Automatic Creation of Opinion-Based Summary Representations

Master's Thesis

Ronald Bezemer

Automatic Creation of Opinion-Based Summary Representations

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Ronald Bezemer
born in Graafstroom, the Netherlands

Section Interactive Intelligence
Department of Intelligent Systems
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Automatic Creation of Opinion-Based Summary Representations

Author: Ronald Bezemer
Student id: 1532901
Email: ronald@prbezemer.nl

Abstract

Customers interested in buying a product, can search on the internet for reviews about that product. For many products, an enormous amount of information and opinions is available. Customers gets overwhelmed by this information and systems are needed to filter out the essential information.

In this research, a model is developed to automatically create a graphical summary of a set of Dutch product reviews. This model is divided into three sub modules. The first module is meant to select sentences expressing opinions. The second module groups those sentences based on their subjects. Finally, the third module generates a graphical representation of the groups of sentences and presents it to the customer.

The most important parts of the model are implemented into a system. An user experiment is conducted to evaluate the performance of this system. This experiment shows that the first module has an accuracy of 81%. This is comparable with systems developed for other languages. Information shown in the graphical summary representation is judged as relevant in 56% of the cases.

Although further research is needed, this model is a good basis to provide insight to the customer about essential information in product reviews.

Thesis Committee:

Chair:	Prof. Dr. C.M. Jonker, Faculty EEMCS, TU Delft
University supervisor:	Dr. ir. P. Wiggers, Faculty EEMCS, TU Delft
Committee Member:	Dr. J.A. Redi, Faculty EEMCS, TU Delft
Committee Member:	Dr. A.E. Zaidman, Faculty EEMCS, TU Delft

Preface

The work you have before you is the result of my Master's graduation project at Delft University of Technology, at the section Interactive Intelligence. I would like to thank everyone involved in providing me with facilities, data and time to discuss my system and thesis.

First, Pascal Wiggers was my supervisor during this project. Pascal, I am grateful for your support and guidance. The meetings with you were good moments to reflect and to look ahead.

Second, I want to thank the persons who have annotated hundreds sentences with a polarity value. Also, I thank the participants of the user experiment used to evaluate the presentation part of the system.

Furthermore, I thank Yavuz Bocu of Beslist.nl for his cooperation. Thank you for sharing your product reviews with us. Without this data, the final product would not have been as it is.

Finally, I want to thank my family, friends and wife for supporting me throughout my thesis project. Nellina, thank you for your kind assistance and not in the last place, the cups of coffee you made for me on the days we worked together at home.

Ronald Bezemer
Delft, the Netherlands
April 17, 2012

Contents

List of Figures

Chapter 1

Introduction

Good summaries are indispensable in this information era. Every day, an enormous amount of data is available for everyone. Although this amount is too large to process, people don't want to miss essential information. Therefore, systems are needed which can process this information automatically and provide condensed summaries to humans.

When humans process information, they have always feelings about it. Those feelings will influence the processing of this information. It has been shown that emotional information is better remembered and recognized than neutral ones (Ferré, 2002). Therefore, it seems that a relation exists between emotion and the way in which information is processed.

Emotion can be split up into affect and sentiment. By affect we mean the underlying state and sentiment is the expression of this state (Pang and Lee, 2008). For example if somebody uses many positive words, this can be an expression of a very happy underlying state. The computer scientific analysis of those phenomena is called sentiment analysis. Since affect and sentiment are very essential for human live we cannot neglect these signals in information.

In this project we combine the fields of summarization and sentiment analysis. We try to make better summaries of texts by evaluating the affective content in these texts. This affective content is summarized and presented to the user. To do so, the affective content is recognized and the results are shown to the user of the system.

1.1 Scope

Both sentiment analysis and summarization are broad research fields in their own right, therefore the scope of this research should be defined. Sentiment analysis is applied to several modalities, including gestures, speech and text. Although summarization is done in several modalities, most research is carried out on text. in this research the focus is on the textual modality as well.

It is obvious that not each text in the textual domain contains an equal amount of affect. For example, it is expected that scientific papers contain less affective information than personal letters. Texts which contain much affective content are for example commentaries in newspapers, online reviews and blogs on the internet. The research field of affective

computing is quite young and therefore much explorative research is needed. Therefore we need a domain which contains much affective content. If we are able to successfully analyse the affective content in those domains, other domains can be treated. On the other hand, if we aren't able to analyse the affective content in those domains, the change of successfully analysing other domains is small .

As said before, the field of affective computing is young. This is in an increasing degree the case for the Dutch field of language. Therefore, the focus of this research is on Dutch. This makes it a very challenging task since the biggest part of all available resources are dedicated for the English domain. Only a few resources are available and the other should generated by our own.

We have decided to use the domain of online reviews. There are several websites which provide opportunities to write a review about products, services or accommodations. Other customers can use these reviews before buying a product or making a trip. Customers can write a review in natural language, and in addition to it, they can give grades to several aspects of the service or item. This gives us the opportunity to use the provided grades to analyse the affective content in a text. Therefore, those reviews can be used in our approach.

1.2 Research Questions

In this section the research questions are described.

First we need to know which investigations are already done in the fields of interest. This is expressed in the first research question.

1. What is state of the art of sentiment analysis and summarization?

To perform sentiment analysis, a system is needed to import and handle texts. These texts should be analysed and the results should be shown to the user. To do this, a computer model is needed. Therefore we would like to know the key concepts of this model. This bring us to the question:

2. What are the key concepts of a model that can reason about the affective states in a text?

An elemental part of the above-mentioned system is the sentiment analyser. The task of this part is to distil affect from a sentence, paragraph or even a whole text. But how can the system recognize sentimental information? It seems to be a pattern recognition task and for such tasks it is very obvious to use features.
Therefore we come to the next research question:

3. Which features can be used to analyse affect?

Affect is a somewhat abstract term and there no representation available which is proven to be the best one. Therefore several different representations are proposed, including dimensional, categorical and component based ones. Investigation is needed to select the most appropriate option. This is expressed in the next question:

4. Which affect representation is the most appropriate one for an affect analyser?

There are several methods to present an abstract. While textual representations are widely used, graphical representations are sometimes more intuitive. For example, the weather forecast can be written out completely, but using a map to present information such as temperature and wind speed enables rapidly information exchange. Possible, a graphical representation can be used for a summary of reviews. This is meant by the next question.

5. What kind of model can be used to graphical represent a summary of reviews?

1.3 Outline

The structure of this thesis is as follows: First a theoretical overview is given about the field of Sentiment Analysis. This is described in chapter 2. After it, in chapter 3, a literature overview is given of the field of text summarization. After those two chapters of theory, the tools used in this project are described. Chapter 5 contains a presentation of the designed model. This model is implemented and the resulting system is described in chapter 6. The system is evaluated and the results are presented in chapter 7. Finally, in chapter 8, a conclusion is given.

The reader without much time is recommended to read at least the chapter about the developed model, chapter 5, the chapter with evaluations, chapter 7 and the conclusion, chapter 8.

Chapter 2

Sentiment Analysis

Sentiment analysis is a sub field of affective computing which has become very popular in the last decade. By sentiment analysis we mean the computational treatment of opinion, sentiment and subjectivity in text. Many papers have been published in different sub fields. The reason for the growth is twofold. The first point is the demand to make the interaction with computers more like human interaction. This requires that computer programs can recognize emotional expressions and also that computers can generate emotional behaviour in some way (Picard, 2000; Pantic and Rothkrantz, 2003). Since the computation power of computers is still growing, it is possible to perform heavier computations.

The second reason is more specific for the text domain and is about the increasing number of online available customer reviews and other digital information. Customers can easily share there experiences with products or services on the internet. Those reviews are read by others which are looking for such kind of product. According to a survey under American adults, customers are willing to pay 20% to 99% more for a 5-star-rated item than a 4-star-rated item (the variance stems from what type of item or service is considered) (Horrigan, 2008). It is clear that insight in those reviews will be very desirable for both customers and marketeers.

First, a list with possible application of sentiment analysis are given. After it, different representation methods are discussed. The third subject is about affect detection in the textual domain. We end up this chapter with a discussion of datasets and evaluation.

2.1 Applications

Sentiment analysis is a very broad field with diverging applications. In this section different possible applications are described, all limited to the text domain. The main goal is to provide an overview of the existing applications.

Many people search on the internet for product reviews before buying that product. Therefore some kind of review- and opinion-aggregation website is suggested by (Pang and Lee, 2008). They mean a website which proactively gathers feedback and review information. This information is not limited to product reviews, but includes opinions about candidates running for office, political issues and so forth.

