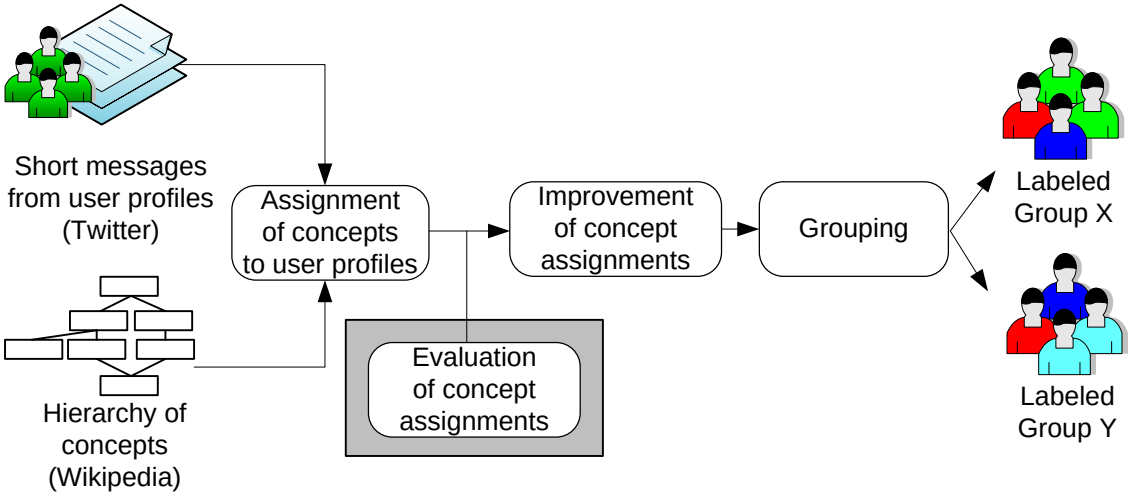

Figure 6.1: Evaluation in the text mining process

6.1 Judging process

To facilitate the evaluation of the naive classification a test collection is required. A test collection consists of correctly assigned concepts to user profiles, in order to be able to measure how the naive classifier and improved classifiers perform. Building a test collection is a manual task, that relies on humans to decide whether a user profile has a relation to a concept or not.

From a set of 46,390 Twitter profiles, 1,503 random profiles were selected that had at least one concept assigned during the naive classification process. For the evaluation, we only evaluate the 18,798 assignments made by the naive classifier. These assignments of concepts to users based on the message stream were evaluated by humans.

To provide a system for the manual evaluation of the collection by humans, a web based evaluation tool (see Figure 6.2) was built. The tool shows the concepts that were assigned by the naive classifier and the messages that were published by the user including some highlighted words. These highlighted words occur in the page or category titles that were used in the query that retrieved the message. They could assist the human judges (or annotators) during the manual evaluation of the initial classification. For every concept, a judge has to choose whether the assignment of the concept to the user profile based on these messages is valid or invalid. If the judge is not sure about whether he should select 'invalid' or 'valid', he could select the 'unknown' option. When this option is selected, another judge will review the profile. During the manual evaluation, 15 judges reviewed the results of the initial classification process using the tool. Profiles were randomly assigned to the judges and every profile was judged once. To get a reliable evaluation of the collection, every judge needs to know the same rules about when an assignment of a concept to a user profile is valid or not. To get as much agreement about the decisions that the judges make, every judge read the judgment guidelines with instructions about the definitions of valid and invalid assignments (Section 6.2).

Due to different interpretations of the short messages, the concepts or the definition of a relation between these two and the naive classifier, evaluation errors could occur that affect the quality of the evaluated data set. Types of errors that could occur are:

- A concept that was never assigned to a user profile by the naive classifier that is based on matching terms (Chapter 5), will never be assigned during the manual evaluation. Only assignments made by the naive classifier were judged.

- Judges evaluate the concept assignments based on the messages that were matched by the naive classifier. Based on this information the classification could be evaluated as invalid, while based on the complete message stream the classification is valid.

- Judges can make mistakes, because they click on the wrong bullet, misinterpret messages, or the guidelines.

The next section describes the used guidelines and Section 6.3 evaluates the reliability of the evaluation process using statistics.

Figure 6.2: Screenshot of the evaluation tool

6.2 Judgment guidelines

The judgment guidelines support objective decision making about whether an assigned concept to a user based on a message stream is valid or invalid. We defined some rules about when there is a relation between a concept and a user that is interesting for discovering groups. The main rule is that when the user explicitly refers to a positive relation between him and the concept, the assignment of the concept is valid. For example, the message 'I like swimming', describes a positive relation from the user to the concept 'Swimming'. However, many messages do not contain an explicit description of a relation user with a concept, while the information could be interesting for discovering groups.

General messages, negative messages or messages related to someone else, like 'When does the swine flu vaccination program start?', 'I don't have the swine flu', and 'My mother has the swine flu and broke her ankle.', do not describe explicit positive relations between the user and the concept 'Influenza'. However, when a user publishes these messages he is quite interested in 'flu'. When the message stream of a user contains two or more messages related to the same concept, we consider the assignment of this concept as valid. This is based on the assumption that if a user spends more time on writing about a specific concept, he is interested in it [5].

Concepts could have an ambiguous meaning and occur in multiple parts in the Wikipedia graph for different reasons. For example, a user profile with a message containing the term 'alcohol' can get assigned the categories 'Antiseptics' (disinfectant to clean things and kill bacteria) and 'Nutrition' (alcoholic drinks). Which assignment is valid depends on the context of the term 'alcohol' in the message.

The parent-child relations in the Wikipedia category structure do not imply 'is-a' relationships (see Section 4.3). This means that when a lower level con-

cept assigned to a user profile is evaluated as valid, this does not imply that assignments of the parent concepts are valid. However, sometimes other type of relation between the concepts result in a valid assignments, for example: 'I take a shower' gets the higher level concept 'Hygiene' based on the word 'shower'. This category is valid, even when there is no 'is-a' relationship between 'Hygiene' and 'shower'.

It occurs that messages with vague information result in the assignment of concepts that are specific. For example, the message 'Au, my foot hurts' could result in the assignment of the concept 'Foot disease' and 'I'm eating a lot of fast food' in the assignment of 'Eating disorder'. When there is no explicit information about a 'disease' or 'disorder', the message is too vague and the assignment is considered as 'invalid'. The concept could also be too specific, for example when the concept 'Influenza vaccines' is assigned to the profile based on the message 'I don't have the swine flu'.

Every judge read the judgment guidelines (Appendix A) with these rules, before starting with the evaluation process.

6.3 Inter-annotator agreement

There are several methods to measure the agreement between judges (or annotators). An often used agreement coefficient for annotation tasks with categorical data is Cohen's kappa [3]. This coefficient shows the agreement between annotators and the reproducibility of the evaluated data set. If different annotators produce similar results during the evaluation of the collection, then we can consider that they have similar understanding of the judgment guidelines and short messages. However, a good agreement does not necessarily ensure validity, because they both can make the same mistakes and misinterpretations.

In order the measure the agreement, one judge repeated the evaluation of 12 randomly picked profiles that were judged by 8 other judges, without knowing the classification the first judge made. These 12 profiles had a total of 120 assignments (**i**) of categories to them that were judged twice.

The kappa statistic distinguishes two types of agreement: the observed agreement and the expected agreement. A proportion of the observed agreement could be caused by chance. The observed value is corrected for chance using the value of the expected agreement. In this evaluation two types of classification were possible: $c \in \{valid, invalid\}$. Table 6.1 shows the number of classifications of type c by judge j (n_{jc}). The observed agreements are based on the number of times both judges evaluate the assignment to the same class:

$$A_o = \frac{41 + 71}{120} = 0.93$$

We take into account the expected agreement based on change, by using the probability distribution ($\hat{P}$) for each judge and each type of classification:

$$\hat{P}(c|j) = \frac{n_{jc}}{\mathbf{i}}$$

The probability that a judge j_1 evaluates a concept to user profile assignment to the same category c as judge j_2 is measured as $\hat{P}(c|j_1) \cdot \hat{P}(c|j_2)$. The sum of this probability for each category results in the expected agreement.

j		Judge 1	
	c	valid	invalid
Judge 2	valid	41	4
	invalid	4	71

Table 6.1: Evaluation by two judges

$$A_e = \hat{P}(valid|j_1) \cdot \hat{P}(valid|j_2) + \hat{P}(invalid|j_1) \cdot \hat{P}(invalid|j_2) =$$

$$\frac{45}{120} \cdot \frac{45}{120} + \frac{75}{120} \cdot \frac{75}{120} = 0.53$$

The kappa coefficient $\kappa$is calculated as:

$$\kappa = \frac{A_o - A_e}{1 - A_e} = 0.86$$

The kappa value measures the strength of agreement. However, it is not defined how to interpret the value in relation to the reliability of the evaluated data set. The best-known convention concerning the interpretation of kappa coefficient values is proposed by Landis and Koch [29]. They proposed that a value of $\kappa > 0.8$ could be considered as reliable. Based on this measurement, we consider that the evaluation process of the data set in this research is reliable and reproducible.

6.4 Results of the naive classifier

This sections explains how the performance of naive classifier is measured using the results of the manual evaluation by the judges (Subsection 6.4.1) and discusses the results of the naive classifier and the evaluated collection (Subsection 6.4.2).

6.4.1 Performance measuring

Precision and recall are measures that are often used to measure the performance of a classifier. They measure the correctness and the completeness of the classification. During the judging process, only the assignment of concepts to user profiles made by the naive classifier are evaluated. The naive classifier considers every assignment as a valid assignment.

We measure the precision (the correctness) based on the assignments made by the naive classifier and evaluated by the judges using the following function:

$$Precision = \frac{correct\,assigned\,concepts}{total\,assigned\,concepts}$$

The concepts that are not assigned to user profiles could have valid relations to the user profiles, however this is not measured. To measure the recall we are limited to the number of assignments that are validated as correct (7,035). The recall is measured as:

$$Recall = \frac{correct\,assigned\,concepts}{possible\,total\,of\,correct\,assigned\,concepts\,(7,035)}$$

We also use precision and recall to measure the performance of the improved classifier that is presented in the next chapter. Section 8.1 explains how these values are calculated in the context of the improved classifier.

6.4.2 Results

The result of the manual evaluation is a validated data set of 1,503 Twitter profiles with short messages, and Wikipedia concepts in the health domain assigned to the profiles. For every assignment made by the naive classifier (18,798 in total) a judge marked this assignment as valid or invalid. Table 6.2 shows the properties of the evaluated data set.

Table 6.3 shows the results of the naive classification process based on the evaluation data. Matching based on page titles (assignment type page) obtain the most retrieved messages (9,180), however this results also in the most invalid assignments of the related concepts the users (62.8%). The concepts that are assigned by matching on category titles (assignment type base) result in fewer errors. Compared to the category titles, page titles have an ambiguous meaning more often, which results in more classification errors.

In Wikipedia, titles with an ambiguous meaning are distinguished by a suffix between brackets, like 'Iron (appliance)', to distinguish removing wrinkles from fabric from the metallic chemical element and a type of golf club. From the stored 29,798 stored page titles related to concepts, 296 have a suffix, while none of the 535 category titles have a suffix. Due to the ambiguous meaning of page titles words, the related concepts could be assigned to the user profile, while the words in the message have a different meaning. When the queries generated from Wikipedia article titles that are marked as ambiguous by the Wikipedia community, are left out from the matching process (second row in the table), this results in 2,118 less assignments of concepts to profiles, while the number of correctly assigned concepts only decreases with 169 (2.4%). Ambiguous words affect the precision of the assignments with 3.8%. Besides the ambiguous words, words in titles could also be used in a figurative sense. In that case the word 'headache' does not have to mean that there is a relation to concepts 'symptoms' and 'pain'.

From the 281 concepts assigned by the naive classifier, 210 are correct at least once. That means that 71 concepts, assigned 279 times to 205 profiles, always result in an error (1.5% of the errors). These type of errors are the result of differences in semantics (ambiguous meaning, figurative) of words in the concept structure and in the short messages.

Root assignments, based on matching of page and category titles with messages and assigning the related root concept, show less errors compared to the other assignment types (60.7 % is valid). These concepts have a broader meaning, which results in a higher probability that they have a relation to the user profile.

The naive classifier assigns every concept that is matched to a message of a user profile (Chapter 5). Properties of the concepts, the user profile or statistics related to the assignments, such as the matching score and the number of times

Variable	Value
Number of user profiles	1,503
Number of messages	129,393
Number of distinct matched messages	10,668
Unique terms	88,837
Associated concepts	18,798
Correct associated concepts	7,035
Avg. number of concepts per profile	12.5
Total concepts	535
Total paths to concepts	792
Total assigned concepts	281
Total valid assigned concepts	210

Table 6.2: Properties of the manually validated data set

a concept was assigned, are not taken into account. To improve the classification results of the naive classifier we use machine learning techniques that use these properties to decide whether the assignment is correct or not. Chapter 7 describes the methods and properties that are used to improve the classification results.

Assignment types	**Profiles**	**Retr. Messages**	**# distinct concepts**	**Distinct # correct concepts**	**# assign-ments**	**% assign-ments**	**# Valid**	**% Pre-cision**	**% Re-call**
All (page, base, parent, root)	1,503	10,668	281	210	18,798	100.0	7,035	37.4	100.0
All without ambiguous page titles	1,491	9,519	273	206	16,680	88.7	6,866	41.2	97.6
Page	1,473	9,180	231	171	9,659	51.4	3,595	37.2	51.1
Base	1,285	4,186	123	93	2,871	15.3	1,613	56.2	22.9
Parent	1,285	4,186	79	64	3,203	17.0	1,638	51.1	23.3
Root	1,285	4,186	4	4	2,094	11.1	1,272	60.7	18.1
Page, base	1,503	10,668	268	197	11,251	59.9	4,424	39.3	62.9
Base, parent	1,285	4,186	171	133	5,969	31.8	3,163	53.0	44.9
Page, base, parent	1,503	10,668	281	210	17,864	95.0	6,794	38.0	96.9
Base, parent, root	1,285	4,186	171	133	7,199	38.3	3,813	53.0	54.2

Table 6.3: Performance of naive classifier using different types of assignments

Chapter 7

Improved user profile classification

To improve the results of the naive classification using the naive classification technique, we use a set of additional features and Support Vector Machines (SVM) models (Figure 7.1). Because this research has a focus on developing a generic method for discovering groups using hierarchical concepts, the model should not depend on training on features that are specific for the content of the concepts in the hierarchy and the messages that we are using. Training on the features related to the content of the messages or the concepts, for example the occurrence of the 'shower' in a message results always to a valid classification of the concept 'Hygiene', will result in a model that is only useful for classification of user profiles with concepts in the health domain, while we want to support classification using other hierarchies.

In Section 7.1 we refer to related work for building classification models. We use the Support Vector Machines library LibSVM and the WEKA Toolkit to build classification models using the validated collection and features based on additional information gathered during the assignment process of concepts to user profiles. The classification model should be able to decide whether these assignments are valid or not. We use the manually evaluated collection with 18,798 of these assignments for training and validating the classification model.

Section 7.2 presents features that are not related to specific categories and are used to build a classification model. These features have a relation to the concepts, the message stream and the number of occurrences of concepts in message stream. Combinations of these features that are used to build classification models are described in (Section 7.3).