A sentiment analysis system can also be used as a sub-component of a broader system. In the literature it is used as an augmentation of a recommendation system by Terveen et al. (1997); Tatemura (2000). For advertisements purposes, a system that detect sensitivity can be useful (Jin et al., 2007). This system detects if a webpage contain sensitive content inappropriate for ads placements. The focus of this application lies on detecting when it is not appropriate to show ads, but one can argue that it will be useful as well to show ads when relevant positive content is detected. Other examples of sub-component systems include: question answering, citation analysis and human-computer interaction (Pang and Lee, 2008).

In the field of business and government intelligence many applications are possible. For example, if a manufacturer is confronted with low sales, the opinions of internet users can be investigated by using sentiment analysis systems. The manufacturer can adapt his strategy on the basis of these analyses. By tracking public viewpoints it will be also possible to predict trends in sale or other relevant events.

All those application need to compute with emotion. Therefore a quantitative representation of emotion is needed. This is described in the next section.

2.2 Representation of Emotion

This section describes how emotion can be categorized and represented in a model. The way of representation is mostly based on theories of emotion. The difference in theories is in part caused by the different research areas. Emotion is studies from a physiological, social, biological and psychological point of view. The last decade it has also been studied by computer scientists. But they have focused their efforts on the challenges of developing affect-sensitive computer interfaces. And, according to (Calvo and D'Mello, 2010, p. 19) they "have remained agnostic to the controversies inherent in the underlying psychological theory". It is essential for computer scientists, they argue, to integrate emotion theory into the design of affective computing interfaces. Therefore, first different models for representing emotions are discussed.

What is a good and workable representation of emotion? Do we use a limited set or a dimensional representation? Those questions will be answered in the next paragraphs. In the literature different approaches are described which can be split up into three different groups. Those approaches are: discrete approach, dimensional description and a description at the basis of components (Hudlicka, 2009). It is good to emphasize that it are all theories and that they should not be viewed as competing for a single ground truth. But rather as distinct perspectives that resulted from different research traditions. As humans and in particular emotions as so complex they cannot exactly be reduced to a theoretical framework. Therefore it is best to view the described theories as alternative explanations, each with its own set of explanatory powers and scope, supporting data, in analogy of a particular theory, as suggested by Picard (2000).

2.2.1 Discrete theories

First the discrete theories of emotions are described. Those theories represent an approach to studying emotions that emphasizes a small set of discrete or fundamental emotions and a view that the emotion itself is the fundamental semantic 'primitive'. The number of fundamental emotions depends on the research traditions but are typically the 'basic' emotions; joy, sadness, fear, anger and disgust (Ekman and Others, 1992).

The underlying assumption of this approach is that the fundamental emotions are directly associated with the neural circuitry. They contain a large innate, 'hardwired' component and are characterized by patterns of triggers, behavioural expression and associated distinct subjective experience. Indeed, some emotions can be explained by such characterizations, but that is not the case for all (Panksepp, 2004).

2.2.2 Dimensional theories

In the second group of approaches, emotions are described in terms of a small set of factors, or dimensions. Mostly two or three factors are used to create a space in which distinct emotions can be located. The most frequently used dimensions are valence and arousal. Valence reflects a positive or negative evaluation and the associated felt state of pleasure vs displeasure. Arousal reflects the degree of intensity or activation of an affective state. It reflects the readiness to act: low arousal reflects less energy to act, high arousal reflects more energy. Models which contain factors like arousal and valence are suggested by Watson and Tellegen (1985); Russell (1980).

Although a two dimensional valence-arousal model is a good way of representing different emotions, there are limitations to this model. For example the emotions fear and anger are both defined by a high arousal and negative valence, but there corresponding experiences and behaviour are quite different. The interested reader is referred to Larsen and Diener (1992) for an discussion about the limitations of two-dimensional models. For this reason, others have suggested models with a third dimension, suggestions include: energetic and tense arousal (Thayer, 1996), hedonic tone, energy and tension (Matthews, 1990). But the most frequently used third dimension is dominance (Osgood et al., 1971; Russell and Lanius, 1984). Dominance corresponds to "a sense of power and control, ability to cope with a situation, and associated willingness to act" (Hudlicka, 2009, p. 123). By using dominance in combination with valence and arousal, the emotions of for example anger and fear can be separated. The patterns of behaviour associated with anger (e.g. approach and attack behaviour) have a high dominance while the patterns of behaviour of fear (e.g. freezing behaviour) corresponds to low dominance values.

The distinct dimensions have been associated with underlying neuroendocrine systems that control motivation and behaviour and the intensity of their interaction (Hudlicka, 2009). Generally, the valence dimension is associated with two neural systems, corresponding to the mediation of positive and negative emotions. Those systems are directly associated with behaviour pattern of respectively survival and withdrawal. For the other dimensions, similar associations are suggested.

2.2.3 Component process theories

The third type of theories are the component process theories. Those theories emphasize a set of domain-independent features of the situation and the appraising agent: the appraisal dimensions. These theories generally contain three key concepts (Hudlicka, 2009). First, emotion is multi-modal and the 'components' are: the cognitive system, mediating appraisal, the autonomic nervous system, mediating arousal, the motor system, mediating expression, the motivational system, mediating action tendencies, the monitor system and mediating feelings (Scherer, 2000). The second concept is the existence of stimulus evaluation checks in which values are calculated of the individual appraisal dimensions. The third concept is parallel processing of the multi-modal components. This is needed since the components can have multiple complex feedback relationships with other components.

The main difference between the dimensional theories and component process theories is that dimensional theories focus on analysing the felt state of the agent while component process theories reflects the stimuli triggering the emotion, and the agent, and the relationship of the agent to its environment. The set of dimensions of component process theories is much larger and as a consequence of it, has a much larger space. Therefore, also very complex affective states can be represented, including mixed states. The other way around, however, is not so trivial. Which emotion corresponds to a given set of multi-modal values? For this question, no easy answer exists yet.

In the section above, three different representation metrics of emotion are discussed. It is clear that not each metric is even suitable for automatic computation. This is also visible in the way in which affect detection is applied on text. Therefore, in the next section we will discuss the techniques used to detect affect from text.

2.3 Affect Detection from Text

As mentioned before, emotion is studied in many different research fields. An important reason for it is that emotions are multi-modal, that is, they are expressed in multiple channels (e.g. face, voice and text). Each channel has its own advantages and disadvantages. A multi-modal analysis of emotions provides therefore challenging opportunities. For example, if facial analysis fails, other channels can be used to detect the actual emotional state.

The focus of this research is on the text domain. It refers to written natural language and transcriptions of oral communication. Therefore, multi-modal analysis is not possible. For that reason only affect detection in the textual domain is discussed. The interested reader is referred to Calvo and D'Mello (2010) for an extended discussion about the other domains. Below, first cultural differences are discussed. After it, several approaches for affect detection from text are described and finally a general solution is given.

2.3.1 Cultural Difference

Before we can do research to the affect in text, it should be clear how people express emotions through text. Is it the same in different cultures? Researchers have tried to answer this question. Osgood et al. (1971) used multidimensional scaling (MDS) to create visualizations

of affective words based on similarity ratings of the words provided to subjects from different cultures. Those words are not limited to words which express emotions, but can be all things (e.g. mother, coffee, logos and even colors). He found that the dimensions "evaluation", "potency" and "activity" can be used to describe emotions in different cultures. Those dimensions are qualitatively similar to valence, arousal and dominance discussed above.

Although Osgood has found similar matrices produced by people of different cultures, this is not the case for the research of Lutz and White (1986). They found differences in the similarity matrices produced by people of different cultures. However, Samsonovich and Ascoli (2006) have compared English and French dictionaries (by using synonyms and antonyms) and found the same set of underlying dimensions. This shows that actually no agreement exists. Therefore, resources developed for a specific language could be used only very careful in another languages.

2.3.2 Corpora-based

Corpora-based approaches are used to categorize texts into a limited set of emotions. Those approaches are based on the assumption that people using the same language would have similar conceptions for different discrete emotions (which is controversial, as described above). Researches have build thesauri of emotional words. An electronic lexical database which has been used by many is WordNet (Fellbaum, 1998). Strapparava et al. (2006) extended WordNet with information on affective terms.

Another corpora-based approach is the Affective Norm for English Words (ANEW), a project to develop a set of normative emotional ratings for a collection of emotional English words (Bradley and Lang, 1999). This collection provide values for the dimensions valence, arousal and dominance. These values are the averages of the ratings of a large group of subjects. While this approach is used to find emotional ratings for single words, the Affective Norm for English Text (ANET) is a similar list for complete texts (Bradley and Land, 2007).

2.3.3 Lexical

Another method for affect detection applies a lexical analysis of texts to identify words which indicate the affective states of the writer. Several investigations uses the Linguistic Inquiry and Word Count (LIWC), a validated computer tool that analyses bodies of text using dictionary-based categorization (Pennebaker et al., 2001). LIWC uses more than 70 language dimensions to determine the degree of positive of negative emotions. Those methods try to identify particular words that are expected to disclose affective content in text. For example, first person singular pronouns in essays (e.g., "I" and "me") have been linked to negative emotions (Calvo and D'Mello, 2010).

2.3.4 Semantic

The above described methods are based on word matching. Other approaches are suggested which perform a semantic analysis of text. Those techniques are more or less the same as used for summarization, described in chapter 3. The similarity is measured between a text and emotional concept words. This works well for joy and fear, but it fails for text

conveying other emotions such as anger, disgust and sadness (Lund, 1996). According to Calvo and D'Mello (2010, p. 28), "it is an open question whether semantic alignment of texts to emotional concept terms is a useful method for emotion detection".

Other, more complex methods use deeper information by constructing affective models from large corpora with world knowledge. Those models are used to identity the affective states in text.

2.3.5 General approach

In all cases discussed above, a sentiment analysis system consist of two modules, namely extraction and classification (Pang and Lee, 2008). In the first module the 'sentimental' parts of a text are marked or extracted, that is, the parts which are likely to contain information about the affective state. This is essential for texts in which multiple subjects are discussed. The author can have different feelings of different subjects. In the second module, this output is analysed and labelled with a particular emotion, state (e.g. positive or negative) or on a continuum (in case of dimensional emotion representation). In both modules, the above described approaches can be used.

Much of the work in sentimental analysis only classifies a text as falling under one of two opposing sentiment polarities, also called sentiment polarity classification. A variant is to locate its position on a continuum between these two.

2.4 Datasets

There exist multiple English datasets which can be used for evaluating sentiment analysis systems. See Pang and Lee (2008) for an overview. Those datasets consist of labelled data. These labels were obtained by manual annotation. However, manual annotation is a time consuming task and therefore very expensive. Researchers tried to find other ways to acquire the labels. Researchers have taken advantage of Rotten Tomatoes, Epinions, Amazon and other sites where users furnish ratings along with their reviews (Pang and Lee, 2008). Those ratings can be used as labels. But it will be clear that those labels can only be used for classifying data into a positive, neutral or negative class, other emotional dimensions are not available.

For system evaluation, those labels can be compared with the output of the sentiment analysis system. In this way the performance of the system can be determined.

2.5 Evaluation

There are also evaluation campaigns in which researchers compare their systems. One of those competitions is the Blog track, involved by TREC. TREC aims to support Information Retrieval research. In the years 2006 - 2008 the focus was on the opinionated character that many blogs have. The participating systems had to retrieve the blog posts expressing an opinion. The results where analysed by Ounis and Macdonald (2008) and they found that lexicon-based approaches constituted the main effective approaches. They also found

that opinion-detecting ability and relevance-determination ability seemed to be strongly correlated.

In the Blog-TREC, a standardized and controlled test collection is used. This collection consists of three components: a collection of documents; an associated set of information needs and a set of human relevance judgements. The resulting test set is called: Blogs06 and can be used freely[1].

The NTCIR multilingual opinion analysis task is a similar program, but contains blogs in Japanese, Chinese and English (Evans et al., 2007). The focus is on detecting opinionated sentences and opinion holders. Optional was the task of polarity labelling and detecting of sentences relevant to a given topic. The differences between languages makes direct comparisons difficult and strict and lenient evaluation standards are used.

Evaluation of sentiment analysis systems is still a challenging task. There are datasets available with labelled data which can be used as training and test sets for the systems. Some datasets are the result of evaluating campaigns, and therefore the scores of other systems are available, which gives these sets extra value.

2.6 Conclusion

In this chapter, different applications of sentiment analysis are described. We have seen that such systems have advantages for both customers and marketeers. After it, three different representation methods for emotion are described. Discrete theories, dimensional theories and component process theories are discussed. We found that dimensional theories are most suitable for analysis in the text domain. After those representation methods, technical approaches are described. We end up with data-resources and evaluation campaigns.

[1] `http://ir.dcs.gla.ac.uk/test_collections`

Chapter 3

Summarization

For more than 50 years researchers tried to build systems that are able to generate summaries for given texts (Luhn, 1958). People had high expectations of computers and thought that almost all problems could be solved. But unfortunately it is so far not possible to automatically generate summaries perfect. In this chapter the state of the art of summarization is discussed.

It is quite hard to make an appropriate categorization for the research applied in this area. Others have used classifications based on input, purpose and output factors, extractive vs non-extractive (Sparck Jones, 2007), single-document vs multi-document approaches (Das and Martins, 2007), classical vs knowledge rich methods (Jezek and Steinberger, 2008) or on surface, entity and discourse level (Mani and Maybury, 1999). Here, the subject is split up into different important factors. First, a definition is given, challenges are discussed and different types of summarizers are presented. After those subjects, trends in summarization are discussed.

3.1 Definition

What do we actual mean with summarization? Considering that summarization can be applied to different modalities, in this research, the focus is on text summarization. Mani and Maybury (1999) describe summarization as follows: Text summarization is "*the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)*". The focus of this definition is on distilling information for a user and a task. This is indeed a possible definition, but one can get the idea that summarization is nothing more than distilling sentences from a text. Although many systems are actually solving the problem in this way, there exist also other systems which generate their own sentences. Therefore, we use the more specific definition from (Sparck Jones, 1999). Sparck Jones defines a summary as "*a reductive transformation of source text to summary text through content condensation by selecting and/or generalisation on what is important in the source*". In this definition, summarizing is thus about both transformation and expression. This is broad enough for the scope of this project.

3.2 Challenges

Though the field of automatic summarization has been investigated for many years, the key challenges have not changed during this period. One of most difficult problems is still caused by word sense ambiguity. This problem is inherent to natural language and means that a word can have multiple meanings. This happens very frequently and examples can be found everywhere. Compare for example the word *cold* in the next sentences: "*I am taking aspirin for my cold*" vs "*Let's go inside, I'm cold*". In the first sentence the disease sense is intended, in the second sentence the temperature sensation sense is meant. Another example is "to record a concert from a broadcast" vs "in record time". The sense of the first is the verb record and in the second it is intended in the sense of performance.

In some cases, it depends on the domain which sense is intended. This becomes clear from the work of Schuemie et al. (2005). He reviewed the current state of the art in Word Sense Disambiguation with the focus on the biomedical domain. An example of word sense ambiguity in this field is the acronym PSA. PSA maps to the thesaurus concept of "prostate specific antigen", but is also used in MEDLINE with the not-in-thesaurus meaning "poultry science association". From this example it will be clear that word sense ambiguity is domain specific. So, the challenge is to find the know-how to resolve the word by using both the words in context and domain specific information. Word sense ambiguity can consist of morphology, collocation, part-of-speech, semantic word association, and so forth (Foong et al., 2010).

Another challenge which occurs particularly in multi-document summarization is overlapping themes. This means that a more or less similar paragraph exists in multiple documents. This is especially the case while summarizing multiple news articles about one topic. Carbonell and Goldstein (1998) have presented the Maximal Marginal Relevance criterion which strives to reduce redundancy in selecting appropriate passages for text summarization. It is shown that MMR results are clearly superior to non-MMR passage selection.

Long sentences are as well a challenge. A long sentence can contain much information, both relevant and non-relevant for summary purposes. Therefore a good summary system needs to pick up only relevant parts of a long sentence. To do so, the system needs knowledge in Natural Language Processing to perform sentence reduction.

The last challenge discussed here is the availability of quantitative evaluation methods for automatic text summarization. The problem is a deep one, according to (Foong et al., 2010). We need to compare the system with an excellent summary, a so-called baseline. Those baselines can be compared to the output of the concerned system by using one of following comparison methods: summary to source, system to human-generated summary and system to system. Unfortunately, human-generated summaries are not unambiguous and differ in mutual consensus. Therefore, the problem of matching a system summary against the ideal summary is very difficult to establish (see section 3.6).

3.3 Types

As mentioned in the introduction, summarizers can be categorized in several ways. It is even possible that a system occurs in more than one category. That's not a problem at all,

but the reader should be aware of it. The mentioned types can be used to compare different systems.

In this section different types of summarizers are given. The type of a system depends on the design choices made in the development phase. According to Sparck Jones (2007) those design choices can be divided into three classes of *context factors*, namely input, purpose and output factors. Input factors characterizing the source material including style and units, *purpose* factors including intended use and audience and *output factors* including reduction and format. In some cases, the output is fixed by the purpose factors, but mostly multiple different output formats are possible.

3.3.1 Input factors

First the input factors are discussed. Is it important to take care of the form of input, for example language, length, genre, medium and structure. The requirements for a system which should be able to process input documents written in different languages, differ from systems for one language. Also the linguistic style is important. Technical articles contains different words than dialogues, meetings and email messages.