7.1 Related work

Research related to classification of text data into a set of pre-defined categories rely often on using a training set of documents that are assigned to their true category (supervised learning). Based on the training data a classification algorithm creates a model for classification of new documents. In this research, we focus on assigning hierarchical concepts to user profiles, which is a classification

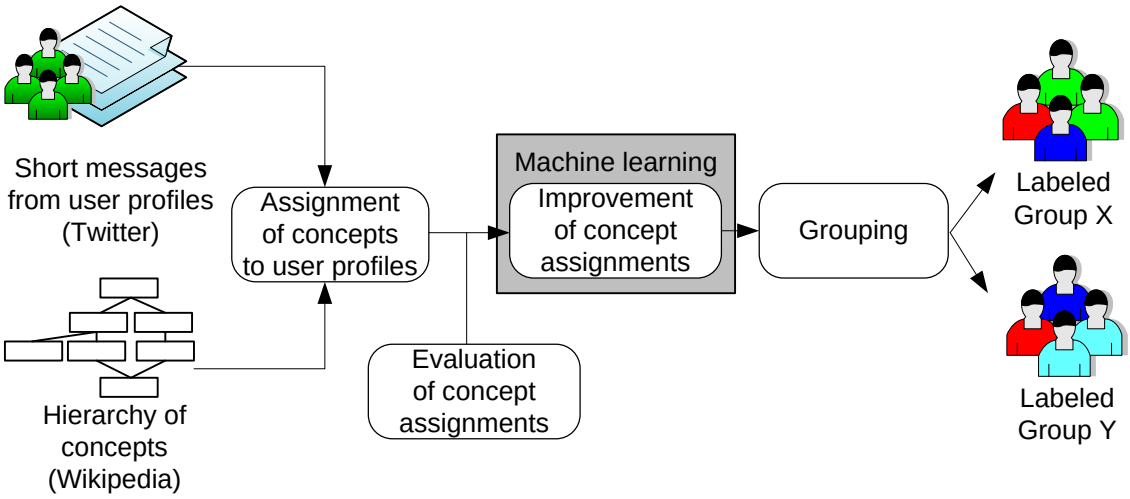

Figure 7.1: Improved classifier in the text mining process

problem of classifying user profiles into multiple concepts. Classification models for text classification are often trained on features based on word occurrences. For example, a document that contains 'sport', 'tennis', 'racket' and 'backhand' has a high probability that is belongs to the concept 'Tennis'. However, in this research we focus on generic classification model that is not trained on features related to specific concepts, to be able to change the set of concepts without creating a new model.

Joachims [26] describes that in text categorization, documents are assigned to one category, multiple categories or no categories at all. In this case the categorization problem could be considered as a multiple binary classification problem for each combination of a document and class or concept: a document belongs to a concept or not. Using the naive classier described in Section 5.2 we make an initial assignment of concepts to user profiles. During this process, statistical data that is independent of the meaning of the concept could be gathered. This information could be used to create a classification model that is able to decide whether the concept belongs to the user or not. For example, an improved model could assign a concept to a user profile when a concept is assigned twice or more to a user profile by the naive classifier.

Many researchers compared different classification algorithms in the domain of text classification. Often used algorithms are Naive Bayes, Bayesian Networks, Decision Trees and Support Vector Machines (SVM). SVMs have been shown good results for text classification by Dumais et al. [11], Dhillon et al. [9], Weiss et al. [48] also in the context of hierarchical classification (Dumais and Chen [10] and Sun and Lim [45]). However, in these approaches the classification model is based on term frequencies of all terms in the documents. In our research, we reduced the input data for the model to concepts and matches to the user profiles.

SVM is an inductive learning scheme for two-class classification problems. The method is defined over a set of features where the classification problem is to find the decision surface that separates the feature values in a vector of one

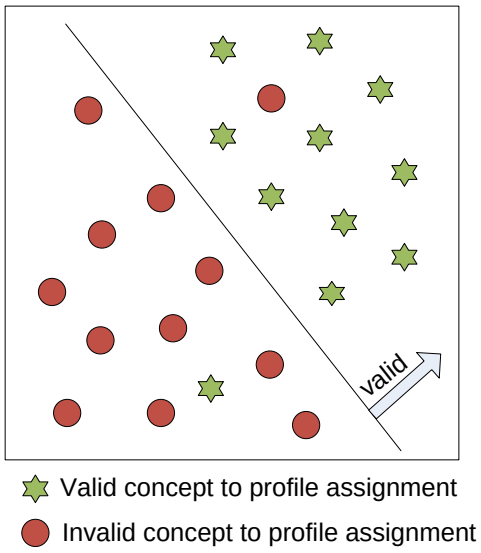

Figure 7.2: Linear Support Vector Machine

class from the other. Figure 7.2 shows a simple example where the data points that belong to two classes are separated by a hyperplane. This hyperplane, that is learned from training data with positive and negative samples, maximizes the margin between the classes. In this example the data is separated linearly in two dimensions, however SVM algorithms support high dimensional data (vectors) and polynomial and radial basis functions, to separate the classes based on data points that consist of many features and are separated in a more complex way. Based on the SVM the classifier calculates on which 'side' of the decision boundary a new input vector occurs to classify the object: in this research the relation between a concept and a user profile.

Meij et al. [32] use classification algorithms for the classification of relations between query strings created by a user for a search action and DBpedia concepts for semantic query suggestion. The process they use consists of two stages. In the first stage, they retrieve candidate concepts from DBpedia based on the query string. In the second stage, they used supervised machine learning to decide which of the candidate concepts should be kept as viable concepts for the user query. They considered the machine learning algorithms Naive Bayes, Decision Trees and Support Vector Machines. As input for these algorithms, they used a set of features related to the concepts and the query. This process is very similar to decide whether a relation between a user profile and a concept is correct or not. The evaluation of this process in Meij et al. [32] shows the best results when SVMs are used for this classification task. Other classification algorithms show worse results.

Because other researches show good results when using SVMs for classification tasks in similar contexts, we focus only on SVMs for the improvement of the results from the naive classifier.

7.2 Features

SVM models are trained on features: measurable (numeric) properties that have a relation to the data. Based on these features the SVM learns when to classify a concept related to a user profile as valid or invalid. Table 7.2 shows the variables and features that are used in this research. Variables contain information that is not directly used as a feature, however they occur as part of functions of the features.

Table 7.1 shows the meaning of the symbols that are used for describing the features in Table 7.2. Appendix B explains the notation style that is used for describing the features and gives a schematic overview of the relation between the different sets in Table 7.1. The schematic overview (Figure B.1) also shows the relation between *concepts* (derived from category titles) and their *related terms* (derived from page titles) in relation to the sets that are used for calculating the feature values.

We distinguish three types of variables and features: related to profiles, concepts or the assignments of concepts to user profiles. These variables and features do not have domain specific relations to the content of the messages or the concepts. Otherwise the SVM is possibly trained on properties that are related to the 'Health'-domain, and in that case the model is not usable for concept hierarchies from other domains. In this section, we give describe the features and why we assume or expect that they are useful for predicting the correctness of the classification made by the naive classifier.

7.2.1 Profile related features

The value of a profile feature is based on characteristics of a user profile. The value is not dependent to a specific combination of a profile and a concept, it has always the same value for the profile p.

Messages in the profile that have a relation to a concept ($AssignedMessages(p)$, $TotalTerms(p)$)

The number of messages in a profile that have a relation to a concept according to the naive classifier is measured by $AssignedMessages(p)$. This is an indication of the overlap between the profile and the concepts. The sum of the terms in these messages is measured by $TotalTerms(p)$. If a user publishes many messages that are covered by concepts the values of these variables become higher.

Ratio of messages that have a relation to a concept ($SelectedRatio(p)$)

If a user publishes many messages that are covered by concepts in the hierarchical structure this ratio gets a higher value. To be able to compare the $AssignedMessages(p)$ value of different profiles, this number is divided by the total messages in the profile.

A low value for this ratio means that only a few messages in the profile got a match with a concept. When a user publishes more messages that have a match with a concept compared to other users, this might increase the probability that these relations between the concepts and the profile exists.

7.2.2 Concept related features

Concept related features have the same value in relation to different profiles if the concept c is the same. The value is based on characteristics of the concept in the concept structure or occurrences in the messages collection. The value is not specific for a relation between the concept and a user profile.

Weighting of the used queries for assigning concepts ($MatchedQueries$, $QueryFrequency(c)$, and $InverseQueryMatchingFrequency(c)$)

Assignments of concepts are based on matching category and article titles that have a relation with the short messages. A query generated from a page title results in a high number of concept assignments to user profiles, if this page title is very generic (e.g. 'joy') and occurs very often in the short messages. The values of $MatchedQueries$, $QueryFrequency(c)$, and $InverseQueryMatchingFrequency(c)$ are used to discover these type of assignments. $InverseQueryMatchingFrequency(c)$ measures this value in relation to the total number of query matches during the naive classification. $InverseQueryMatchingFrequency(c)$ becomes lower when a concept is assigned to many profiles by different retrieval actions using the generated query. This could be an indication that the concept often does not have a relation to a profile because it is assigned via terms that have an ambiguous meaning or occur very frequently in texts.

The number of words in a concept ($WordsInConcept(c)$)

There is a higher probability that concepts with many terms are more complex concepts. This could affect the probability that the assignments to user profiles with these concepts are correct or incorrect. Concepts, such as 'Ailments of unknown etiology', 'Human MHC mediated diseases', 'Insect vectors of human pathogens' 'Vegetarian companies and establishments', could be to detailed to result in a valid assignment to a user profile.

Depth of the concept in the Wikipedia structure ($Level(c)$)

Concepts that are near the selected root concept of the Wikipedia structure, have a broader meaning, which results in a higher probability that it covers the concepts in the messages. Concepts with a higher distance between a concept and the root concept are often have a more specific meaning.

7.2.3 Assignment related features

Assignment features are based on the combination of profile p with concept c. The value depends on characteristics of the profile and the concept together. Different combinations of c and p could result in different feature values.

Number of relations between a concept and a profile ($mf(c,p)$)

When a concept is assigned to a profile (based on term matching) more often, for example when a user publishes multiple messages in which the concept terms occur, the probability that the relation is valid increases. However, when the number of assignments becomes very high, this might be caused by a concept (or related terms) that has multiple meanings. In that case, the probability that the relation is invalid, increases.

We distinguish different types of counting the number or relations (mf): type and term. Type refers to the assignment type that is used by the naive classifier. For example, when the concept c is assigned two times to profile p by a parent-assignments then $mf_{parent}(c, p) = 2$.

'Term' refers to a term that occurs in the messages that are matched during the matching process of concept c to profile p. These general terms, do not have relations to specific concepts. Examples of terms are: I, me, we, going, today, http.

Weighted number of relation between a concept and a profile ($mfipf(c, p)$)

In order to weight the importance of the n values mentioned in the previous paragraph, we use a function based on traditional term-weighting scheme that is used in information retrieval: tf-idf. Tf-idf (term frequency-inverse document frequency) weights how important a term is to the document in a collection [31][32]. When a term has a high term frequency in a document, but also occurs very often in other documents, this term is not very important in the collection.

In this research, term frequency is translated to message frequency ($mf(c, p)$), as in a value based on the number of times a concepts was assigned to a profile. We call the *idf* part, the inverse profile frequency ($ipf = log(\frac{ProfileCollectionSize}{cf(c)})$, which measures the importance of the concept in the profile collection. Weighting the message frequency of a concept results in the $mfipf(c, p)$ feature.

These values could also be measures using different assignment types and different terms, using the corresponding *mf* and *cf* values to weight the message frequency.

Matching score of the naive classifier ($...Score(c, p)$)

Each assignment of a concept to a user profile is based on retrieving short messages using a query generated from concept related information. The retrieved short messages by the query are ranked based on the outcome of the scoring function discussed in Subsection 5.3.3. The assignment of a concept to a user profile could result in multiple scores ($Score(c, m)$). To calculate a score that is related to the assignment of a concept to a user profile, three different scoring types are used: the total score, the average score and the maximum score. Higher rankings of assignments of concepts to user profiles based on term matching could have a higher probability of a correct classification.

Duplicate concept assignments by the same message ($Duplicate(c, p)$)

There are concepts that occur in multiple paths in the Wikipedia graph, because they have multiple parents (e.g. 'Headache' in Figure B.1). These concepts often have a more generic meaning. In addition, there are also concepts where the parent concepts contain a subset of the terms compared to child concept or related terms (e.g. 'pain' and 'neck pain'). Due to multiple paths in the concepts structure or overlapping terms between concepts, these concepts would be assigned multiple times to the same profile, based on the same message. However, the other features (such as *mf*) count the number of assignments of concepts to user profiles based on distinct messages.

To take into account that an assignment of a concept with a broader meaning to a profile, has a higher probability to be a correct assignment, the $Duplicate(c, p)$ feature counts the number of times a concept would be assigned to a profile a

profile based on matching terms that are related to the concept c to the same message in the profile p.

Number of assignments with messages containing terms ending with -ing ($Ing(c, p)$)

The -ing suffix is added to English verbs to make a present active participle. These types of words in messages are often used when users describe activities of what he or she is doing. This type of messages could possible affect the probability that there is relation between a user and a concept. However, there are also words ending with -ing that are not verbs, like 'morning' or 'evening' that affect the value of this feature.

Number of assignments using a query containing ambiguous page title ($Ambiguous(c, p)$)

When a concept is assigned to a user profile based on a query that is generated from terms in a Wikipedia page titles that is marked as a title with an ambiguous meaning (by the Wikipedia community), this leads to errors (see Section 6.4). $Ambiguous(c, p)$ measures the number of occurrences of these type of assignments, which could be used in the classification model to decide whether the relation between to concept and a profile is valid or not.

Importance of the concept in the profile ($SelectedConceptFrequency(c, p)$, $ConceptRatio(c, p)$)

If a concept is assigned more frequently to a profile than other concepts, the values of $SelectedConceptFrequency(c, p)$, $ConceptRatio(c, p)$) become higher. This happens when the concept occurs on a higher level in the concept structure and is assigned as a parent of (multiple) other lower level concepts. In addition, when the concepts do occur more frequently, the probability that the relation is (eventually) valid, increases, except when the assignment is always based on matching an ambiguous words that has a relation to concept c, and occurs frequently is messages because it has a generic meaning. In that case a high value is an indication of wrong assignments of the concept to the user profile.

Importance of concept matches ($MatchingFrequency(c, p)$)

Chapter 6 showed that page assignments result in a low precision, but are responsible for many assignments of concepts to profiles. When a concept is matched based on a page title, and this title is a word that is very frequently used in short messages, while the concept related to this page title has a different meaning, this results in classification errors. However, when different page titles lead to the assignment of the same concept, the probability that the concept is valid increases. $MatchingFrequency(c, p)$ measures how often this happens. If the assignment of the concept 'Hygiene' in based on matching terms with two different messages in the profile, $mf(Hygiene, p) = 2$, When this is based on matching the words 'shower' in both messages, $MatchingFrequency(c, p) = 1$, while if the concepts is assigned by two different matches (e.g. 'shower' and 'washing') the value is 2.

7.3 Feature sets

Combinations of features are used to train a classification model. To determine the features that affect the existence of the relation between a concept and a user profile automatically, we use a correlation-based feature selection algorithm. The selection of a relevant feature subset is based on a set of features that highly correlate with the class (valid or invalid assignment). Hall [20] describes an algorithm to select a correlation-based feature (sub)set (CFS):

"*CFS is a simple filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features should be ignored because they will have low correlation with the class.*"

We applied the CFS algorithm on our set of features. Table 7.3 shows the result set of features. The terms of the $mf_{term}(c,p)$ features are selected based on common terms in the collection (see Subsection 7.3.1).

We use different features sets based on the features that occur in the correlation-based feature set. The $mf(c,p)$ and $mfipf(c,p)$ features that occur in the correlation-bases feature set (Table 7.4) are expanded, to a set where the *term* and *type* variants of the features both exists in mf and $mfipf$ form. This set (Table 7.5) is used as the base feature set for training a SVM model.

To discover the effect of the different features on the classification results we add features incrementally to the base set. The next chapter shows and discusses the classification results of the models created based on these different feature sets.

7.3.1 Common terms

The features discussed in the previous section, have a relation to the concepts. The classification is based on matching terms related to these concepts with terms in the short messages. However, other terms in that occur in the short messages could also affect the correctness of the classification made by the naive classifier. To discover if there is correlation between terms that occur in the short message collection and the correctness of the classification we selected terms in the short messages collection that are not part of the matching process by the naive classifier.

The risk of introducing features that are based on common terms, is that the classification model will be trained on characteristics of our data set. However, to evaluate the effect of introducing more specific features on the quality of the classification model, we use a selection of common terms. In the next chapter we discuss if introducing additional features based on common terms affect the classification results positively.

To reduce the number of irrelevant and rare terms, we selected terms that occur 20 or more times in the short message collection. For all these terms we used the CFS algorithm to select a subset of 70 terms based on the $mf_{term}(c,p)$ values and validation value of the manual evaluation of the naive classifier. This subset contains the following 70 terms:

Symbol	Meaning
c	A concept in the Wikipedia concept structure. This concepts could consists of multiple words.
C	Collection of matched concepts. $C = \{c_1, c_2, ..., c_n\}$
q	Term(s) related to a concept according to the hierarchical structure of Wikipedia, that is used as query in the matching process of the naive classifier. This could be the terms in the concept itself or the page titles related to the concept.
Q_c	The set of the (groups of) terms related to the concept c that are used by the matching process of the naive classifier to assign the concept to a user profile. $Q_c = \{c, q_1, q_2\}$ $Q_{hygiene} = \{`hygiene', `shower', `hair\ care'\}$
O_q	The number of paths from the root node to the page or category title in the Wikipedia structure, that is used as source for generating q.
m	A message in a user profile.
p_{user}	A user profile with messages. $p_{user} = \{m_1, m_2, m_3\}$
P	Collection of profiles. $P = \{p_1, p_2, ..., p_n\}$
M_p	Messages in the user profile p that matched with a concept. $M_p = \{m_1, m_2\}$, where $m_n \in p$
CM	Concept matches, set of distinct$(c, q, m, type)$ tuples. Where c is the matched concept, q the terms $(q \in Q_c)$ that matched with the message m, $type$ the assignment type that is used $(type \in \{page, base, parent, root\})$.

Table 7.1: Symbols and their meaning[1]

{i've, that'll, must, keeping, finished, liking, quick, editing, off, hoping, today, http, i, am, my, me, i'm, went, going, some, day, days, youtube, blogspot, happy, meeting, love, kids, children, watch, watching, working, work, mom, likely, like, have, blogging, rt, www, tinyurl, us, our, your, they, this, with, yesterday, on, can, are, him, his, their, her, morning, ly, not, don't, doesn't, wouldn't, bye, office, addthis, haven't ain't, noise, appear, beautiful, always}

Results of using these features together with the features in Table 7.3, are also discussed in the next chapter.

[1] See Appendix B for a schematic overview of the symbols and sets.

Variable or feature	Description
Profile related	
$ProfileCollectionSize$	$\|P\|$ Total number of profiles in the collection (= 1,503).
$ProfileSize(p)$	$\|p\|$ Total number of message in the profile p.
$AssignedMessages(p)$	$\|M_p\|$ Total number of distinct message retrieved from user profile p during the assignment of concepts.
$SelectedRatio(p)$	$\frac{AssignedMessages(p)}{ProfileSize(p)}$
$TotalTerms(p)$	$\sum_{m \in (p \cap \{mm \in CM\})} WordsIn(m)$ Total number of terms in messages retrieved from user profile p during the assignment of concepts.
Concept related	
$ConceptCollectionSize$	$\|C\|$ Total number of matched concepts (= 281).
$MatchedQueries$	$\|CM\|$ Total number of time a query matched a profile (= 25,616).
$cf(c)$	$\|\{p \| (mc, mq, mm, mtype) \in CM, p \in P : c = mc \wedge mm \in p\}\|$ Concept frequency, the total number of profiles that have concept c assigned to it.
$cf_{type}(c)$	$cf(c)$, where *type* is the type of assignment (page, base, parent or root, see Subsection 5.3.2).
$WordsInConcept(c)$	Number of words in the string of concept c.
$QueryFrequency(c)$	$\sum_{q \in Q_c} \|\{mq \in CM \,;\, q = mq\}\|$ Number of queries that results in an assignment of concept c to a profile.
$InverseQueryMatching-Frequency(c)$	$\frac{MatchedQueries}{QueryFrequency(c)}$
$Level(c)$	Distance of the Wikipedia concept c to the main node.

Variable or feature	Description
Assignment related	
$mf(c,p)$	$\lvert\{(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p\}\rvert$ Number of messages in the profile p that are related to the concept c.
$mf_{type}(c,p)$	$\lvert\{(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge type = mtype \wedge mm \in p\}\rvert$ Number of message in the profile p that are related to concept c, where type is the type of assignment.
$mf_{term}(c,p)$	$\lvert\{(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p \wedge mm \,\text{contains}\, term\}\rvert$ $mf(c,p)$ where the assignments are counted when the retrieved message contains the term $term$.
$mfipf(c,p)$	$mf(c,p) \cdot log(\frac{ProfileCollectionSize}{cf(c)})$[31]
$Duplicate(c,p)$	$\lvert\{(mc1,mq1,mm1,mtype1) \in CM\,,\,(mc2,mq2,mm2,mtype2) \in CM \,:\, c = mc1 \wedge mc1 = mc2 \wedge mm1 \in p$ $\wedge mm1 = mm2 \wedge mq1 \neq mq2\}\rvert + \sum_{q \in \{mq \mid (mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p\}} (O_q - 1)$ Number of possible duplicate assignments of the concept c to profile p based on the same retrieved message.
$Ambiguous(c,p)$	$\lvert\{(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p \wedge mq \,\text{is ambiguous according to the Wikipedia community}\}\rvert$ Number of assignments of concept c to profile p, caused by queries generated from article titles that have an ambiguous meaning according to Wikipedia.
$Ing(c,p)$	$\lvert\{(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p \wedge mm \,\text{contains a term ending with -ing}\}\rvert$ Sum of $mf(c,p)$, where terms are all words ending with -ing.
$SelectedConcept\text{-}Frequency(c,p)$	$\frac{mf(c,p)}{AssignedMessages(p)}$ Fraction of messages from the profile that is used during the assignment process.
$ConceptRatio(c,p)$	$\frac{mf(c,p)}{ProfileSize(p)}$
$Matching\text{-}Frequency(c,p)$	$\lvert(mc,mq,mm,mtype) \in CM \,:\, c = mc \wedge mm \in p \wedge mq \wedge mtype\{page,base\}\rvert$ Number of queries that resulted in a match of concept c to profile p.
$Score(c,m)$	Retrieval score (Subsection 5.3.3) of message m from the index, using a query generated from concept c.
$TotalScore(c,p)$	$\sum_{m \in p} Score(c,m)$
$AverageScore(c,p)$	$\frac{TotalScore(c,p)}{AssignedMessages(p)}$
$MaximumScore(c,p)$	$\max(Score(c,m)$ for all $m \in p)$

Table 7.2: Definition of variables and features

Correlation-based feature subset	
$mf(c,p)$	$TotalScore(c,p)$
$mf_{base}(c,p)$	$MaximumScore(c,p)$
$mf_{i}(c,p)$	$Duplicate(c,p)$
$mf_{i\,am \text{ or } i'm}(c,p)$	$Ing(c,p)$
$mfidf_{page}(c,p)$	$mf_{that'll}(c,p)$
$mfidf_{base}(c,p)$	$mf_{today}(c,p)$
$mfidf_{parent}(c,p)$	$mf_{http}(c,p)$
$mfidf_{i}(c,p)$	$mf_{am}(c,p)$
$SelectedConceptFrequency(c,p)$	$mf_{my}(c,p)$
$ConceptRatio(c,p)$	$mf_{going}(c,p)$
$TotalTerms(p)$	$mf_{some}(c,p)$
$ProfileSize(p)$	$mf_{day}(c,p)$
$InverseQueryMatchingFrequency(c)$	$mf_{have}(c,p)$
$MatchingFrequency(c,p)$	$mf_{this}(c,p)$
$QueryFrequency(c)$	$mf_{beautifull}(c,p)$

Table 7.3: Correlation-based feature subset selection

Correlation based $mf(c,p)$ and $mfidf(c,p)$
$mf(c,p)$
$mf_{base}(c,p)$
$mf_{i}(c,p)$
$mf_{i\,am \text{ or } i'm}(c,p)$
$mfidf_{page}(c,p)$
$mfidf_{base}(c,p)$
$mfidf_{parent}(c,p)$
$mfidf_{i}(c,p)$

Table 7.4: $mf(c,p)$ and $mfidf(c,p)$ from the correlation-based set

Base set of $mf(c,p)$ and $mfidf(c,p)$
$mf(c,p)$
$mf_{page}(c,p)$
$mf_{base}(c,p)$
$mf_{parent}(c,p)$
$mf_{i}(c,p)$
$mf_{i\,am \text{ or } i'm}(c,p)$
$mf_{we}(c,p)$
$mfidf(c,p)$
$mfidf_{page}(c,p)$
$mfidf_{base}(c,p)$
$mfidf_{parent}(c,p)$
$mfidf_{i}(c,p)$
$mfidf_{i\,am \text{ or } i'm}(c,p)$
$mfidf_{we}(c,p)$

Table 7.5: Base set of $mf(c,p)$ and $mfidf(c,p)$ features

Chapter 8

Evaluation of the improved user profile classification

This chapter presents the results of improved user profile classification using Support Vector Machines. In Section 8.1 we give an overview of how we evaluate the classification models. Section 8.2 shows and explains the results of the evaluation of the classification models. In Section 8.3 we summarize the most remarking features and feature sets that improve the classification results.

8.1 Evaluation metrics

For the training and validation of the classification model based on Support Vector Machines we used WEKA Toolkit. Using the evaluated collection (Chapter 6) we are able to measure the validity of the models. Figure 8.1 shows a schematic overview of the evaluated collection (the box) and the classification results of a classification model (the ellipse). The left side of the box represent the assignments of concepts to user profiles that were marked as valid during the manual evaluation process. The right side contains the invalid assignments. Inside the ellipse the assignments are classified as correct assignments and outside as invalid assignments. The figure shows the following situations:

- True positives (TP): assignments of concepts to user profiles that are correct and also classified as correct.
- False positives (FP): assignments of concepts to user profiles that are incorrect and classified as correct, also called error of the first kind.
- False negatives (FN): assignments of concepts to that are correct and classified as incorrect, also called error of the second kind.
- True negatives (TN): assignments of concepts to user profiles that are incorrect and classified as incorrect.

The number of occurrences of these situations are used to measure the validity of the classification process. To perform the validation we use the 10-fold cross-validation technique and we measure the precision, recall and F-measure.

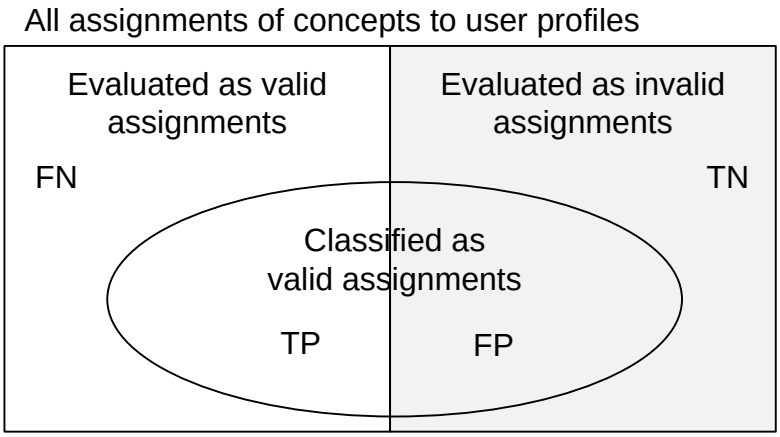

Figure 8.1: Validation of the classification model

8.1.1 K-fold cross-validation

For the training and validation of the classification model using Support Vector Machines, data from the manually evaluated collection is used. However, this training data should not overlap the data that is used for validation of the model. When using the same data for training and validation, the validation results are not reliable. Because in that case the model is possibly trained only for the evaluated collection specifically. To avoid this, there is k-fold cross validation [51]. A fold is a partition of the evaluated collection. Based on a fixed number of k folds, $k-1$ one folds are used for training and one is used for validating the model. When using $k = 10$ (10-fold cross-validation), the evaluated data is split (randomly) into 10 approximately equal partitions. Nine partitions are used for training the classifier, the tenth is used for validation (a fold). During the validation, the classification results are measured. This process is repeated, in order to use all partitions once for validation and the other partitions for training. The final results are based on an average of the results of the ten validations. This 10-fold cross-validation technique is often used, because the number of 10 folds is considered as the right number to get the best estimate of error.

In this research, we train and validate the classification model using the 10-fold stratified cross-validation approach. Stratified means that the folds are selected so that the number of valid and invalid assignments according to the manual evaluation is approximately equal in all folds.

8.1.2 Precision

Precision measures the exactness or correctness of the classification: how many of the assignments that are classified as valid by the classifier are evaluated as valid in the evaluated data set? In Figure 8.1 this is the left part of the ellipse as part of the whole ellipse, which is measured by:

$$Precision = \frac{|tp|}{|tp| + |fp|}$$

8.1.3 Recall

Recall measures the completeness of the classification: how many of the total evaluated set of valid assignments of concepts to user profiles is classified as a valid assignment by the classifier? In Figure 8.1 this is the left part of the ellipse as part of the left side of the box, which is measured by:

$$Recall = \frac{|tp|}{|tp| + |fn|}$$

8.1.4 F-measure

There is often a trade-off between precision and recall [16, 48]. When the precision increases, this often affects the recall negatively and the other way around. In this research, we consider that precision and recall are equally important. Because a bad recall results in an insufficient number of categories related to user profiles, which result in that there is not enough information about relations between concepts to discover groups. On the other hand, if too many concepts are classified as that they have a relation with a profile, while the relation does not exists, this results in invalid groups and user profiles in in wrong groups.

F-measure is a score that is often used to calculate the weighted average of precision and recall [45, 16, 48]:

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

An F-score for of one is the best score, while zero is the worst score. Together with precision and recall, we use the F-score to measure and interpret the results of the classification models created using different feature sets.

8.2 Evaluation of the classifications

In Chapter 7 we introduced features and feature sets to train classification models using Support Vector Machines. In this section, we discuss the effect of the features on the result, compared to our hypothesis that these features affect the classification results. Table 8.1 shows the results classification models using SVM and different feature sets, compared to the naive classifier (first row). The results of the precision, recall, and F-measure are measured based on 10-fold cross validation. Appendix C contains the full table with classification results, including values of the true positives, true negatives, false positives and false negatives.

The values of the features discussed and described in Table 7.2 and Section 7.2 have a relation to profiles, concepts and the assignment of concepts to profiles. We evaluate the effect on performance of the classifier of these features by them the feature set. With performance, we mean the results of evaluation metrics: precision, recall and F-measure.

In the following subsections we discuss the (type of) features that result in an increase of the classification validity (Subsection 8.2.1), small increase or a decrease in performance (Subsection 8.2.2) and the results of using different sets created by the naive classifier (Subsection 8.2.3 and Subsection 8.2.4). In these

subsections the numbers in the headings of the paragraphs refer to the feature sets in Table 8.1.

8.2.1 Features with a positive effect on classification results

Number of relations between a concept and a profile (set 1)

The naive classifier assigns every concept to a user profile where $mf(c,p) > 0$. Compared to the naive classifier, the SVM model assigns the concepts based on another n value. According to the F-score the result is better and has a more balanced precision and recall. The precision increases, while the recall drops. The increase of the precision is important, because it affects the number of profiles that result in right group positively.

(Weighted) number of different types of relations between a concept and a profile (sets 2 and 3)

Introducing features that distinguish the type of assignment, helps to increase the precision, however it has a negative effect on the recall. For the minimal set (2), the F-score is even worse than the simplest feature set (1) and the naive classifier. However, the features in set 3, result in an increase of the precision, resulting in a F-score similar the naive classifier.

The features that measure the occurrence of the terms 'i', 'i'm' and 'i am' in the messages that have a match with a concept, has probably a positive effect on the classification results. According to our definition of a relation between a concept and a user profile that is used during the manual evaluation of the naive classifier (Section 6.2), this features support the creation of a model that is closer to this definition.

Matching score of the naive classifier (sets 4-6)

The features based on the matching score have a small influence on a better recall of the classifier, compared to the base feature set. Because the negative effect on the precision is smaller than the positive effect on the recall, the matching score features give a better F-score.

Importance of the concept in the collection (set 9)

While the features in set 1 to 3 are based on the frequency of concepts in messages, the original assignment of the concept to a user profile is often also based on matching terms related to a concept (in page titles). The $InverseQuery\text{-}MatchingFrequency$, is a weighting factor for the matched queries and has increases the performance of the classifier. The F-score shows an increase of 0.08 compared to the previous feature set, which is caused by the much better recall of the classifier. The importance of the concept in a concept measures by the $InverseQueryMatchingFrequency$ is a good feature for classification of relations between the concept and a user profile.

Frequency of concepts and related terms in the structure (set 11)

Some concepts have a more broader or abstract meaning that is reflected in concept structure, which is explained in the $Duplicate(c,p)$ section in Chapter 7. The $Duplicate(c,p)$ feature measures if a concept would be assigned more often to a profile by matching the same concept with the same messages (see

Table 7.2). The results show that this feature helps to increase the performance of classifier. Precision and recall both increase when adding this feature to the feature set. Concepts with multiple relations in the Wikipedia graph, fit in multiple categories, because they has a broader meaning. In addition, concepts that share terms with terms in their underlying page tiles also have a broader meaning. This feature probably results in a better classification model, because the probability that a concept with a broader meaning has a relation to a user profile is higher.