Other input factors include information as metadata, author and subject type. As last mentioned input factor, units are mentioned. In many cases, a summary of a single document is needed, but with news streams a summary of multiple documents is needed. In the last case the input unit is not a single document but a topic.

3.3.2 Purpose factors

The purpose factors describe the intended use. Jezek and Steinberger (2008) give three different types of use: indicative, informative and critical/evaluative summaries. An indicative summary should preserve the most important passages and the typical length ranges between 5 till 10% of the complete text. An informative summary is typical 20-30% of the complete text. The last category captures the point of view of an author on a given subject. Indeed, the length of the output is determined by a purpose factor. This is also the case for the audience. Authors who write academic abstracts, assume a technically-informed audience like the one for the source. But summarizing 'interesting' science papers for newspapers assumes a different type of audience. Therefore the terms used in the summary should match the knowledge of the audience. As last purpose factor, Sparck Jones mentions time, location, formality and destination.

3.3.3 Output factors

The last class of factors is output factors. Input and purpose factors constrain output, but do not determine it. Many choices remain, e.g. sentence syntax choices. As first the type of summary is concerned. This can be found in (Jezek and Steinberger, 2008). If the summary consist completely of word sequences copied from the original document, it is called *extract*. The words in the summary are extracted from the source document. The word sequences can be words, sentences or even complete paragraphs. Problems which may occur with this type are: inconsistency, lack of balance, broken anaphoric references and lack of cohesion.

The other form of summary is called *abstract*. In this the summary may contain word sequences not present in the original text. It is still a very challenging task, and therefore most of the current systems generate extracts. Other factors concerning the output are language, linguistic style, medium, structure and genre. Those are the same as the factors regarding the input document.

In this section we have seen the factors which cause different types of summarizing systems. Based on the choices made for input, purpose and output factors, multiple different systems can be constructed. Those systems will mutually differ, but this does not mean that one solution exist which is sufficient in all situations. Therefore, those factors should be kept in mind while evaluating systems. But system evaluation is described in another section (see section 3.6).

3.4 Trends in Text Summarization

The different ways of processing can roughly be spit up into surface- and deeper-level approaches, according to Jezek and Steinberger (2008). Surfaces approaches represent information as shallow features such as: statistical and positional terms, cue phrases and domain-specific terms. Deeper-level approaches uses some semantic analysis and Natural Language Processing to determine salient parts.

3.4.1 Surface approaches

Systems with a surface approach uses location information (e.g. header, first sentence, last sentence) and statistical information such as word frequency, but less knowledge. Therefore, these are shallow approaches.

The first investigation in the field of summarization is done by Luhn (1958). He uses statistical information derived from word frequency and distribution to compute a relative measure of significance. These numbers are computed first for individual words and then for sentences. The sentences with the highest significance are extracted and used in the 'auto-extract'. In sum, relevance is measured by word frequency.

Other shallow indicators of relevance are also investigated. Edmundson (1969) has demonstrated that by the use of a combination of cue-words, title words, and the position of a sentence, an abstract can be generated similar to an human one.

Others have investigated this idea further and tried to decrease the impact of frequently used words. This leads to the widely used $tf \times idf$ relevance measure in which the relevance of a term is inversely proportional to the number of documents in the corpus containing them (Salton and Buckley, 1988). The $tf \times idf$ value exists of a inverse document frequency and a term frequency. The term frequency $tf(t,d)$ is simple a count how often term t occurs in document d. The inverse document frequency can be computed as follows:

$$idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

with

- $|D|$: cardinality of D, or the total number of documents in the corpus
- $|\{d \in D : t \in d\}|$: number of documents where term t appears.

Then the $tf \times idf$ is calculated as

$$tf * idf(t,d,D) = tf(t,d) \times idf(t,D).$$

Another method in text summarization is Latent Semantic Analysis (LSA) (Ozsoy, 2011). It can be used to analyse relationships between a set of terms. The assumption of LSA is that words which are close in meaning, will occur close together in text. Although this assumption is too strong from a linguistic point of view, it leads to good results.

To use the relationship information the next three steps are performed: First a matrix is generated for the input document, second a singular value decomposition is applied on this matrix and finally sentence selection is performed. See Ozsoy (2011) and Steinberger (2004) for a detailed explanation of this method.

3.4.2 Deeper approaches

The above described approaches uses mostly statistical and positional information. In this section some deeper approaches are described which make use of the discourse structure in a text. It has been shown that the discourse structure plays an important role in the strategies used by human abstractors and in the structure of their abstracts (Mani and Maybury, 1999). Therefore, a summary is not just a collection of salient sentences, but is well organized, coherent and represents the argumentation used in the source.

Discourse models can be classified in terms of the linguistic distinction between cohesion and coherence (Halliday in Mani and Maybury, 1999). Text cohesion involves the relations between words or referring expressions. Some examples of this are: anaphora, ellipsis, conjunction, reiteration, synonymy and hypernymy. Models based on coherence represent the overall structure of a multi-sentence text in terms of macro-level relation between sentences or clauses. The example given by Mani makes it clear: the cue phrase "in order to", one could argue, expresses some sort of purpose relation between clauses; likewise clauses linked by "although" express some sort of contrast relation.

Boguraev and Kennedy (1997) use an entity-level approach, by exploiting relationships based on text cohesion. They aggregate and select phrasal terms by syntactic parsing and recognition of anaphoric relationships between the terms. Those relations are determined by ranking all the possible antecedents at the basis of local features. They claim 75% accuracy. The summary is produced by using a segmentation method relying on similarity between blocks of text based on vocabulary overlap. A global "discourse salience" measure based on local salience and frequency calculations is constructed. This measure is used to identify the most globally salient entities which they call "topic stamps". The actual summary is a list with co-referential phrases associated with the topic stamps, completed with some information from the surrounding context.

Another approach which make use of text cohesion relationships is carried out by Barzilay and Elhadad (1997). They produce a summary by exploiting "lexical chains", namely,

sequences of related terms grouped together by text cohesion relationships. This include repetition, synonymy, hypernymy, antonymy and holonymy (part-of relation). They suggest that the reader will get a better overview of the text by grouping together words into lexical chains than by using the most frequent words. Sometimes, a chain of low frequent words is more indicative for a salient topic than high frequent words, they argue. The chains are build in two stages: first, chains are build for individual text segments; then, the chains of different segments are merged when they contain a common term with the same sense. Finally, sentences are extracted from chains based on a variety of heuristics. The authors give three limitations of their approach. The first deals with sentence granularity. Only whole sentences can be extracted as single units. The second is that extracted sentences can contain anaphora links to the rest of the text. The last mentioned limitation is that their method does not provide any way to control the length and level of detail of the summary.

The two papers described above, use text cohesion, but as mentioned above, text coherence is also an option. One well-known approach is to use the rhetorical structure theory (RST). It is a descriptive theory of a major aspect of the organization of natural text (Mann and Thompson, 1988). It describes the structure of a text in terms of relations that hold between parts of the text. Therefore, two kinds of nodes are defined, namely a nucleus node (which represents information which is more essential to the writer's purpose) and a satellite node. Between those nodes predefined relations can exist. Some examples of those relations include: antithesis, background, motivation, concession, contrast, elaboration and justification. The use of RST is illustrated by the next example.[1] It is a short newspaper editorial with a political purpose. It shows an argumentation style in order to persuade the reader. From this example it will be clear how a RST graph can be and also that there exist many different relations.

Assuming that nuclei are more salient than satellites, salience of information can be determined based on tree depth. Those salience information can be used to extract sentences or clauses to form summaries. Some research which applies RST includes: (Ono et al., 1994; Marcu, 1997; De Vries, 2005). An extensive overview of applications of RST can be found by (Taboada, 2006).

Another approach which uses relations is Relational Frame Theory (RFT). Relational frame theory is a form of functional contextualism, a behavioural approach in psychology (Greenway et al., 2010). This theory is based upon the ability of humans to make relations between stimuli. Key properties are: mutual entailment, combinatorial entailment and transformation of stimulus function. Empirical demonstrations of linguistic phenomena are described with practical examples. There are not yet natural language systems developed which apply this theory. Although Greenway et al. (2010) have suggested applications, they are not yet implemented. Likely, summarizers might benefit from this theory.