Correlation-based feature set (set 17)

Compared to the other features sets, the set containing all features selected by the CFS algorithm (Section 7.3), shows the best performance when looking to the F-score (0.68). This feature set has the best recall of all models created using SVMs. Only the feature sets with base and scorings features (3 to 5) showed scores with a better precision, but they have a poor performance in recall. Compared to set 11 there is an increase on the recall. The main difference with this feature set is the addition of mf feature related to some common terms in short messages, such as $mf_{http}(c,p)$ (hyperlinks), $mf_{today}(c,p)$ and $mf_{day}(c,p)$. The discussion of set 18 in the following section discusses the addition of these common terms in more detail.

8.2.2 Features not improving the classification results

This subsection discusses the features that do not show major improvements on the classification results and features that affect the results negatively.

Importance of messages and the concept in the profile (set 8)

Features that measure how important a concept is in the profile and the feature that measures the overlap between the concept structure and the message in a profile show a minor increase of the precision, with a decrease in recall. If a user publishes messages related to a specific concept often and once a message related to another concept, that does not affect the probability that the relation between the concept and the profile exists.

Concepts classification based on ambiguous words (set 10)

The results in Subsection 6.4.2 showed that ambiguous words affect the classification results negatively. Using the number of matches with ambiguous page titles by the naive classifier as a feature does not help to create better model using SVMs. 2,029 assignments of concepts to user profiles are involved with matches of ambiguous page titles. 1,948 of these assignments are invalid. That means that the major part of these type of assignments belongs to the set of negatives. Using this feature trains the SVM on the negatives, which is already the largest part of the training set. Because other features have also good indications of the negatives in the training set, the contribution of this feature to a better classification is minimal.

Frequency of messages with -ing (set 12)

Using this features that measures the frequency concept matches with messages containing words ending with -ing, results in a slightly better precision and

recall. The assumption is that users describe their activities using active verbs and that it affects the probability that there is a relation to a concept. When looking at the performance the effect is minimal. This is possibly caused because using an these verbs does not always mean a valid relation to a concept, and there are other words ending with -ing, which are not verbs.

Level of the concept (set 13)

There is almost no increase in performance when using the level of the concept as a feature. This is may be caused because the level of the concepts also affects the values of other features. The value of level is an integer between 1 and 5. Concepts at level 1 get a lot of root assignments, which is already measured by $mf_{base}(c, p)$ and $mfidf_{base}(c, p)$. Concepts at the deepest levels have fewer parent assignments, which is also measured by mf and $mfidf$ features. These features give a better indication of the relation between a concept and a profile than this concept feature *Level* only.

Common used terms (person related and 'not') (sets 14 and 15)

These feature sets use features with frequencies of certain terms in the matched messages. The terms related to person that published the messages ('i've', 'my', 'me') have a very small positive effect on the performance of the classifier. The features that measures the number of messages with 'not' in it affect the performance negatively.

Profile features (set 16)

To measure the effect of the profile features added in feature set 16, we add these features to the feature set 12, because of the minimal effect of these features on the performance of the classifier. Using the profile features does not improve the classification results. The improvement in recall, results in loss in precision. These features have a strong relation to the profile, just like the features used in set 8. We conclude that characteristics of the profile are not a good indication of the validity of concepts related to the user profiles.

Common domain unrelated terms (set 18)

To discover if there is a relation between common used terms in messages and the validity of the assignment of a concept to a user profile, we selected 70 of these terms (see Subsection 7.3.1). With the remark that using these features could results in training on this specific data set, we use these extra features for training an SVM model. The results show a decrease in performance when using these additional features. The recall drops, while the precision increases slightly. The F-score is worse compared to the correlation-based features set.

These words, that do not occur in the concept structure that is used for the naive classification, do not affect the validity of the relation between concepts and user profiles more than already measured by other features that are more independent of the data set.

8.2.3 Combination of naive classification and SVM

In Subsection 5.2.5 we discussed that ambiguous meaning of words is a problem in text classification. In the results of the evaluation of the naive classifier

(Subsection 6.4.2) we showed that page titles in Wikipedia that are marked as ambiguous affect the performance of the classifier. With feature set 10, discussed in the previous section, we tried to use this information from Wikipedia as a feature for the improved classifier. However, this does not lead to an improvement in the classification results.

When the naive classifier ignores matches between concepts and user profiles based on terms that are marked as ambiguous, this classification results are better. Using these results (second row in Table 6.3 on page 48) in combination with correlation-based feature set to create a improved classification model (set 19), results in a small increase in performance.

Instead of a drop in recall, due to 169 false negatives by the naive classifier filtering out possible ambiguous assignments, there is a small increase in recall. This means that these type off assignments also affect the results of the classification model by the

8.2.4 Leaving out root assignments by the naive classifier

Table 6.3 showed that many assignments are caused by assigning one of the four concepts on the first level of the concept hierarchy as root concept to a user profile. There is a possibility that these concepts assigned using these types of assignments do not contribute to discovering good groups.

We leave out these assignments, using only the page, base, parent assignments by the naive classifier as input for building an improved model (set 20). The classification results using this input, show a similar classification performance when looking to the F-score.

The recall value is corrected for losing 241 valid assigned concepts that the naive classifier ignores. However, when we consider these types of assignments as not useful and the ignorance of all these assignments by the naive classifier not as a problem, the recall is 65.8, which results in an F-score of 0.68.

We conclude that leaving out the root assignments, does not affect the performance when using an improved classification model.

#	Feature set	# features	% Precision	% Recall	F-score
	Naive classifier (binary)	1	37.4	100.0	0.54
1	$mf(c,p)$ (number of assignments)	1	56.6	59.5	0.58
2	$mf_x(c,p)$ and$mfidf_x(c,p)$ (Table 7.4)	8	69.2	41.8	0.52
3	Base set (Table 7.5)	14	70.4	43.3	0.54
4	Base set and $MaximumScore(c,p)$	15	70.8	43.8	0.54
5	Base set and $TotalScore(c,p)$	15	69.8	46.0	0.55
6	Base set, $TotalScore(c,m)$ and $MaximumScore(c,m)$	16	68.9	46.9	0.56
7	and [1] $WordsInConcept(c)$	17	68.4	47.6	0.56
8	and $SelectedRatio(p)$, $ConceptRatio(c,p)$, $SelectedConceptFrequency(c,p)$	20	69.5	47.1	0.56
9	and $InverseQueryMatchingFrequency(c)$	21	68.7	60.6	0.64
10	and $Ambiguous(c,p)$	22	68.7	60.8	0.64
11	and $Duplicate(c,p)$	23	69.5	63.6	0.66
12	and $Ing(c,p)$ (Extended set)	24	69.6	64.0	0.67
13	and $Level(c)$	25	69.3	64.7	0.67
14	and $mf_{i've}(c,p)$, $mf_{me}(c,p)$, $mf_{my}(c,p)$	28	69.8	64.8	0.67
15	and $mf_x(c,p)$, where $x \in \{not, don't, doesn't, wouldn't, haven't, ain't\}$	34	69.8	64.3	0.67
16	Extended set and $AssignedMessages(p)$, $TotalTerms(p)$	26	68.6	66.0	0.67
17	Correlation-based feature set (Table 7.3)	30	69.6	66.5	0.68
18	Correlation-based and common (domain unrelated) terms	90	70.4	62.2	0.66
19	Correlation-based without ambiguous page assignments[2]	30	70.0	66.7	0.68
20	Correlation-based without root assignments[2]	30	71.0	63.9	0.67

Table 8.1: Validation results of the classification models

8.3 Summary

The evaluation of the naive classifier showed that the precision of the classification is low. This could affect the grouping process negatively, because discovering groups based on incorrect information results in invalid groups and users in groups that do not belong to these groups.

In this chapter showed that machine learning techniques could improve the classification results. SVM models based on features related to the concepts and the assignments of concepts to user profiles, show an increase of the precision from 37.4% to 69.6%. However, this increase in the precision results in a decrease to 66.0%. The F-score shows that this is an improvement from 0.54 to 0.68.

Considering the results of classification using SVMs and different features sets, showed that certain features work better than others to improve the results. Features related to the frequencies of concepts in the messages of a profile (mf) and the frequencies of concepts in the collection ($InverseQueryMatchingFrequency$)

[1] The features after 'and' in this row and the rows to set 15 are appended to the feature set in the previous row.

[2] Values corrected for full evaluation set, see Appendix C for details.

and the concept structure (*Duplicate*) help to create a better classification model.

Features related to the characteristics of the user profiles (set 8 and 16), do not affect the classification results very positively. This information about profiles is probably more useful for decide whether the profile is useful or not, but not for classification of the relation between the profile and a concept.

We conclude that the classification could be improved by using SVMs. However, the results are still not perfect. In the next chapter, we show how the classification results are used to discover groups.

Chapter 9

Discovering groups

The final part of the text mining process is the grouping process, which results in labeled groups with users (Figure 9.1). In this chapter, we explore the possibilities of discovering groups based on clustering of concepts related to user profiles. First, we discuss researches related to clustering and discovering knowledge based on text data (Section 9.1). In Section 9.2, we present the research method on how we analyze the discovered groups based on clustering results. The clustering approach using user profiles with concepts as input is described in Section 9.3 and in Section 9.4 we present the extracted groups. In the last two sections, we analyze the results according to the research method (Section 9.5) and discuss the results and the method (Section 9.6).

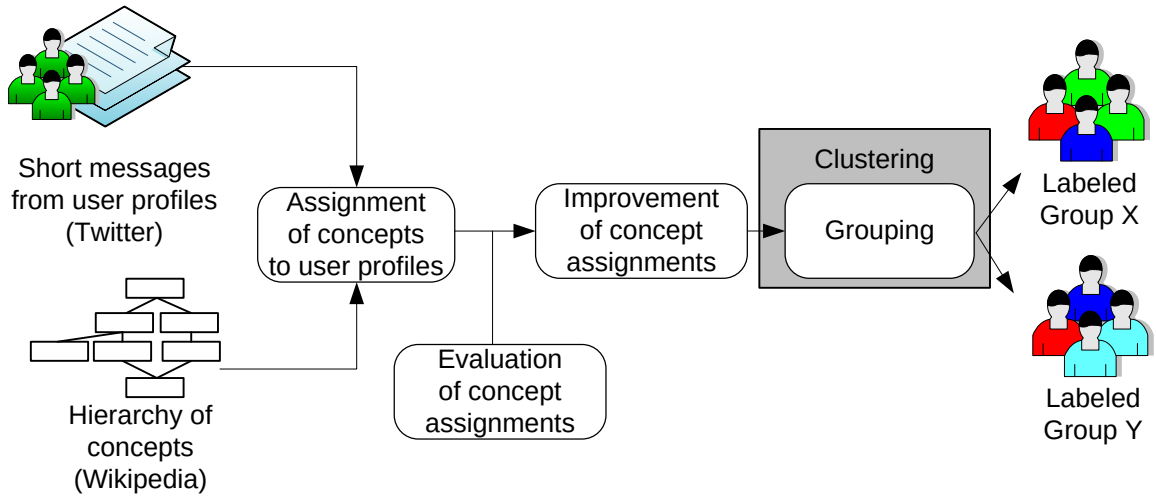

Figure 9.1: Grouping in the text mining process

9.1 Related work

Clustering is an approach that is often used for grouping objects. In the context of text mining, text clustering is used to group similar documents, without defining what the documents need to have in common to become in the same cluster. Feng and Allan [15] use agglomerative clustering to generate a hierarchy (dendrogram) based on term frequencies in documents. They use the dendrogram for the detection of topics in text and analyzing the relations among topics and sub-topics and events for better understanding of these relations. Ponti and Tagarelli [36] use the same clustering strategy to create dendrogram from weighted document term vectors. The different clustering solutions obtained at each level of the dendrogram reflect an organization of the documents into sets of topics.

These clustering approaches use the frequency of terms in documents to discover topics and groups of documents. Cut-offs on different levels of the dendrogram are used to produce results of different granularity. If we translate this approach to our data set, the input would be a set of user profiles with the frequencies of concepts in these profiles (optionally together with terms in messages). This results in groups of profiles, where profiles with similar concepts result in the same group and profiles occur in only one groups. These results will only tell something about the specific collection of profiles, and not about the relations between concepts in the profiles. Because we want to discover (new) groups and describe them using the introduced hierarchical concepts we focus on a more knowledge discovery approach.

El Sayed [12] discusses methods for knowledge acquisition from text. Their approach is based on extraction of terms from documents, to detect groups of terms sharing a relation. These terms form hierarchical related concepts. The relation between terms is discovered based on statistics of these terms in the collection: the co-occurrences of terms in documents. Agglomerative clustering is used to extract concepts from terms that have a relation. We could translate this approach to our data set, by measuring the co-occurrences of concepts related to user profiles. Applying a clustering algorithm on this information results in groups (clusters) of concepts that have a relation.

Because we focus on discovering and describing groups and less on dividing the individual profiles in groups, we use the approach of El Sayed [12]. The relation of the user profiles to these groups is based on the occurrences of the group concepts in the profiles.

9.2 Research method

Our research method to discover groups based on the occurrences of concepts in user profiles (similar to El Sayed [12]), starts with applying a agglomerative clustering algorithm on the data set of concepts related to profiles. The output of the algorithm is a dendrogram, where concepts are grouped together, based on the co-occurrences in profiles. Due to the approach of assigning hierarchical related concepts, concepts that have a hierarchical relation often co-occur in profiles. This would help to discover semantic relation between profiles. However, to discover groups of unexpected relations between concepts, we focus on

clusters with concepts that do not have a parent-child relation in the Wikipedia concept hierarchy.

When two concepts in a cluster, do not have a hierarchical relation in the Wikipedia structure, we analyze how often these relations occur in collection of user profiles. The next step is adding more concepts to the group according to a dendrogram that is produced by agglomerative clustering algorithm. We analyze and clarify the results of this process on the discovered groups.

To analyze the effect of the hierarchical assigned concepts during the classification process on the grouping process, we run the grouping process using a data set with and without using parent and root assignments of concepts to user profiles. The analysis consists of comparing the set of discovered groups using these different data sets. Based on the differences in the discovery of semantic related groups, we conclude how the use of a hierarchical concept structure affects the grouping process.

We also analyze the effect of errors by the classifier on the grouping process. By comparing, the discovered groups using the perfect classified set of profiles with concepts ('ground truth') and the results set of our classifier. Using this approach, we focus on if the errors in the original data set could results in similar groups, or that it also results in incorrect groups.

9.3 Concept clustering approach

9.3.1 Association score

For our grouping process we apply a hierarchical clustering algorithm on a set with information about co-occurrences of concepts in user profiles. To create a set with this information, the co-occurrences of concepts is measured by an association score. Well-known association scores are PMI, Chi-squared Test and Dice Coefficient. We use the most widely used measure: PMI (Pointwise Mutual Information) [12]. PMI is defined as follows:

$$PMI(x, y) = \log \frac{p(x, y)}{p(x) \cdot p(y)}$$

The probability that a profile in the collection has concept x assigned to is it, is measured by $p(x)$. $p(x, y)$ measures the probability that the concept x and y occur together in the same profile. The mutual information is high when the concepts x and y occur together more often. Calculating the PMI for all combination of concepts results in an adjacency matrix with the association scores, which is the input for the clustering algorithm.

9.3.2 Concept pruning

Not all concepts that have a relation to a user profile are interesting for discovering groups. When a concept occurs only in a very few times in a profile, it is not relevant to use it as a definition of a group. Figure 9.2 shows the distribution of occurrences of concepts in user profiles for concepts that occur in 20 or less profiles. 49 concepts occur only in one of the 1503 profiles in the collection. We consider concepts that occur in less than 1% of the profiles as not useful, which means that we filter concepts that occur in less than 16 profiles (total

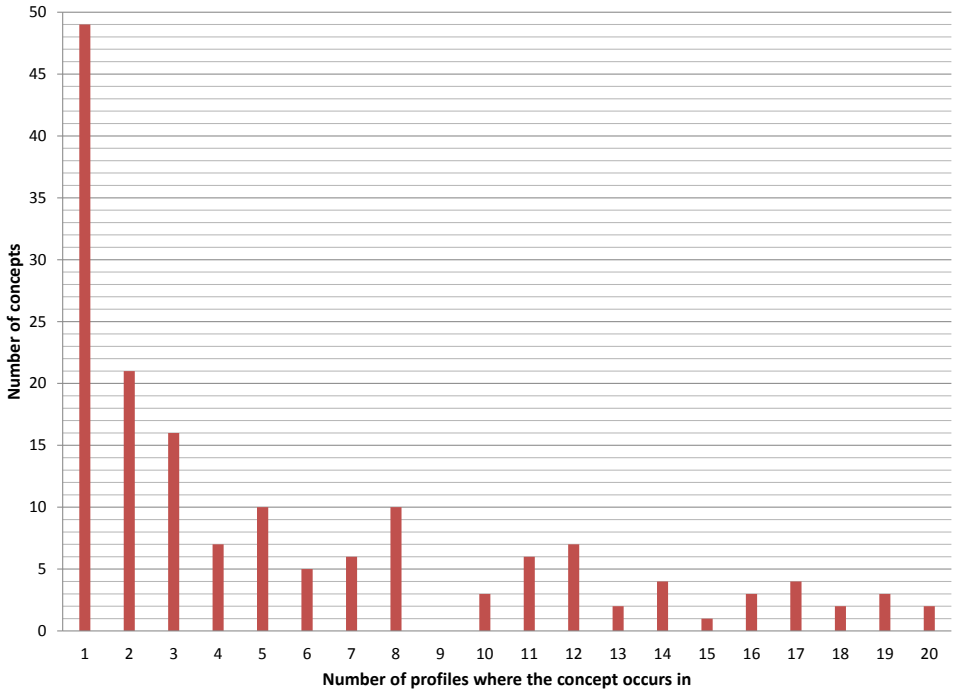

Figure 9.2: Distribution of concept occurrences in user profiles

148). During the assignment approach, the root concepts occur very frequently in profiles. This makes them not useful for discovering groups. Leaving these concepts out results in a total number of 58 concepts that are used for discovering groups. Appendix E shows the parts of the Wikipedia category graphs that are left after pruning the concept structure.

9.3.3 Clustering algorithm

We use an agglomerative hierarchical clustering algorithm. This means that based on the initial data, every concept is considered as a cluster. By iterations of the clustering algorithm, similar clusters are merged together [22]. When a cluster is merged, the new point of the cluster is calculated as the mean values between all the elements in the cluster. This new value is calculated using the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) function, and is also called average linkage clustering. Algorithm 9.1 describes this process.

When applying the algorithm on the data set until all elements are merged into one cluster, the result in a rooted tree called a dendrogram. Appendix D shows the dendrogram after clustering on the set PMI association matrix based on all valid assignments of concepts to user profiles. We base our initial selection of groups on these clustering results and compare them with results of using other sets, while applying the same clustering approached discussed in this section.

Algorithm 9.1 Agglomerative hierarchical clustering algorithm using UPGMA [19, 42]

Input: Matrix M with the initial PMI values of combinations of clusters x and y. All concepts are initialized as individual clusters.
Output: Output tree T (the dendrogram).

Initialization:

1. Initialize n clusters, where n is equal to the number of concepts.
2. Set the size of each cluster to 1. $n_x = 1$.
3. In the output tree T (the dendrogram), assign a leaf for each concept.

Iteration:

1. Find the x and y that have the largest similarity according to value in the matrix M.
2. Create a new cluster xy, which has $n_{xy} = n_x + n_y$ concepts.
3. Connect x and y on the tree T to a new node (a group), which corresponds to the new cluster xy.
4. Compute the new values for the matrix M, using the UPGMA merging criterion, for the new cluster xy to every other cluster (represented as z), except x and y, using:

$$M_{xy,z} = \frac{n_x}{n_x + n_y} \cdot M_{x,z} + \frac{n_x}{n_x + n_y} \cdot M_{y,z}$$

5. Delete the columns and rows in M that correspond to clusters x and y, and add a column and row for the new calculated cluster values (step 4) for the cluster xy.
6. Return to 1., until there is only one cluster left that contains all concepts.

9.4 Selection of groups

According to the research method described in Section 9.2 we select groups (clusters) when there are two concepts in a cluster that do not have a hierarchical relation in the Wikipedia structure. The Wikipedia structure consists of 10 sub-graphs and 10 standalone concepts (Appendix E). Table 9.1 lists the concepts that occur in the same cluster (Figure D.1). For three clusters, we list concepts that do occur in the same Wikipedia sub-graph and have a hierarchical relation, to show the effect of the assignment strategy on the clustering results. These groups are marked with 'Yes' in the 'Parent-child?' column. The other groups consist of concepts that occur in different sub-graphs of the Wikipedia structure. We refer to groups using the cluster id's in Appendix D and the concepts or groups of concepts x and y. The number of profiles where the concepts occur in, is defined by $n(x)$ and $n(y)$. The number of times x and y occur together in a profile is measured by $n(x, y)$.

When we use cluster id's of (Appendix D) for describing x or y, then we count the profiles that contain at least one concept of both branches in the profile. For example, $n(Symptoms(C73))$ counts the total number of profiles that contain at least one concept of the C73 cluster ('Symptoms', 'Pain', 'Nociception', 'Headaches'). The value of $n(Symptoms(C73), Sleep(C70))$ counts the profiles that contain at least one concept from the 'Symptoms (C73)' branch and at least one concept from the 'Sleep (C70)' branch. Exceptions are marked and explained with footnotes, remarkable groups discussed in the next section are bold.

Cluster ID	x	n(x)	y	n(y)	Parent -child?	n(x,y)	n(x,y)/ n(x) %
C58	Breast cancer	33	Breast diseases	34	Yes (is-a)	32	97.0
C70	Beds	210	Sleep	390	Yes	176	83.8
C65	HIV/AIDS-Sexually transmitted diseases (C61)	18	Immune system disorders	19	No	12	66.7
C71	C65	25	Syndromes	20	No	11	44.0
C83	**Laundry**	**115**	**Cleaning**	**154**	**No**	**37**	**32.2**
C80	**Symptoms (C73)**	**324**	**Sleep (C70)**	**424**	**No**	**155**	**47.8**
C80[1]	Symptoms (C73)	34	Sleep (C70)	176	No	12	35.3
C92	Vegetarianism	17	Garlic	24	No	3	17.6
C98	Analgesics	24	Bedding	45	No	4	16.7
C81	Diets	71	Nutrition	338	Yes	59	83.1
C91[2]	Sleep-Symptoms (C80)	155	Nutrition-Diets (C81)	350	No	57	36.8
C75	Dietetics	17	Obesity	22	No	8	47.1
C99	**Diabetes**	**16**	**C75-C82**	**45**	**No**	**3**	**18.8**

Table 9.1: Groups based on interesting clusters using the 'ground truth' set

[1]The p-values for this group assume that a profile should contain all Symptoms concepts (Symptoms, Pain, Nociception, Headaches) and all Sleep concepts (Sleep and Beds)

[2]The p-values for this group assume that a profile should contain at least one concept from the Sleep branch (C70), at least one concept from the Symptoms branch (C73) and at least one concept from the Nutrition branch (C81)

9.5 Analyzing the groups and profiles

The results in Table 9.1 are based on clustering data of the manually classified data set of profiles with concepts. We call this set the 'ground truth'. In the next subsections, we discuss the grouping results and the effect of using different input sets for the grouping process compared to using the 'ground truth' set.

9.5.1 Groups based on the ground truth set

Groups with parent and child concepts

As expected, concepts that have a hierarchical relation in the Wikipedia structure are grouped together at the lower levels of the dendrogram. The strongest relation between concepts is between 'Breast cancer' and 'Breast diseases' (C58). This can be explained by the fact that the relation between these concepts is an 'is-a' relationship. So when a person is writing about the lower level concept, it is also about the parent concept, however not the other way around. In this collection, the 'Breast cancer' concept occur probably relatively often, because the message were obtained during the 'Breast cancer awareness month'.

The strongest relation between concepts that do have a parent-child relation and not an 'is-a' relation is the group of 'Beds' and 'Sleep' (C70). Message with words related to the concept 'Beds' are often related to the concept 'Sleep'. However, the other way around there are more profiles with a relation to the concept 'Sleep', because when people do not explicitly refer to 'Bed' there is no relation with this concept. Noticeable is that according to the Wikipedia structure, the concept 'Dreaming' has a relation to the concept 'Sleep', while these concepts do not occur in the same group at the lower levels of the dendrogram. This is probably caused by the fact that the concept 'Dreaming' is used in a figurative sense: a daydream. In that case, the concepts often has not a relation to 'Sleep'.

The grouping results show many groups consisting of concepts that have a relation in the Wikipedia structure. These types of groups are not remarkable when looking for unexpected relations between concepts. However, when looking for a semantic relation between profiles, the hierarchically related concepts are merged together and the profiles in these groups have a relation at a more abstract level. A good example is the 'Symptoms' group (C73) that merges the concepts 'Symptoms', 'Pain', 'Nociception' and 'Headaches'. Messages in profiles could refer to different types of symptoms, while by grouping on concepts and related terms like 'pain' and 'headache', these profiles belong to the same group.

Groups with semantically related concepts

The groups C71, C75 and C79, consist of concepts that do not have a relation according to the Wikipedia structure. However, based on the semantic meaning of the concepts in these groups a relation could be expected. For example, the concept 'HIV/AIDS' occurs in the same groups as 'Immune system disorders' and 'Syndromes' (group C71), while the Wikipedia category 'HIV/AIDS' does not occur in the other categories. The concepts do often occur together in the same profile. This is caused by the terms related to the categories. The term 'AIDS' is related to all these categories, so when a message in a profile contains

this term, these tree concepts will be assigned to the profile.

The same phenomenon occurs with the concepts 'Dietetics' and 'Obesity' (C75), which both have the related term 'weight loss'. In addition, 'Breast cancer' and 'Aging-associated diseases' (C79) share a related term: 'cancer'.

In these cases, the grouping results show some information on the Wikipedia category structure could be improved. For example, the 'HIV/AIDS' category could be added as sub-category to 'Syndromes'. These similar related terms to concepts prevent discovering groups of concepts that have a different semantic meaning. However, just like the groups of parent and child concepts discussed in the previous section, the profiles would be grouped according to similar semantic content.

Groups with concepts from different Wikipedia branches

There are groups of concepts that are merged together, while there is no clear relation in meaning or in the Wikipedia structure between the concepts. For example, the relation between concepts related to 'Sleep' and concepts related to 'Symptoms' (C80). In almost 50 % of the profiles that have a relation to a 'Symptoms' concept, also have a relation to a 'Sleep' concept. An explanation for this discovered group, could be that these concepts occur relatively frequently in profiles, which increases that they occur together. Topics related to these concepts are often mentioned in the short messages.

The results show also a group of 'Laundry' and 'Cleaning' (C83). These concepts could be considered as semantic related, such as the concepts discussed in the previous paragraph. However, these concepts do not have related terms in common that are used during classification process. For these concepts, the explanation that these concepts occur frequently in profiles still holds. Interesting is the fact that the relation of 'Cleaning' with 'Laundry' is stronger that the relation with 'Cleaning products', while these concepts have a parent-child relation.

Another interesting group is the group of 'Diabetes' in relation to concepts related to 'Dietetics' and 'Obesity' (C99). Compared to the other remarkable relations, this relation is not very strong. However, the relation is stronger than relations to other concepts that are not grouped together with 'Dietetics' at this level of the dendrogram. We are able to view the profiles with the messages in a group. These profiles show that the relation between these concepts is based on profiles with a focus on a healthy life style. The users mentioned these concepts in different messages.

There are also standalone concepts according to the pruned Wikipedia structure that are grouped together by the clustering algorithm. Examples are 'Vegetarianism' with 'Garlic' (C99) and 'Analgesics' (painkillers) with 'Bedding' (C98). Although these relations are weak, it is remarkable that the C98 group consists of a concept from the Wikipedia 'Symptoms' structure and the 'Sleep' structure, just like group C80.

Due to the used set of concepts, which is limited to the health domain, the discovered groups are probably less remarkable than when concepts from different domains were grouped together. In addition, very common concepts like 'Laundry' and 'Cleaning' are maybe less interesting than a strong relation between concepts that occur less frequent in the collection.

9.5.2 Effect of errors in classification results

In this section, we analyze how errors in the classification results affect the grouping results. We use a data set that is based on the evaluation folds of the improved classifier. The set has the same precision (69.6 %), recall (66.5 %) and F-measure (0.68) as set 17 in Table 8.1, with almost the same number of true and false positives and negatives.

Besides the effect of false positives and negative in the grouping process, these errors also affect the number of profiles that do actually occur in the discovered groups. However, in this section we only focus on if the groups presented in Section 9.4 still remain in the results when the data set contains errors.

The results of the clustering algorithm (Figure D.2), show (groups of) concepts that do not occur in the results of the 'ground truth' set. For example, 'Epidemics', 'Pandemics', 'Cutaneous conditions' and 'Drinking water'. The other way around, there are also concepts that occur in the 'ground truth' set and not in the set with classification errors. Examples are 'HIV/AIDS' and 'Diabetes'. The differences in concepts affect the grouping process negatively, because it results in other (incorrect) groups.

A remarkable grouping error is the group of 'Nutrition' and 'Exercise' (C69). In the results of the 'ground truth' set these concepts occur in different groups.

The most important relations discovered using the 'ground truth' set, remain. The relation between 'Laundry' and 'Cleaning' and the relation between 'Sleep' concepts and 'Symptoms' concepts exist in the grouping results. However, the last one is mixed up with the previously mentioned incorrect group C69.

These results show that the quality of the improved classifier is still not sufficient for discovering the correct groups, according to the 'ground truth' set. While these groups are not the same as the 'ground truth' groups, this does not mean that they are not interesting or not useful. There is a relation between these concepts, considering that these concepts or their related terms occur frequently together in profiles with short messages. These groups based on relations between the concepts could be correct and useful, while the profiles do not belong the group.

9.5.3 Effect of the hierarchy

The effect of assigning abstract or higher level concepts to profiles based on matching more specific concepts, is analyzed by using two different data sets. We review the grouping results of data sets containing assignments of concepts to profiles on the two lowest levels (Figure D.3) and the base level only (Figure D.4).

The groups in Figure D.3 still contain semantically related concepts, such as the 'Symptoms' and the 'Exercise' group. When only the base concepts are used for the grouping, the discovered groups are very different. The 'Cleaning' and 'Laundry' group still exists, and the 'Exercise' related concepts, except 'Walking' occur in the same group. However, the concepts 'Symptoms', 'Pain' and 'Headaches' are spread over different groups. In addition, the 'Sleep' related concepts are not grouped together.

We cannot conclude that these groups are less interesting. The combinations of concepts differ from the results that use the hierarchical information of the Wikipedia structure. However, we can conclude that there is less semantic grouping of concepts (and profiles). Profiles with messages related to different type of symptoms will become in different groups, while when using the hierarchical information, these profiles are grouped together.

9.6 Discussion of the results

The grouping results show that there are is a relation between the concepts 'Laundry' and 'Cleaning' and between 'Symptoms' and 'Sleep' related concepts. These results do not tell something about how usable the discovered groups are in the 'real world'. We explained most reasons why concepts where grouped together. Because the concepts are limited to the health domain, concepts have already a certain relation, which has an effect that discovered relations are less special.

Based on the results we discovered that there are also semantic relations between concepts, which are not correctly reflected in the Wikipedia category structure. While the discovery of these types of relations and groups is not the main goal of this research, this strategy could be used to fix errors in the Wikipedia category structure or add new relations to the category structure.

The size of the used collection of profiles is limited, which results in discovered groups and relations occur less frequently in profiles, such as 'Obesity', 'Diabetes' and 'Dietetics'. Because these concepts do not occur very often in profile collection, this does not prove that these groups are correct. In addition, the approach of using PMI as association metric could affects the results in a way that two concepts that occur a few times and occur together (by coincidence), they get a high score, which in the end results in a group. Pruning the concepts reduced these types of effects on the results, however they could still remain.

Considering these characteristics of our collection and the used approach, future investigations with a more varied concepts structure and a larger profile collection is required. More research on the validity of the discovered groups is required to evaluate our clustering approach to discover groups. In addition, other association metrics could be used, such as a metric similar to the Normalized Google Distance used by Crabtree et al. [7].

Chapter 10

Conclusions & future work

People use online social networking sites to express themselves. They publish (personal) information in short text messages in their user profile, that could be very useful for marketing and advertising. We researched the possibilities of discovering groups using the short text messages from online social network profiles.

To accomplish the research we grouped user profiles from online social networking sites based on similarity of concepts in the profiles. We obtained a hierarchical structure of concepts from Wikipedia, to be able to discover conceptual relations between user profiles on more abstract levels. Together with a gathered collection of profiles with short messages from Twitter and a manual evaluated collection of the Wikipedia concepts related to the Twitter user profiles, we managed to discover groups.

In the first three sections of this final chapter we discuss, evaluate and answer the research questions related to the results of using the hierarchical concepts from Wikipedia, assigning these concepts to user profiles based on the short text messages and discovering groups based on the occurrences of concepts in profiles. In addition, we give some recommendations for future work that is related to these specific different parts of our text mining process. In the last section we draw conclusions and give suggestions for future work on the whole process of discovering groups in social networking sites.