3.5 Summarization and Affect Detection

In chapter 2 the subject of sentiment analysis is described. In this section, the subjects sentiment analysis and summarization are combined. Can sentiment analysis be used to

[1] This example is taken from `http://www.sfu.ca/rst`

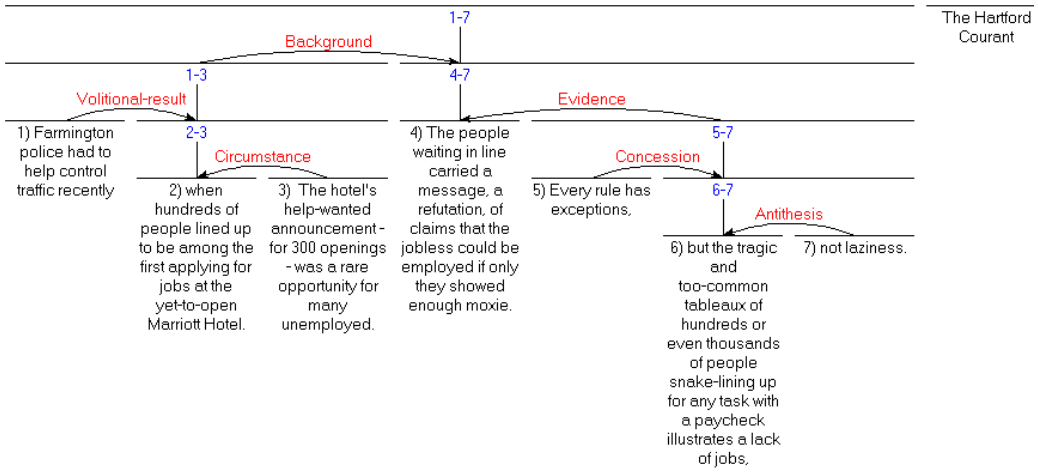


Figure 3.1: An example of an RST graph

build a summary? So, we would like to create an abstract of the affective content in a document.

An investigation which combines summarization and subjectivity analysis, is the work done by (Seki et al., 2005). They have build a system for the multi-document summarization task of the Document Understanding Conference (DUC) 2005. This system extract sentences from the source documents for the summary. Their system consists of two stages. The first stage is the paragraph clustering stage. Here the different documents are segmented into paragraphs and the term frequencies are computed for each paragraph. Based on the Eucledian distances between feature vectors, the paragraphs are clustered, using Ward's method (Ward, 1963). The next stage is called the sentence extraction stage. Here, for each cluster, a feature vector is computed with term frequency and inverse cluster frequencies. Those clusters are ordered and hereafter each sentence in a cluster is weighted based on several factors including location, number of content words, number of heading words and term frequencies. For the subjectivity analysis, a boolean factor is added, which is 1 if a sentence is subjective and 0 otherwise. From each cluster one sentence was extracted and conjunctions, such as "And", "But", "However", at the beginning of a sentence were removed and the initial character of a sentence was capitalized.

For the subjectivity analysis they used a Support Vector Machine (SVM) with features based on polarity type frequencies by using adjective entries and General Inquirer (Stone, 2000). As features they used the frequency of polarity adjectives (positive and negative), gradable adjectives (positive and negative) and finally positive and negative words, both strong and weak. But, finally all the sentences falls under one of the categories: subjective or objective. They report improvements for several selected topics.

The limitation of this research is that they only use a boolean factor for subjectivity and don't specify the polarity or even the intensity. By doing this, much sentimental information

is dropped.

Others have used subjectivity classification in combination with Information Extraction (IE) (Riloff and Phillips, 2005). The goal of IE systems is to extract facts from texts related to a particular domain. They observed that many incorrect results of those systems are caused by subjectivity. To overcome this problem they classified each sentence in a text as objective or subjective. In their system, only (more or less) objective sentences are used for the actual information extraction. To classify sentences as subjective they have used the system developed by (Wiebe and Riloff, 2005). In this system, the classification is done in different steps. In the first step, the occurrences of predefined general subjective clues are counted. If a sentence contains at least two clues, then the sentence is labelled as subjective. This classifier achieves high precision but low recall.[2] In the next step, the labelled sentences are used as training data for a subsequent learning phase. Here extraction patterns are learned. As third step, a Naive Bayes classifier is constructed. The features of this classifier are defined as counts of (1) the clues, (2) the expressions matched by the extraction patterns of the second step and (3) pronouns, modals, adjectives, cardinal numbers and adverbs. They report improvements in IE performance by dropping subjective information. They expect further benefits for IE by improving subjectivity classifiers.

In this work, only the 'objective' sentences are used for the information extraction task. One can argue that in IE, factual information is needed and that subjective information is of less importance. But the results of the described research provides no evidence for this argument since both the precision and recall are lower by using only the subjectivity filter. The recall and precision is in the end improved by adding some extra tricks (such as selective filtering and statistics). Therefore, based on this research it is not possible to conclude that subjective information is not relevant for information extraction tasks and further research is needed.

More work in the same field is done by Yu and Hatzivassiloglou (2003), who have build a system for answering opinion questions. First they classify documents into two groups, namely facts and opinions, by using a Naive Bayes classifier. They use the type of an article (such as editorial, letter to editor, business and news) for the correct classification label. They also classify on sentence level. To classify opinions from facts, they have developed three different approaches. Those three systems rely on the assumption that each sentence in a document has the same label as the document itself. First they compare sentences and measure the similarity between words, phrases and synonym sets. The second approach uses a Naive Bayes classifier. As features they include words, bigrams, trigrams and parts-of-speech tags. The counts of positive and negative words are also included.

In the last approach to classify sentences, they use multiple Naive Bayes classifiers, each relying on a different subset of the features. They use the sentences correctly labelled by the first classifier as input for the next one. This is repeated until all classifiers are trained.

For the sentences found by the pre mentioned approaches, the polarity is identified. Each sentence is classified of positive, negative of neutral. This is done by measuring the co-occurrence with words from a known seed set of semantically oriented words. As seed words, they used subsets of the manually classified list compound by Hatzivassiloglou and

[2]Precision of objective labels is 82% and 90% for subjective labels. Recall is approximately 37%.

McKeown. Similar results are obtained with the ANEW list of adjectives.

The results of there research are that their document classification is very good, an F-measure of 97% is achieved. Sentence classification is quite harder, and this is visible in the recall and precision values. Detecting opinions has a higher recall and precision (80-90%) than for facts (50%). The best performance is achieved by using the following features: words, bigrams, trigrams, part-of-speech and polarity.

3.6 Evaluation

The goal of this section is to provide an overview of the different evaluation methods for summarization systems. Some methods can be used in both fields, others are specific for one. There are several strategies for evaluation summarization systems, each with another focus. Which strategy the most suitable is, depends on the type of the system. The concepts described below are based on the work of Sparck Jones (2007). The interested reader is referred to that work for an extensive overview.

3.6.1 Text quality

A summary does not have to consist of running text, in some cases a table or list is more suitable. But when a running text is required, it is reasonable to check for 'proper' sentences and 'proper' discourse, in other words: the quality of the text. This can be done by checking specific syntactic properties such as subject-verb agreement. To check global well-formedness is a harder task. For example referents for anaphoric expressions and text cohesion are not so easy to recognize. According to Sparck Jones, text quality is too weak to be a system discriminator.

3.6.2 Concept capture

Does the summary capture the key concepts in the source? That's even a harder question. The first problem is to define the key concepts. This is a matter of human judgement and obviously hard to control. Even for simple versions, humans do not agree. Another strategy to judge summaries, is to use questions which can be answered with the source and should be answerable with the summary. Kolluru and Gotoh in Sparck Jones (2007) argue that this method is robust against human subjectivity. But the problem remains for rich sources for which the range of possible questions is enormous. To deal with this fact, the questions should be according to the purpose of summary.

3.6.3 Gold standard

Another group of strategies has encouraged system summary evaluation against human reference, model or *gold-standard* summaries. Humans know how to summarize and therefore, their summaries can be compared to system generated summaries. Another positive point is that humans can make a summary for a specific purpose which corresponds with the purpose of the summarizer. Although this does not guarantee equal summaries, it is a

step in the right direction. Unfortunately, proper-purpose driven evaluation is difficult and expensive (Sparck Jones, 2007).

3.6.4 ROUGE

An automatic evaluation package for summarization is ROUGE (Lin, 2004). ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation and it is able to automatically determine the quality of a summary by comparing it to other summaries. Those summaries are ideal summaries created by humans. Different versions of ROUGE are available. ROUGE-N measures n-gram recall between a candidate summary and a set of referent summaries. ROUGE-L measures the longest common subsequence between two summaries. It computes the recall, precision and F-statistics on both sentence and summary level. ROUGE-W is an extension of the longest common subsequence measure by adding a weight factor. Another variation is ROUGE-S which measures the overlap of skip-bigrams between a candidate summary and a set of reference summaries. Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. The extended version of ROUGE-S is ROUGE-SU which adds a begin-of-sentence marker at the beginning of candidate and reference sentences. Conducted comprehensive evaluations of the automatic measures showed that particular versions of ROUGE worked well in single document summarization tasks, namely ROUGE-2, ROUGE-L, ROUGE-W and ROUGE-S (Lin, 2004). For very short summaries (or headline-like summaries) the following versions of ROUGE performed great: ROUGE-1, ROUGE-L, ROUGE-W, ROUGE-SU4 and ROUGE-SU9. For multi-document summarization it was quite hard to achieve high correlation (90%), but ROUGE-1, ROUGE-2, ROUGE-S4, ROUGE-S9, ROUGE-SU4 and ROUGE-SU9 worked reasonably well when stopwords were excluded from matching. So concluding, ROUGE is an automatic evaluation package which performs great in comparison with human evaluators.