10.1 Hierarchical structures of concepts

- *Which existing types of hierarchical structures of concepts are according to literature useful for finding relevant groups, taking semantic relation into account?*

To discover relations between user profiles on more abstract conceptual levels, we used a hierarchical structure of concepts. Several hierarchical structures exist. Due to the broad number of covered concepts in many domains and the availability of the structure that contains named entities, we used a hierarchical structure that is based on the Wikipedia category system.

In this research, the collection of concepts was limited to concepts in the health domain. Due to this selection, there is a limited variation in semantic meaning of the concepts. This affected the results of discovering special groups of related concepts and profiles, which is discussed in Section 10.3.

In addition, errors in the structure and the ambiguous meaning of terms in the concept structure affect the classification and grouping process. These problems are also evaluated in the following two sections.

Future work

Future researches that deal with concept structures from Wikipedia and classification, could focus more on a better automatic selection of relevant concepts and reducing the number of concepts that cause many classification errors.

10.2 Classification of concepts with user profiles

- *Which classification strategy is suitable for automatic assigning concepts from a hierarchical structure to user profiles based on short text messages?*
- *How to evaluate the quality of the assignment of concepts to user profiles?*
- *What is the quality of the assignment of concepts to user profiles?*

We used a classification strategy of matching terms in short text messages with terms in Wikipedia concepts (category and page titles) to automatically assign concepts to user profiles. This naive classification approach is relatively fast and works without manual training. However, the correctness of the results, a precision of 37.4 %, is not sufficient enough for the grouping of concepts related to user profiles.

An improved classification approach, that uses additional features for classifying the relation between concepts and profiles helps to gain better classification results. Especially features related to the concepts in the collection and the relation between the concept and the user profile, show improvements of the classification results, while features that tell something about the user profile only do not affect the results positively. The results are more precise, however this also affects the completeness (or recall) negatively. Considered, precision and recall both as important, by measuring the F-score, the improved classifier is 14 % better that the naive classifier with an F-score of 0.68.

Future work

The approach of matching terms of concepts with terms in messages does not deal with classical classification problems like ambiguous words, misspellings and multiple forms of words. Incorporating additional resources, such as WordNet for the disambiguation of words and (parts of the) content of Wikipedia articles might help to increase the precision of the classifier. To limit the number of matches, in order to be able evaluate the classifications manually, we did not used a stemmer and considered only matches of the full terms. In future research stemmers could be used to increase the recall. However, this would probably also result in a lower precision.

To know if the used approach of matching terms is better than other term matching approaches, such as matching short messages with Wikipedia articles, more research is required. Besides introducing and using different techniques for the classification, another important part of future research could be considering different definitions of a relation between concepts and user profiles. In this research, a relation between a concept is based on if a user refers to the concepts in relation to himself or if he refers to the concept more than once. For marketing, it would also be interesting to know if there is a positive, negative or neutral relation to the concept (the polarity). This field of text mining is called sentiment analysis and research by Wilson et al. [50] shows approaches to extract the polarity of a phrase. This would make it possible to identify for example if a user likes soccer, or that he or she hates it.

10.3 Clustering to discover groups

- *Which clustering strategy is suitable for discovering groups based on hierarchical concepts related to user profiles?*
- *What is the effect of using the hierarchical concepts on the discovered groups?*

Our clustering strategy to discover groups is based on the co-occurrences of (hierarchical) concepts in user profiles. With this strategy, profiles could occur in multiple groups and the discovered groups could be interesting if there are relations between concepts that do not have semantic or other easy explainable relation. Because our concept collection is limited to the health domain, the (number of) discovered groups are might be less remarkable. The 'Sleep' and 'Symptoms' group and the 'Laundry' and 'Cleaning' group are the most remarkable groups of concepts we found. These relations occur because these concepts occur frequently in profiles, thus also more frequently together.

We cannot conclude that these groups are useful. Analyzing the messages in profiles that belong to these groups showed that users refer to these concepts in their profile. However, that does not mean that this information applicable in for example a marketing process.

In addition, we defined a membership of a group as a profile that contains at least one concept in each Wikipedia sub-graph of concepts. Variations on this definition result in different numbers of profiles that belong to a group. Future research is required to discover how valid and useful the groups are, what the best way is to select the groups from the results, and to discover special relations by using a Wikipedia structure with concepts from multiple domains.

As the results show, the selection of a concept structure guides the grouping results in to a certain direction. Our approach supports fast selection of concepts from different domains without knowing all related domain specific terms and other concepts. This makes it possible to discover relations between profiles and concepts focusing on specific domains where someone is interested in.

The effect of using the hierarchical structure of concepts when assigning also higher-level concepts to profiles during the classification process, is the discovery of groups with semantic related profiles. When looking at the similarity between profiles on term level, a profile with the term 'pain' and a profile with the term 'headache' are not similar. However, with our approach, we look on a

higher conceptual level to the profiles, and in that case these profiles do have a relation: terms related to the concept 'Symptoms'. Clustering approaches without incorporating the hierarchical structure will not discover these type of semantic relations between profiles.

A side effect of our approach is the discovery of missing relations between concepts in the Wikipedia category system. There are concepts that are assigned to profiles that do not have a relation in the Wikipedia category system, but share similar terms. Based on matching terms that occur frequently in short messages, these concepts are assigned to the profiles. After clustering these concepts are grouped together. The hierarchical relations between these concepts (in this research for example 'HIV/AIDS' and 'Syndromes'), is not reflected in the relation between Wikipedia categories, while it would be a correct mapping. A method of using (user related) documents, in our case short text messages from social networking sites, could help to discover concepts that occur in different branches of the Wikipedia structure, while they often occur together in documents. This information is useful to improve the Wikipedia category structure.

10.4 Discovering groups in social networks based on short messages and the future

- *Can we automatically categorize and group user profiles from online social networking sites based on the semantic similarity of short text messages using an existing hierarchical structure of concepts?*

Overall we showed that it is possible to extract groups from social networking sites based on short text messages. Short messages contain only a few terms, which makes it hard to discover semantic similarity between profiles based on messages, because text mining techniques often rely on similary between other (semantically) related terms in the context of the document. Introducing a hierarchical structure of concepts by assigning concepts from different levels of a hierarchical structure of concepts to user profiles, helps to discover semantic similarity between user profiles based on short messages. In the end, this results in the discovery of groups based on higher-level concepts and the discovery of relations between these concepts.

Future research and applications

Because the automatic approach of matching terms from concepts with terms in short messages is not precise. An additional classification method using Support Vector Machines is used to improve the assignment of concepts to user profiles. More research and improvement is required on this part of the process. In addition, research to validation criteria and the validation of the discovered groups and relations is required.

In order to be able to use a text mining solution to discover groups in social networking sites, there are several issues to think about. For example, the crawling process and selection of user profiles. Crawling the messages at one moment in time results in a static data set, while users continue publishing new messages. The date and time a message was published could also affect on how useful the information is. Old messages might not reflect the user's current interests. A

more streaming based classification method could take into account when there are new messages in a profile. A search-based approach could be used to select relevant profiles in the first place, for example by querying for relevant terms in profiles on the search engine of the social networking site, instead of crawling a collection profiles randomly. Another important issue, when handling personal data, is privacy. Users might have a public profile, which does not mean that it is allowed to approach users for marketing purposes. However, people maybe are willing to 'trade' their personal information for special offers from a company.

In this research, we explored the possibilities of discovering groups in online social networks based on short text messages. In this part of the text mining field there are many paths that can be taken to achieve a solution. We explored only a few of these paths, which gives a view on the techniques and strategies that will work and not work. Results are sometimes different than expected and we discovered solutions for problems that we were not looking for. There are still many paths that can be taken in order to be able to apply these techniques in practice and to take advantage of the opportunities texts in online social networking sites give.

Bibliography

[1] Antonellis, I. and Gallopoulos, E. Exploring term-document matrices from matrix models in text mining. In *Proceedings of the SIAM Text Mining Workshop 2006, 6th SIAM SDM Conference*. Maryland (2006).

[2] Apache Software Foundation. Similarity (Lucene 2.4.0 API). http://lucene.apache.org/java/2_4_0/api/org/apache/lucene/search/Similarity.html (2008).

[3] Artstein, R. and Poesio, M. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596 (2008).

[4] Bloehdorn, S. and Hotho, A. Boosting for text classification with semantic features. *Lecture Notes in Computer Science*, 3932:149–166 (2006).

[5] Claypool, M., Le, P., Waseda, M., and Brown, W. Implicit interest indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI '01)*, pages 33–40. USA (2001).

[6] Conrad, J., Al-Kofahi, K., Zhao, Y., and Karypis, G. Effective document clustering for large heterogeneous law firm collections. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 177–187. Bologna, Italy (2005).

[7] Crabtree, D., Andreae, P., and Gao, X. Query directed web page clustering. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 202–210 (2006).

[8] D'Alessio, S., Murray, K., Schiaffino, R., and Kershenbaum, A. The effect of using hierarchical classifiers in text categorization. In *Proceeding of RIAO-00, 6th International Conference Recherche d'Information Assistee par Ordinateur*, pages 302–313 (2000).

[9] Dhillon, I.S., Mallela, S., and Kumar, R. A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research*, 3:1265–1287 (2003).

[10] Dumais, S. and Chen, H. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263 (2000).

[11] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. Inductive learning algorithms and representations for text categorization. In *Proceedings of*

the seventh international conference on Information and knowledge management, pages 148–155 (1998).

[12] El Sayed, A. *Contributions in Knowledge Discovery from Textual Data*. Ph.D. thesis, Université Lumière Lyon (2008).

[13] Errecalde, M., Ingaramo, D., and Rosso, P. Proximity estimation and hardness of short-text corpora. In *DEXA '08: Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*, pages 15–19. IEEE Computer Society (2008).

[14] Feldman, R. and Sanger, J. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press (2007).

[15] Feng, A. and Allan, J. Hierarchical topic detection in TDT. 2004. Technical report, Technical Report.-Center for Intelligent Information Retrieval. University of Massachusetts, Amherst (2005).

[16] Forman, G. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305 (2003).

[17] Gabrilovich, E. and Markovitch, S. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the National Conference On Artificial Intelligence*, volume 2, pages 1301–1306. Boston, Massachusetts (2006).

[18] Gaussier, E., Goutte, C., Popat, K., and Chen, F. A hierarchical model for clustering and categorising documents. *Lecture Notes in Computer Science*, 2291:121–125 (2002).

[19] Gronau, I. and Moran, S. Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210 (2007).

[20] Hall, M.A. *Correlation-based feature selection for machine learning*. Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1999).

[21] Hearst, M.A. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10 (1999).

[22] Huang, J.Z., Ng, M., and Jing, L. Text clustering: Algorithms, semantics and systems (2006).

[23] Jacob, E.K. Classification and categorization: A difference that makes a difference. *Library Trends*, 52(3):515–540 (2004).

[24] Java, A., Song, X., Finin, T., and Tseng, B. Why we Twitter: Understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, San Jose, California (2007).

[25] Jing, L., Zhou, L., Ng, M.K., and Huang, J.Z. Ontology-based distance measure for text clustering. In *Proceedings of the Fourth Workshop on Text Mining; Sixth SIAM International Conference on Data Mining*. Bethesda, Maryland (2006).

[26] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142. Springer (1998).

[27] Kim, H. and Chan, P. Learning implicit user interest hierarchy for context in personalization. *Applied Intelligence*, 28(2):153–166 (2008).

[28] Krishnamurthy, B., Gill, P., and Arlitt, M. A few chirps about Twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24. ACM, Seattle, WA, USA (2008).

[29] Landis, J. and Koch, G. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174 (1977).

[30] Liu, J. and Birnbaum, L. Measuring semantic similarity between named entities by searching the web directory. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 461–465 (2007).

[31] Liu, Y. and Loh, H.T. A simple probability based term weighting scheme for automated text classification. *Lecture Notes in Computer Science*, 4570:33–43 (2007).

[32] Meij, E., Bron, M., Huurnink, B., Hollink, L., and de Rijke, M. Learning semantic query suggestions. *Lecture Notes in Computer Science*, 5823:424–440 (2009).

[33] Netscape Communications Corporation. ODP - Open Directory Project. http://www.dmoz.org/ (2010).

[34] Pedersen, T., Patwardhan, S., and Michelizzi, J. Wordnet: similarity-measuring the relatedness of concepts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025 (2004).

[35] Pew Research Center's Internet & American Life Project. Adults on social network sites, 2005-2009. http://www.pewinternet.org/Infographics/Growth-in-Adult-SNS-Use-20052009.aspx (2009).

[36] Ponti, G. and Tagarelli, A. Topic-Based hard clustering of documents using generative models. *Foundations of Intelligent Systems*, pages 231–240 (2009).

[37] Ponzetto, S.P. and Strube, M. Deriving a large scale taxonomy from Wikipedia. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 1440–1445. AAAI Press, Vancouver, British Columbia, Canada (2007).

[38] Ramanathan, K., Giraudi, J., and Gupta, A. Creating hierarchical user profiles using Wikipedia (2008).

[39] Rodrigues, M.M. and Sacks, L. A scalable hierarchical fuzzy clustering algorithm for text mining. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing* (2004).

[40] Salton, G., Wong, A., and Yang, C. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620 (1975).

[41] Schönhofen, P. Identifying document topics using the Wikipedia category network. *Web Intelligence and Agent Systems*, 7(2):195–207 (2009).

[42] Shamir, R., Inbar, Y., and Hartman, T. Algorithms in molecular biology. http://www.cs.tau.ac.il/{~}rshamir/algmb/00/scribe00/html/lec08/node21.html (2000).

[43] Sieg, A., Mobasher, B., and Burke, R. Inferring user's information context: Integrating user profiles and concept hierarchies. In *Proceedings of the 2004 Meeting of the International Federation of Classification Societies*. Chicago, USA (2004).

[44] Stavrianou, A., Andritsos, P., and Nicoloyannis, N. Overview and semantic issues of text mining. *ACM SIGMOD Record*, 36(3):23–34 (2007).

[45] Sun, A. and Lim, E.P. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, volume 528 (2001).

[46] Tagarelli, A. and Karypis, G. A segment-based approach to clustering Multi-Topic documents. In *Workshop on Text Mining, in conjunction with the 8th SIAM International Conference on Data Mining (SDM '08)*. Atlanta, Georgia, USA (2008).

[47] Wang, W., Meng, W., and Yu, C. Concept hierarchy based text database categorization in a metasearch engine environment. *Knowledge and Information Systems*, 4(2):132–150 (2002).

[48] Weiss, S., Apte, C., Damerau, D., Oles, F., Goetz, T., and Hampp, T. Maximizing Text-Mining performance. *IEEE Intelligent Systems*, 14(4):63–69 (1999).

[49] Wikimedia Foundation. Wiki category membership link records (categorylinks.sql) and base per-page data (page.sql) dumps of 29th september 2009. http://download.wikimedia.org/enwiki/20090929/ (2009).

[50] Wilson, T., Wiebe, J., and Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354 (2005).

[51] Witten, I. and Frank, E. Data mining: practical machine learning tools and techniques. In *Data mining: practical machine learning tools and techniques*, pages 149–151. Morgan Kaufmann, second edition edition (2005).

Appendix A

Judgment guidelines

Every judge read the following instructions before starting the evaluation. These instructions and guidelines were available in Dutch and in English. This appendix contains the English instructions.

Instructions

You are going to validate if categories are correctly related to a user profile based on short text messages in this profile. When you start the validation tool, you will see the categories on the left, and messages on the right. If a category is related to the messages, you can mark it as **valid**, by clicking the radio button in the valid column, near the name of the category. If it is not related, you mark it as **invalid**.

When you hover your mouse over the category, you will get some information. This information shows why the category was added to the profile. However, it is not enough information decide whether the classification is correct or not. A category is related to the messages are about a category. It is not required that the category name appears in the message. For example: 'I'm washing my hands' is related to the category *Hygiene*, and *Hygiene* belongs to *Health effectors*. So these are valid categories.

The next sections describe scenarios when it is probably not clear how to classify the profile. It give instructions how when mark it as valid or invalid.

General messages, negative messages or messages related to someone else

- 'My mother has the swine flu and broke her ankle.'
- 'When does the swine flu vaccination program start?'
- 'There is a vaccine to protect against swine flu.'
- 'I don't have the flu.'