3.6.5 Baselines and benchmarks

It has become common to set *baselines* for summary performance. As baselines sometimes sentences are selected at random. For news articles the *lead* sentence (or n words) is selected. It is clear that this is not an option for all the domains. Another option is to use a *benchmark* performance set which contains sentences selected by some versions of $tf \times idf$ word weighting.

3.6.6 Purpose evaluation

Sparck Jones (2007) argues that summarizers should be developed on the basis of input, purpose and output factors. Those factors can be used by evaluating a system. It seems to be obvious to evaluate only the output, but Jones argues that both output and purpose factors should be evaluated. Then sentences are evaluated on relevance for the audience, for example. There exist pseudo purpose evaluations including evaluating Newsblaster.

3.6.7 Conclusion

The best way to evaluate summarizers is to combine the strategies above described. This will prevent from being to narrow and evaluate a system from different points of view. The framework suggested by Sparck Jones (2007) gives a good starting point.

Chapter 4

Data and Tools

The goal of this work is to develop a model which summarizes the affect in a given Dutch input set. This choice to develop a system for the Dutch language field, has big consequences. While for English there are plenty of resources available, this is not the case for Dutch. Especially resources which are suitable for sentiment analysis and summarization are very hard to find. For that reason we have decided to collect our own dataset. This dataset is described in the section below. Also the other used datasets are described. After it, tools used for this work are presented.

4.1 Product Reviews

We start this section with a description of our Product Review corpus. This corpus is collected specially for this research.

On the internet, there are several website available for product comparison. Those websites are connected to multiple online product shops and customers can use them to find products and the shops selling those products. In most cases, a list is given with the prices per shop. In addition to this factual information, customers have the opportunity to write a personal note about the products and shops. Others can use this information to make a decision about which product to buy by which shop. The personal notes are indeed written in natural language, but customers express their experience by using a 'star'-system. They can vote with zero to five stars on a limited set of predefined aspects. For example price/quality ratio, quality, graphical performance, and possibilities. Those stars are displayed near the text and are usually used to provide some kind of summary.

4.1.1 Dutch Language

As said before, there are several Dutch product comparison websites. Therefore our expectation was that it should not be a problem to build a corpus with product reviews. But the opposite was the case. Most of the approached organisations were not willing to contribute. Our valuation is that organisations are not able to see the advantages and have limited interest in scientific research. Another problem was to find the person with power to take

decisions and the request was handled by a sales employee which lacks the needed technical background. Organisations were afraid that contribution also implies much effort, but that was actually not the case.

Fortunately, we found one organisation willing to share their data with us, namely Beslist.nl.[1] We are very grateful for their kind cooperation. Beslist.nl has given us access to their online webservices, so that we were able to download the desired reviews. In the next section the results of an exploratorive data analysis are given.

A big advantage of this dataset is that it consists of 'real world' data. The data is not put together in a laboratory situation, but is in the format used by customers. Therefore, the results of this system can be compared with real applications.

4.1.2 Explorative Data Analysis

After downloading all the reviews from Beslist.nl, we have extracted the unique reviews. We have also removed all incomplete reviews, such as missing grades or review text. We end up with a dataset with 50,048 unique reviews, which are written between February 2006 and October 2011. The reviews contain on average 61 words. An histogram of the review lengths is given in figure 4.1.

Concerning the grades, we see that the data is heavy skewed to the higher grades. The mean is 8.03 and the median is 8.0. A plot of the distribution of grades is given in figure 4.2. The skewness of the data has impact on the manner of data handling. For example, a classifier build on this data which doesn't take into account the distribution of the data, can reach a reasonable good overall score by assigning the mean to all reviews. Then, the classifier is doing almost nothing, but is seems a good classifier. Indeed, this is a problem for all classification tasks with skewed data. A solution frequently used, is to use a dataset in which the distribution of classes is equally spread. Unfortunately, this is probably not a good option since the lower grades are so infrequent used that the final dataset will become too small.

It depends on the application what the best solution is for the skewness of the grade distribution. For some applications, it is good enough to reduce the classes to negative, positive and neutral. This is an approach used in many investigations in the field of sentiment analysis. Other solutions include transforming the data or using a correcting factor by classifier evaluation.

Furthermore, evaluation of the classifier can be done with the precision and recall metrics, which are combined in the F-score. A confusion matrix will also give insight into the performance per class. Since this section is not about evaluation but describes the dataset, evaluation metrics are only mentioned and not further discussed.

4.1.3 Corpus Description

A review consists of several fields. In figure 4.3 an overview is given of the datatypes in the corpus. Most fields speak for themselves but some need explanation. As one can see, there are two types in the database, namely review and category. The category corresponds to the

[1] `http://www.beslist.nl`

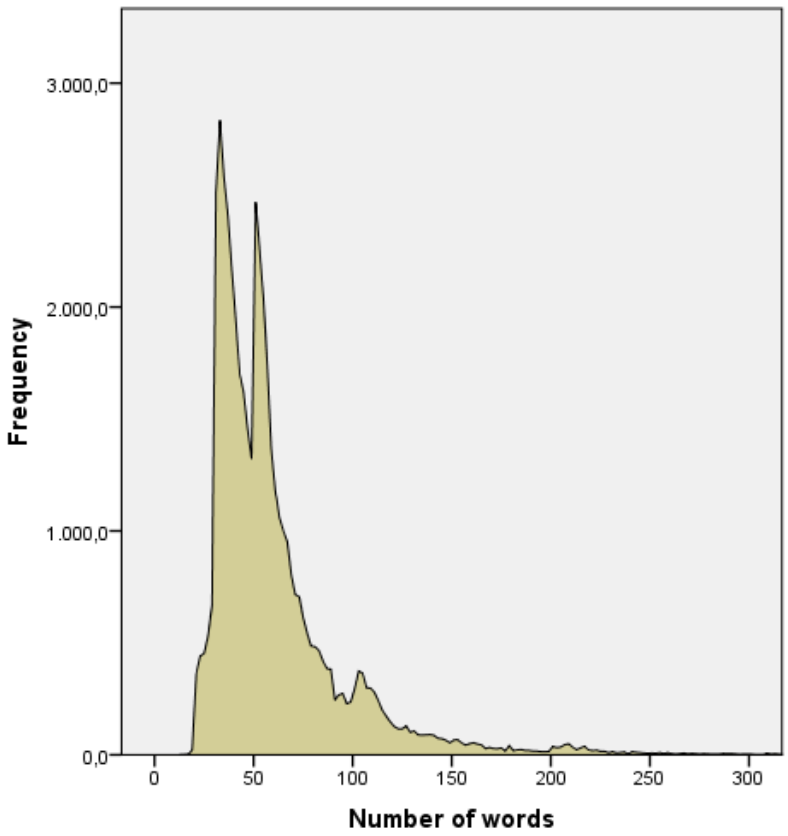


Figure 4.1: Histogram of review length in words

category on Beslist.nl to which the item about which the review is written belongs. In total there are 31 different main categories and 2338 unique sub categories. ItemId indicates about which product the review is written, ReviewId is a unique identifier for a review, Reviewer indicates the author, and Deeplink a url to the review on the internet. Title, Text, Pro and Con are text fields filled in by the author.

Since the focus of this research is on summaries, we have investigated the number of reviews per item. In total, the reviews are written about 18,023 different products. For the greater part, only a few reviews are written per product, but there are products for which many reviews are written. The product with the most reviews contains 234 reviews. In figure 4.4 is it displayed. Horizontal, the number of reviews per product is given, and vertical the number of those sets. The vertical axis is logarithmic scaled. As one can see, there are several products with more than 20 reviews, but the most products have less than 20 reviews.

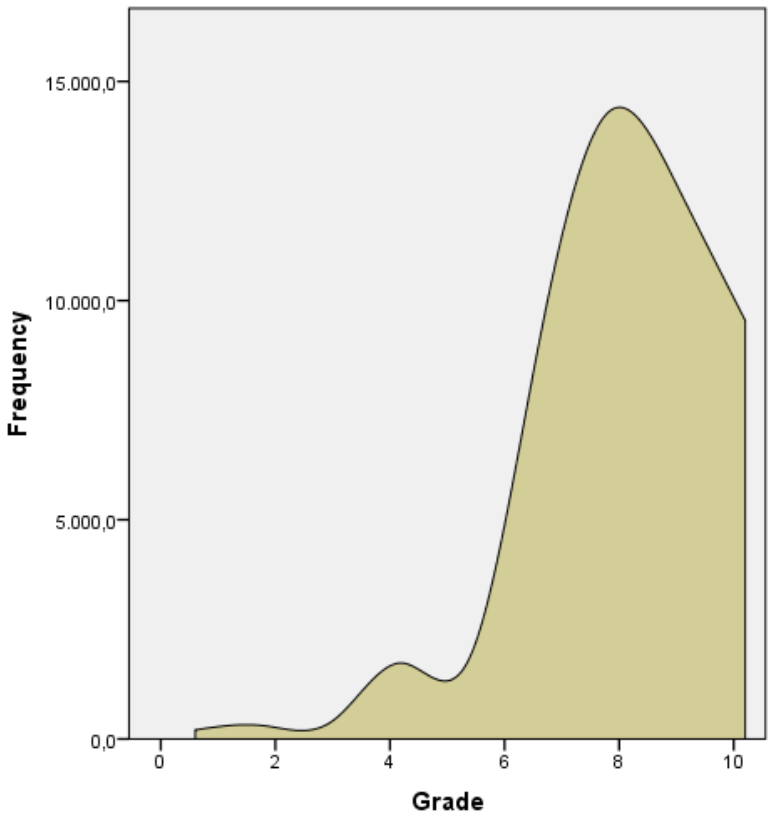


Figure 4.2: Distribution of grades

4.2 Cornetto

Cornetto is a lexical database for Dutch, which covers over 92K lemmas corresponding to 118K word meanings (Hofmann et al., 2008). It includes the most general and central part of the language. The database contains both semantic relations and combinatorial information. Semantic relations can be found in for example wordnets, in which groups of synonyms are connected by mostly vertical connections. For example, *car* ⇒ *vehicle* and *tea* ⇒ *drink*. Besides this, there are also horizontal relations such as roles, part-whole relations and causal relations. Combinatorial information such as lexical functions, selectional restrictions, collocations and syntactic-semantic frames. Those combinatorial constraints are specific for a language and Dutch contains many of this specific rules.

The Cornetto database is aligned with the English WordNet so that the domain ontologies are compatible. The database is created by merging two already existing Dutch lexical databases, namely *Referentie Bestand Nederlands* and *Dutch Wordnet*. Besides this, new concepts and relations are acquired from other corpora by use of acquisition toolkits.

There is an open-source and public database system available. Besides this, the data

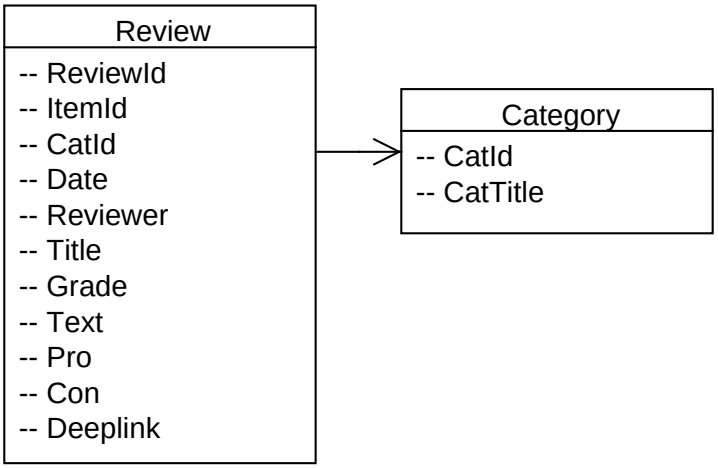


Figure 4.3: Overview of corpus

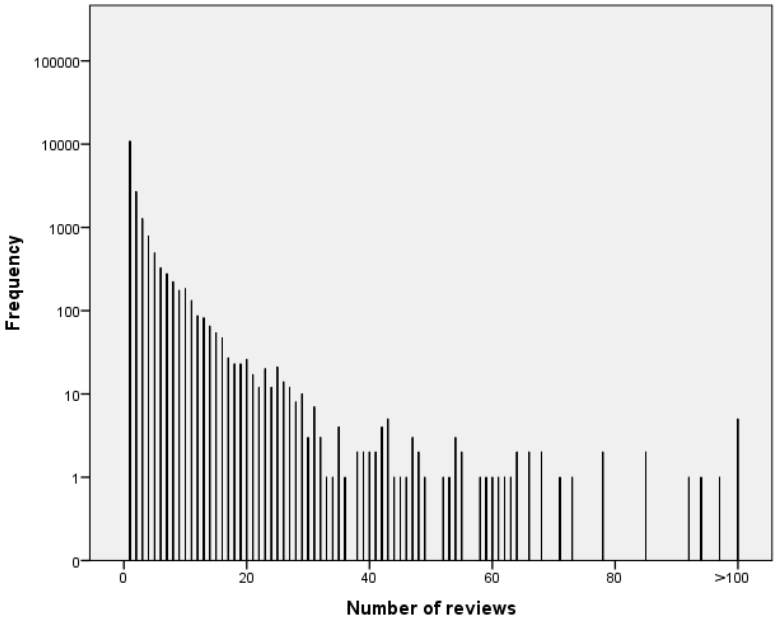


Figure 4.4: Number of reviews per product

can be acquired at the Dutch-Flemish HTL Agency, for Dutch language resources.[2] We have used the source data for the reason of performance issues. A local database is able to handle requests much faster than a online database system.

4.3 Duoman

Duoman is a Dutch subjectivity lexicon which contains a polarity value for all the words it includes. It is based on an English subjectivity lexicon, an online translation tool and a Dutch general purpose thesaurus: Wordnet. The building process and its evaluation is described in Jijkoun and Hofmann (2008). A PageRank-like algorithm is used to bootstrap from the Dutch translation of the English lexicon and rank the words in the Dutch thesaurus by polarity. The polarity runs between -1 and 1, and such a value is assigned to all the 83,125 words in the Dutch thesaurus.

The polarity values for all the words in the thesaurus, Wordnet, are obtained as follows. First an English subjectivity lexicon is automatically translated to Dutch and this lexicon is used as initial seed data for a bootstrapping algorithm. The algorithm considers a wordnet as a graph structure where similar concepts are connected by relations such as synonymy, hyponymy, etc. The initial seed data is used to assign high weights to positive seed examples and low weights to negative seed examples. These weights are then propagated through the graph via the relations in the wordnet. After a specified number of iterations words are ranked according to their weight. Word nodes with the lowest resulting weight are considered to have negative polarity, and word nodes with the highest weight a positive polarity. Those words are ranked and the top of this list are assumed to be positive and words at the bottom of the list are assumed to be negative.

The lexicon generated with the above described method is evaluated based on the manual assessment of 9,089 words (Jijkoun and Hofmann, 2009). The lexicon achieves an accuracy of 0.82 at the top 3,000 negative words, and 0.62 at the top 3,000 positive words.

4.4 TiMBL

A tool used for our investigation is TiMBL which stands for Tilburg Memory-Based Learner. In this section an introduction to this tool is given. First, an extensive overview of the inspirations of Memory-based language processing is given. After it the functionalities of TiMBL are described.

4.4.1 Inspirations of Memory-Based Language Processing

TiMBL is based on the ideas of Memory-based language processing. This is a machine learning technique which uses complete examples instead of extracted rules or features Daelemans and Van den Bosch (2005). This is a different approach then applied in the past, where they first tried to solve the problem of NLP by using knowledge-rich approaches by applying very specific grammars and rules. After it more statistical approaches are tried,

[2] http://www.tst.inl.nl

but the problem with such approaches is that they perform on average well, but fails for individual cases. In other words, language contains more exceptions than rules and therefore for each exception only a few examples exist. This leads to unreliable statistical values. Therefore, the idea of memory-based language processing is that statistical approaches are not so suitable, and its better to use all the examples instead of rules extracted from those examples.

Analogical Reasoning

In contrast to knowledge-rich and statistical approaches, Memory-based language processing tries to use all examples instead of derived rules or estimations. It is based on the idea that the direct re-use of examples using analogical reasoning is a more suited solution for solving language processing problems. This is inspirited by linguistics who stressed the role of analogy and the role of the related dynamic concept of induction. According to De Saussure each language user has the capacity to access some sort of memory: "Any creation must be preceded by an unconscious comparison of the material deposited in the storehouse of language, where productive forms are arranged according to their relations" (De Saussure, 1916, p. 165).

Humans are able to create new sentences based on several sentences they heard before. Parts of those 'old' sentences are used to construct a new one, they never heard before. This lead by Bloomfield to the hypothesis that speakers use 'analogy' and 'habits of substitution' to generate sentences never head before (Bloemfield, 1933, p. 275). He makes a difference between 'regular' non-exact, creative analogical generation from examples, and 'irregular' memory retrieval of examples: "Any form which a speaker can utter only after he has heard it from other speakers, is irregular" (Bloemfield, 1933, p. 275).