If there is only a single message related to a category (e.g. *Influenza*), this is not an important interest of the user, so we mark it as **invalid**. When there are

more (than one) messages of this type related to the same category, we assume that it is an interest of the user, and we mark it as **valid**.

Ambiguous meaning

A category or word can be related to other categories in different contexts. For example, the word *alcohol* occurs in the following related Wikipedia structures:

- *Health effectors*⇒*Nutrition* (alcoholic drinks, which affect health)
- *Hygiene*⇒*Antiseptics* (desinfectant to clean things and kill bacteria)

Which categories are correct depends on the context of the messages.

Parent-child relations

Relations in the Wikipedia category graph to not imply 'is-a' relationships. For example:

- *Sleep medicine* is a parent category of *Sleep disorder* (but a medicine is not a disorder)
- *Attention-deficit hyperactivity disorder* (ADHD) contains the page *Diet,* but there is not always a relation between those two.
- *Bariatrics* contains the page *Obesity*, but a message with the obesity does not imply bariatrics (prevention and treatment of obesity).

The messages probably contain the words of the sub pages or sub categories, but if there is no relation to the parent category and the messages don't contain this information, the classification is **invalid**.

However, sometimes this relation is **valid**, for example:

If a profile, with 'I take a shower' gets the categories *Hygiene* and *Health effectors*. These categories are valid, because taking a shower is related to hygiene, which affects health.

Category outside health domain

Most of the categories are related to 'Health'. However, there are categories outside the health domain.

Knowledge sharing (*Knowledge*⇐*Memory*⇐... *Dyslexia*)
English spelling (*Language orthographies*⇐... *Dyslexia*)
Data unit (*Memory*⇐*Dyslexia*)

If there are messages related to the category (based on the other rules), mark it as **valid**. It does not matter that the category is outside the health domain.

Vague information

'Au, my foot hurts'⇒*Foot disease*

'I'm eating a lot of fast food'⇒*Eating disorder*

Based on this information, mark the classification as **invalid**, if it is not clear if the message is about (in this case) a disease or disorder. If other messages make clear that it is a disease or disorder, mark it as valid. In this case *Pain* and *Health effectors*, would be **valid** categories.

Category is too specific

For example:

- *AIDS Pandemic*, when the messages are only about AIDS.
- *Influenza vaccines*, when the messages are only about vaccines.

In these cases, the classification is **invalid**.

Mouse over text

The infomation in the box showed when you put you mouse over the category title is not enought to make conclusions if it is valid. The category could be valid, because of the text messages and not because of the highlighted words, parents or root.

Automatic hints

When you mark certain categories as invalid, parent or root categories that have only the selected category as child, get a yellow background color. In that case you know that you have mare a previous related category as 'invalid', but you still have to check whether the marked categories are valid or invalid.

Appendix B

Symbol and set notations

Table 7.1 describes the symbols that are used to describe the features in Table 7.2. Figure B.1 gives a schematic overview of the sets C, P and CM and the related symbols mentioned in Table 7.1.

In the following sections, we explain the notation style of the different set operations that is used in Table 7.2. The used set operations show on which characteristics of the different sets the features values are based. It is not a definition of the most optimal implementation strategy to calculate the feature values.

Number of elements in a set

We use the operator $|X|$ to count the number of distinct elements in the set X.

Selection of elements from a set

To define the features, we often use operations and selection on the CM set (relations between concepts and messages). This set contains tuples and we use two different types of selecting elements from set. The first is selecting the whole tuple: $(mc, mq, mm, mtype) \in CM$. The other is selecting specific elements from the tuple: $mq \in CM$, which means that only the 'query' part of the tuple is selected from the set. We use mc for the (matched) concept, mq for the query, mm for the message and $mtype$ for the assignment type. The m prefix is used to distinguish the letters from the c and $type$ in the function of the feature. Multiple selections are separated with a comma (,).

Building new sets

To select (sub) sets of the set of concepts assigned to user profiles based on term matches with messages (CM), we defined a set-builder notation. We use a structure of $\{selection : conditions\}$ to create a new set based on a selection of elements from an existing set (see previous paragraph) that match a given condition. The input variables from the feature functions, such as c, p and $type$ are often used in this condition. When multiple selections are used, the structure $final\ selection|selection, selection : conditions$ is used. The $final\ selection$ part defines the variable or variables that should be used in the final set.

Two examples:

- The definition of $mf(c, p)$ is $|\{(mc, mq, mm, mtype) \in CM : c = mc \wedge mm \in p\}|$. In words, this means: select every tuple from CM where the concept is c is assigned to a profile (mc) by a match with a message (mm) that occurs in profile p. Then count every distinct tuple in the result set.
- The definition of $cf(c)$ is $|\{p|(mc, mq, mm, mtype) \in CM, p \in P : c = mc \wedge mm \in p\}|$. In words, this means: select every profile from the profile collection P and every assignment tuple from the CM set. When the message (mm) is published in profile p and c is an assigned concept to the profile (based on message mm), store the profile p in the new set. Then count all the (distinct) profiles in the final set.

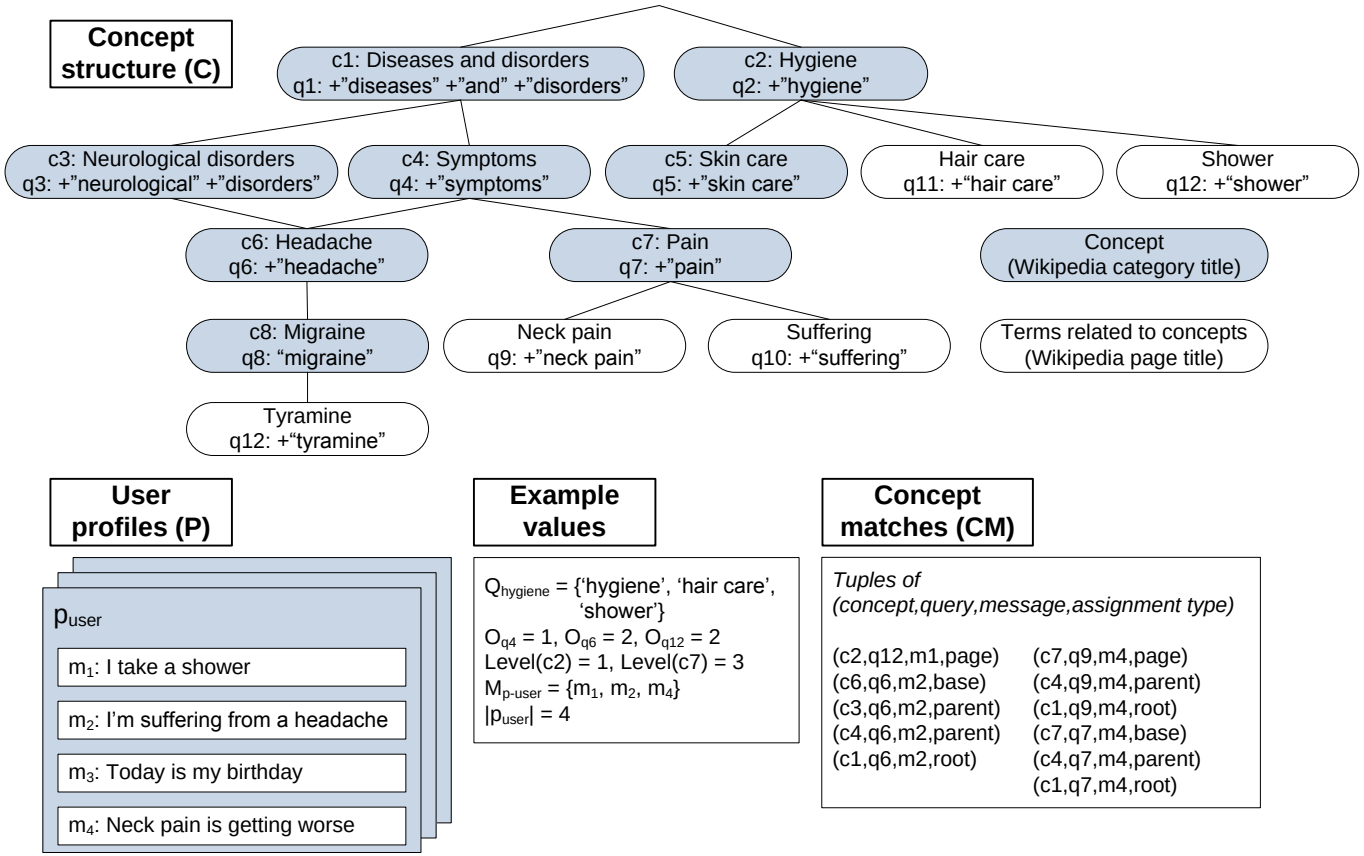

Figure B.1: Schematic overview of used sets and symbols

Appendix C

Classification results

The table on the next page shows the validation results of the improved classification models created with different feature sets. The values for the feature set 19 and 20 are corrected for errors made by the naive classifier. The superscripted number in the table have the following meaning:

[1] The features after 'and' in this row and the rows to feature set 15 are appended to the feature set in the previous row.
[2] 7,804 true negatives by the improved classifier and 1,948 true negatives by the naive classifier.
[3] 2,175 false negatives by the improved classifier and 169 false negatives by the naive classifier.
[4] 9,242 true negatives by the improved classifier and 692 true negatives by the naive classifier.
[5] 2,322 false negatives by the improved classifier and 241 false negatives by the naive classifier.

#	Feature set	# Features	# Correct	% Correct	# Incorrect	TP	FP	TN	FN	Precision	Recall	F-score
	Naive classifier (binary)	1	7,035	37.4	11,763	7,035	11,763	0	0	37.4	100.0	0.54
1	$n(c,p)$ (number of assignments)	1	13,032	69.3	5,465	3,978	2,708	9,054	3,057	59.5	56.5	0.58
2	$n_x(c,p)$ and$mfipf_x(c,p)$ (Table 7.4)	8	13,394	71.3	5,403	2,938	1,306	10,456	4,097	69.2	41.8	0.52
3	Base set (Table 7.5)	14	13,532	72.0	5,265	3,044	1,274	10,488	3,991	70.4	43.3	0.54
4	Base set and $MaximumScore(c,p)$	15	70.8	43.8	0.54	3,241	1,405	10,357	3,794	70.8	43.8	0.54
5	Base set and $TotalScore(c,p)$	15	69.8	46.0	0.55	3,082	1,274	10,488	3,953	69.8	46.0	0.55
6	Base set, $TotalScore(c,m)$ and $MaximumScore(c,m)$	16	68.9	46.9	0.56	3,298	1,489	10,272	3,737	68.9	46.9	0.56
7	and[1] $TermsInConcept(c)$	17	13,607	72.4	5,190	3,350	1,550	10,257	3,685	68.4	47.6	0.56
8	and $SelectedRatio(p)$, $ConceptRatio(c,p)$, $SelectedConceptFrequency(c,p)$	20	13,620	72.5	5,177	3,315	1,457	10,305	3,720	69.5	47.1	0.56
9	and $InverseQueryMatchingFrequency(c)$	21	14,087	74.9	4,710	4,265	1,940	9,822	2,770	68.7	60.6	0.64
10	and $Ambiguous(c,p)$	22	14,090	75.0	4,707	4,276	1,948	9,814	2,759	68.7	60.8	0.64
11	and $Duplicate(c,p)$	23	14,271	75.9	4,526	4,477	1,968	9,794	2,558	69.5	63.6	0.66
12	and $Ing(c,p)$ (Extended set)	24	14,302	76.1	4,495	4,505	1,965	9,797	2,530	69.6	64.0	0.67
13	and $Level(c)$	25	14,295	76.0	4,502	4,555	2,022	9,740	2,480	69.3	64.7	0.67
14	and $n_{i've}(c,p)$, $n_{me}(c,p)$, $n_{my}(c,p)$	28	14,348	76.3	4,449	4,557	1,971	9,791	2,478	69.8	64.8	0.67
15	and $n_x(c,p)$, where $x \in \{not, don't, doesn't, wouldn't, haven't, ain't\}$	34	14,328	76.2	4,469	4,526	1,960	9,802	2,509	69.8	64.3	0.67
16	Extended set and $AM(p)$, $TotalTerms(p)$	26	14,275	75.9	4,522	4,642	2,129	9,633	2,393	68.6	66.0	0.67
17	Correlation-based feature set (Table 7.3)	30	14,399	76.6	4,398	4,677	2,040	9,722	2,358	69.6	66.5	0.68
18	Correlation-based and common (domain unrelated) terms	90	14,294	76.0	4,503	4,374	1,842	9,920	2,661	70.4	62.2	0.66
19	Correlation-based without ambiguous page assignments	30	14,443	76.8	4,354	4,691	2,010	9,752[2]	2,344[3]	70.0	66.7	0.68
20	Correlation-based without root assignments	30	14,406	76.6	4,391	4,472	1,828	9,934[4]	2,563[5]	71.0	63.9	0.67

Appendix D

Clustering results

The following pages contain the output trees (dendrograms) of the agglomerative hierarchical clustering process of concepts related to profiles. These results are used in the analysis in Chapter 9.

The concepts in the dendrograms are concepts that occur in more than 1 % of profiles. The other concepts are removed, see Subsection 9.3.2.

The dendrograms are created based on the following data sets and occur in the following order:

- Concept occurrences in profiles in the 'ground truth' data set.
- Concept occurrences in profiles in the data set by the improved classifier (2050 false positives and 2338 false negatives).
- Concept occurrences in profiles in the 'ground truth' data set with page and base assignments.
- Concept occurrences in profiles in the 'ground truth' data set with base assignments.

Appendix D. Clustering results

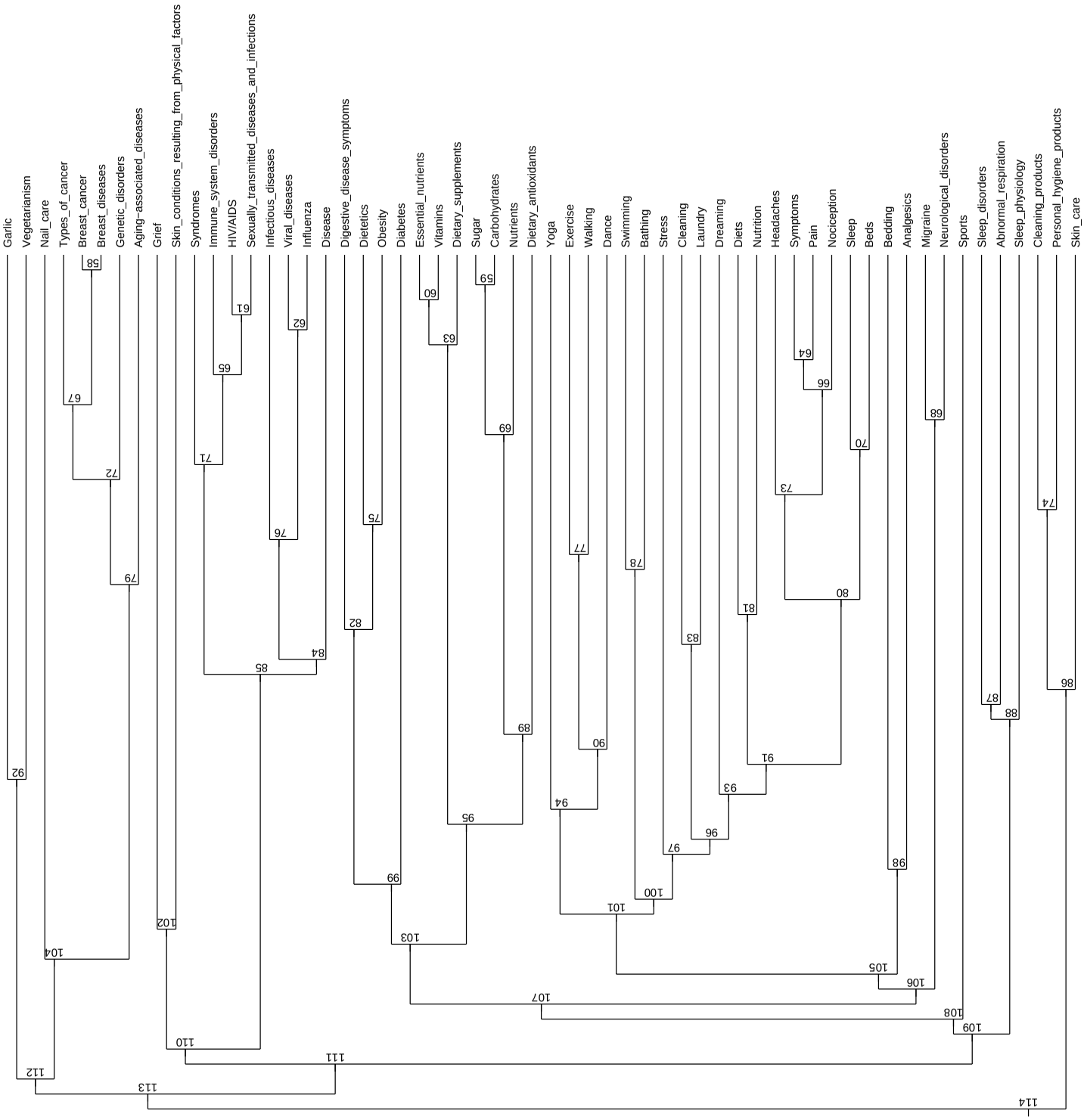

Figure D.1: Clustering results of the 'ground truth' set

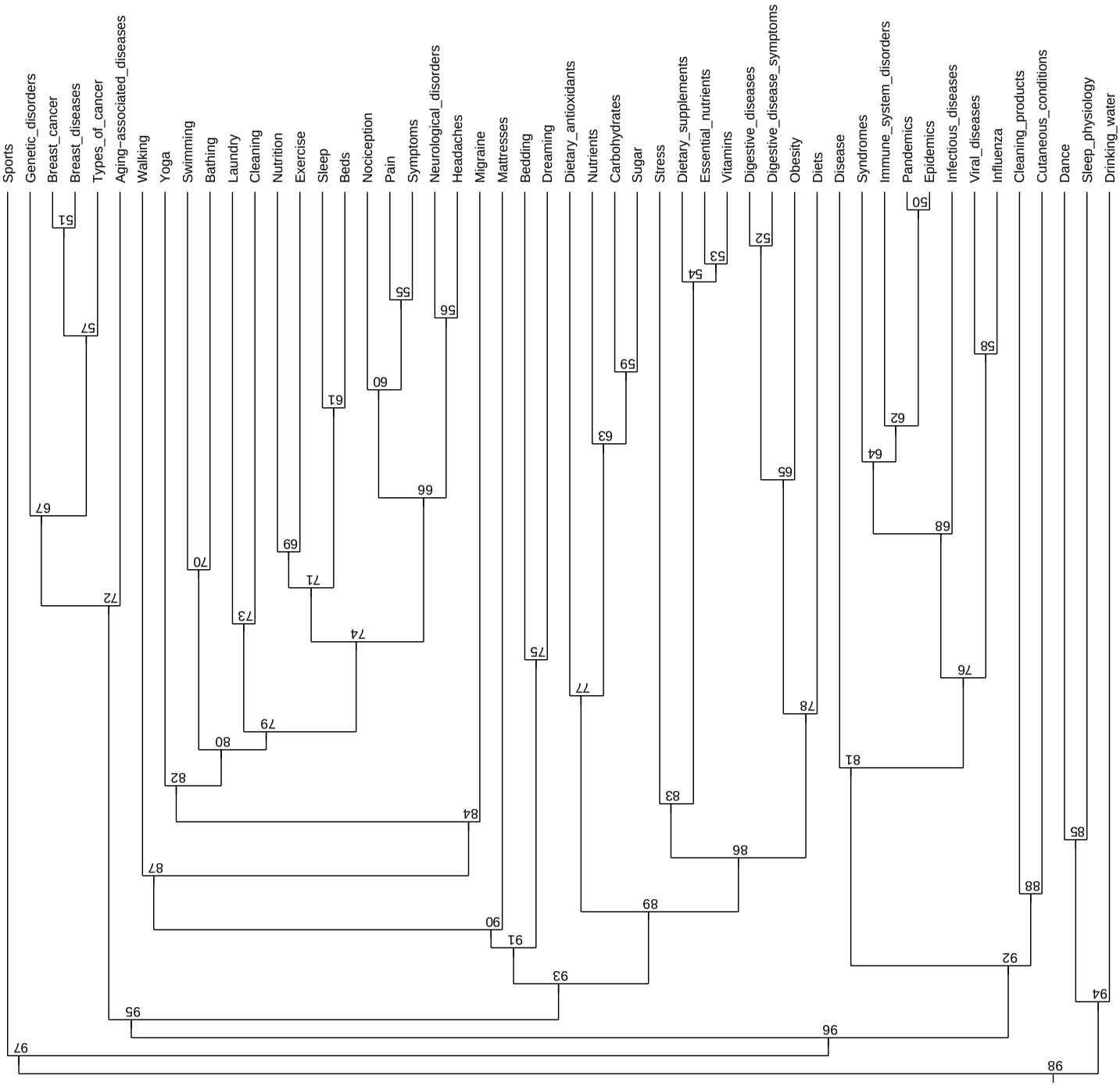

Figure D.2: Clustering results of the improved classifier set

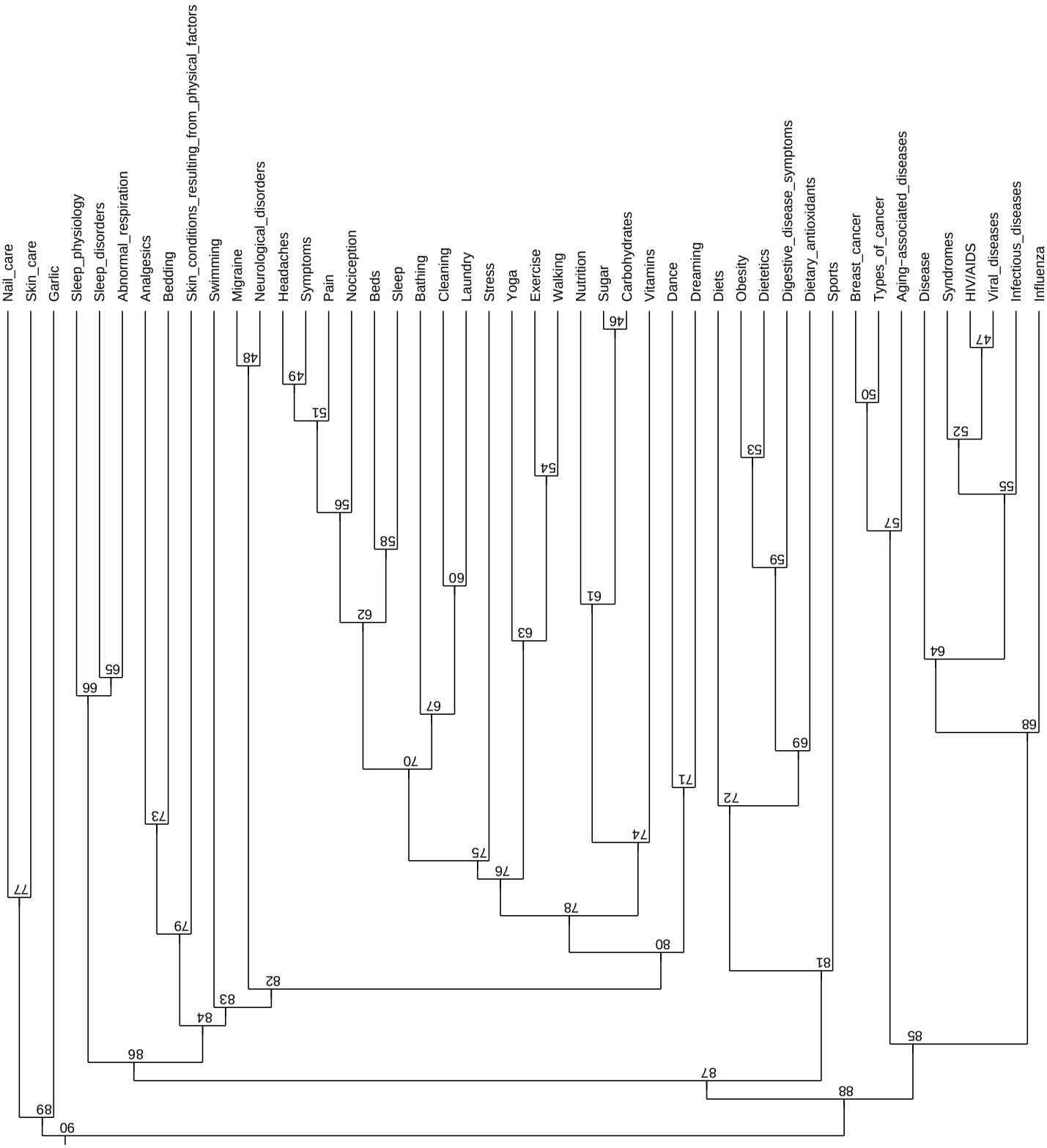

Figure D.3: Clustering results using page and base assignments

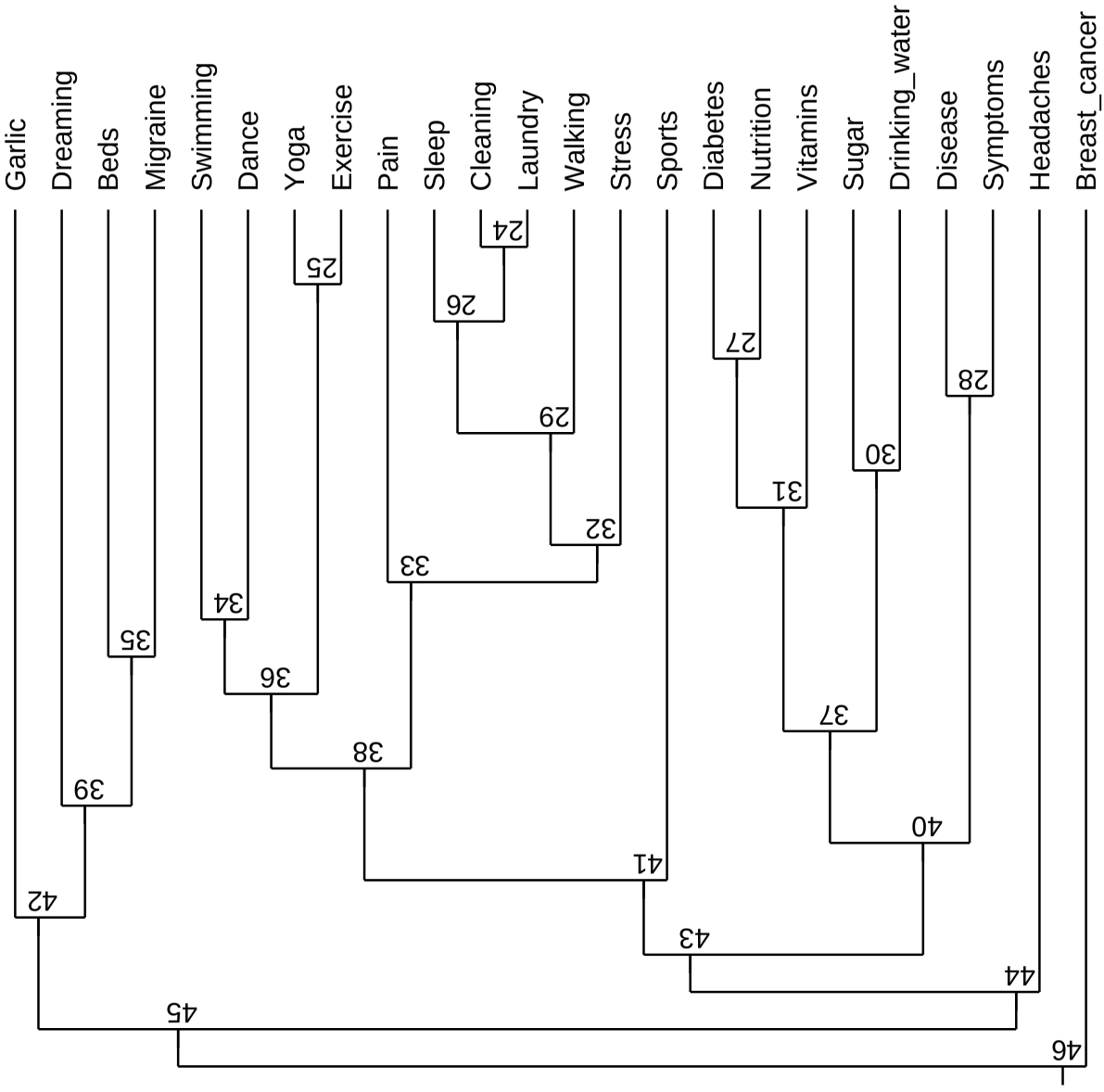

Figure D.4: Clustering results using base assignments

Appendix E

Wikipedia graphs

The following page contains the 10 sub-graphs of the Wikipedia graph of concepts after pruning. The following concepts also exist after pruning, however they do not have a relation to other concepts in the structure:

- Aging-associated diseases
- Diabetes
- Disease
- Immune system disorders
- Laundry
- Nail care
- Skin care
- Skin conditions resulting from physical factors
- Sleep disorders
- Syndromes

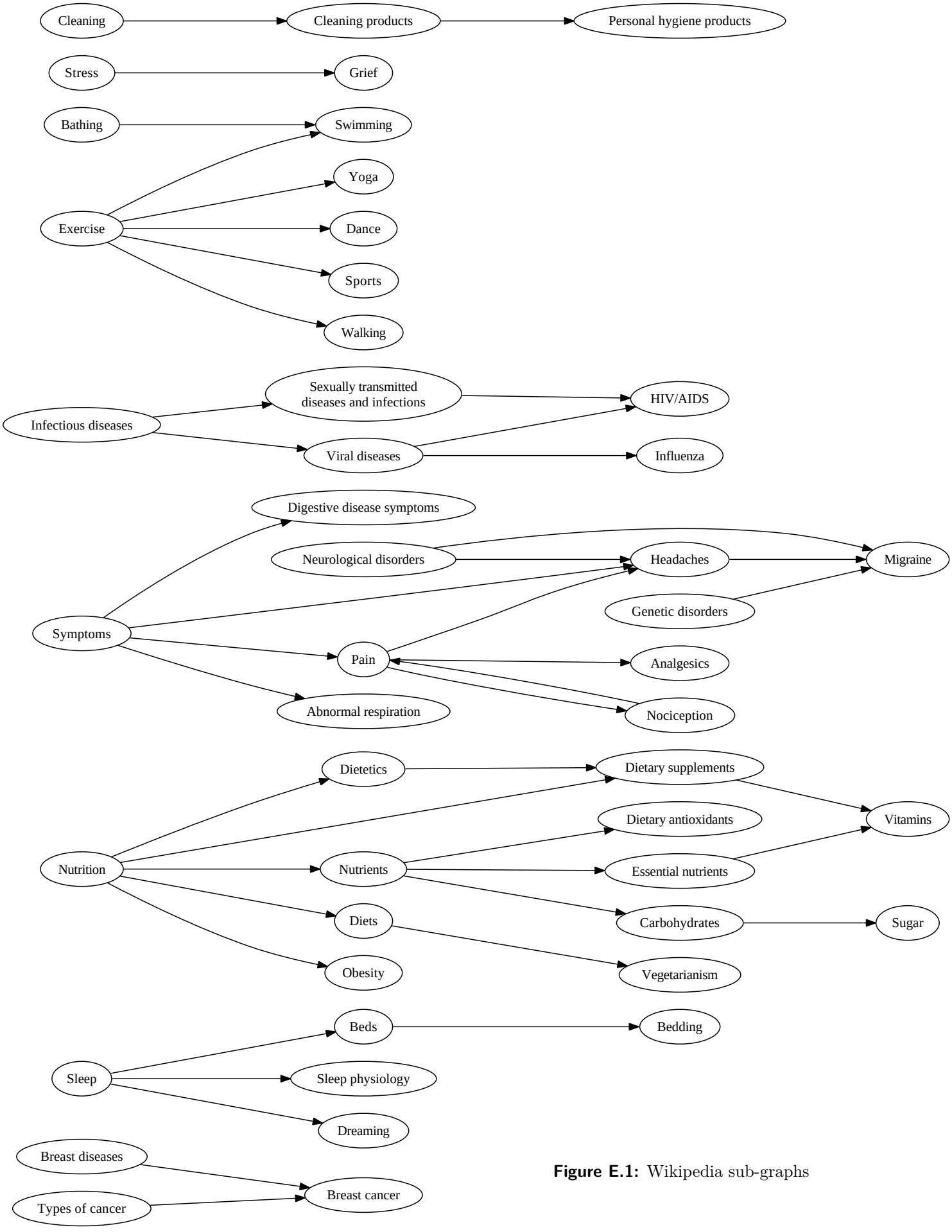

Figure E.1: Wikipedia sub-graphs