Real-world Data

While De Saussure and Bloomfield stress the importance of analogy, John Rupert Firth stressed that having real-world data is central to the development of any model of language. A theory can only be useful and valid if it continually refer to experience, he argues. Therefore, his approach is more data-driven and statistics are important.

Also from a psychological and cognitive linguistic point of view, exemplar-based models are proposed (Smith and Medin, 1981; Nosofsky, 1986). These models assume that people represent categories by storing individual exemplars in memory rather than rules. Categorization decisions are then based on the similarity of stimuli to these stored examples.

Artificial Intelligence

The above described fields come together in Memory-based language processing. The more technical part is inspired by artificial intelligence. The nearest-neighbour classifier methods (most commonly named *k*-NN classifiers) were developed in statistical pattern recognition from the 1950s onwards (Daelemans and Van den Bosch, 2005). They are still actively being investigated in the research community. In these methods examples are represented

as points in an example space with dimensions the numeric features used to describe the examples. A new example is, based on its position in sample space, compared with the k nearest examples in its neighbourhood. The desired class is obtained by extrapolation the classes of the neighbours. Mostly, nearness is defined as the reverse of Euclidean distance.

4.4.2 Functionalities

The last section described which fields inspired Memory-based language processing. Now, an overview is given of the functionality of TiMBL , which can be seen as an application and further consequence of the inspirations.

Decision-tree Structure

In typical NLP learning tasks the focus is on discrete data, very large numbers of examples, and many attributes of differing relevance. In addition, classification speed is a critical issue. The implementation of TiMBL is optimized for those constraints. This resulted into an architecture which compresses the typical flat file organization found in typical k-NN implementations, into a decision-tree structure. While the decision tree can be used to retrieve the exact k-nearest neighbours, it can also be deterministically traversed as in a decision-tree classifier. This makes TiMBL one of the fastest discrete k-NN implementations (Daelemans and Van den Bosch, 2005).

Installation

TiMBL is available as open source software and is written in C++. The installation on a Linux distribution such as Ubuntu is very straightforward. There are installable packages available on the website.[3] After installation a new TiMBL classifier can be trained and tested with the command line. The training data can be inserted by using a text file. The user can supply the data into several different formats, for instance comma-separated values.

Command Line Interface

The data should be well formatted and each line should contain an equal number of features. By using the command line interface, one could specify the type of each feature and also the used algorithm. In the reference guide is described in detail which command line options are available. [4]

Evaluation

The TiMBL package contains several evaluation metrics for determining metrics such as precision, recall, F-score and AUC (Area under the curve). Those evaluation metrics have become common in information retrieval and machine learning in general. See section 5.6 from the Reference Guide for more information.

[3] http://ilk.uvt.nl/timbl-packages
[4] http://ilk.uvt.nl/timbl

TiMBL -Server

A server interface to TiMBL is also available. The server can be used (from the same or other machines) over Telnet. It is also possible to connect a client to the TiMBL -server. The command line interface is the same as TiMBL .

Python Interface

As said above, TiMBL is written in C++. Since many NLP is done in Python, it is good to know that a Python interface exists. In fact it is an interface which uses Boost-python.[5] The python interface can be downloaded via the TiMBL website.

4.5 Frog

We have used Frog, a modular memory-based morphosyntactic tagger, lemmatizer, morphological analyzer and dependency parser for the Dutch language. It is developed by the University of Tilburg, and it is based upon it's predecessor TADPOLE. Using Memory-based learning techniques, Frog tokenizes, tags, lemmatizes, and morphologically segments word tokens in incoming Dutch UTF-8 text files. It assigns a dependency graph to each sentence. The primary aim is being accurate, but the design of the system is also driven by optimizing speed and memory usage.

We have used Frog to find the part-of-speech tags of words. We used a local server implementation.

[5] http://www.boost.org/doc/libs/1_49_0/libs/python/doc/

Chapter 5

Model

In the previous chapters, a theoretic overview is given about the subjects in the domain and the used tools are described. In this chapter, this theoretical background is applied into a model for a sentiment summarization application. The aim of this application is to give an overview of the opinions and sentiment in a given dataset. As described before, our focus is on online product reviews written by customers. So, our dataset exists of written reviews about a product and the user of this application would like to have a quick but reliable overview. The next definitions are used in this thesis.

User By user is meant a customer who is looking for information about a product. The information is used to decide if and which product he will buy.

Application The program in which the designed model is implemented.

Dataset The set of reviews written by customers about one or more products.

In this chapter we will design a model which can be used to build such an application. Therefore, first the goals of this application are described, second, the requirements are given. Further, some general decisions are presented. After it, our model is introduced.

Since our model is divided into three modules, those modules are described are described one by one. The first module is the meant to recognize sentiment, the second module to cluster the sentences with sentiment, and the third module presents the clusters to the user.

5.1 Goals

The model should be designed according to the next goals:

- The user is able to get an overview of the opinions in the given dataset. An overview means that the opinions are combined and condensed, therefore the output is shorter than the input.

- The pre-mentioned overview is reliable. A reliable overview means that the provided results are found in the given dataset and that those results are relevant. In more technical terms, the overview has both a high precision and high recall.
- The results are provided in a for humans easily understandable format. Since the focus of this research is on customers, the results should be understandable without complicated exercises. Therefore, the format of output should be understood within half an hour.

5.2 General Marks

There are always more solutions for a technical problem, and that's also the case for the problem of this research. Therefore, first an alternative solution is presented and then our solution.

In short, the system we build has to recognize sentiment and present it to the user. The recognition task is some kind of data classification task, since we want a class label for each sentence. This class label expresses the sentiment in the sentence. The approaches for data classification tasks can be roughly divided into two types, namely supervised and unsupervised methods. Those naming deal with the data used to build the classifier, supervised means that all the input data contains a label of the desired class. Unsupervised means that those desired class labels are not provided. In general, generating labelled datasets requires much manual effort.

A model which applies supervised learning methods needs classified data. For our case, this means that we need a Review Corpus for which each sentence contains a sentiment classification. This classification can be obtained by manually annotating all the data. For languages such as English, there are annotated corpora available. However, this is not the case for Dutch and we have to generate an annotated corpus by our own. Since our Review Corpus contains more than 50K unique reviews, and more than 700K sentences, it is clear that this will be a very time consuming job. For this reason, supervised learning methods seems not to be an appropriate solution for our case. This doesn't imply that supervised methods are not suitable for sentiment analysis tasks. For other languages, there are multiple systems build by use of supervised learning methods (Pang and Lee, 2008; Wiebe et al., 2006).

Our decision to design a model which makes use of unsupervised learning methods is driven by the data available. The positive point of such approach is that the solution can be applied to other fields without much adjustments. Therefore, our solution will be very general applicable and is not dedicated for the used Review Corpus.

The starting point of this solution is, as has been noted, that we need as little as possible manually annotated data. Therefore, we have to analyse the data with aid of external resources which include the needed knowledge. By use of those resources, the goal is to find information about the opinions in the dataset. There is no need any more that those resources are designed considering sentiment analysis or summarization. In this way, we are able to use general purpose resources. The positive point is that although there are little to no specific Dutch sentiment lexicons available, there are general lexicons for Dutch. For

instance, the Cornetto corpus is a semantically rich lexical database with relations and combinatorial information. This corpus can be used to analyse the semantic overlap between terms.

To apply general resources in appropriate way requires a careful and precise design of the model.

5.3 General Lexicons

With the above-mentioned points in mind, we have designed a model which requires as less as possible manually annotated data and uses general lexicons. This model is graphically represented in figure 5.1. The green arrows indicates a data flow, the deepblue blocks indicates sub modules of the system. The orange cylinders indicates data resources. The thin lines indicates that the modules makes use of the connected data resources.

It is clear that the input of the system is a set of reviews and the output a diagram. The input are raw reviews, since all processing is done within the system. As I have said, the output is a diagram in which a graphical summary is given. The reasons for it are given in the section about presentation.

5.4 Overview

The overview displays that the model is split up into three modules, all with its own task. The first module is designed for the recognition of sentences containing opinions. The system used to recognize those sentences, will be trained as follows. First, the reviews from the Review Corpus are divided into a number of groups. After it, all the reviews are split up into sentences. Each sentence is compared with the different review groups. The sentence is classified with the label of the most resembling group. Those sentences are used to build a classifier. The described training process is iteratively repeated. Once a classifier is constructed, it is used to find the most opinionated sentences of a set of reviews.

The second module is meant for clustering sentences on subject. The input of this module is the set with most opinionated sentences outputted from the first module. Those sentences are grouped on their content. For all the words in the sentences we try to find more abstract concepts. Those abstract concepts and the words are compared based on their relative frequency. The output of this module is a set with clusters and those clusters contain sentences.

The clustered sentences are input of the third module which is used to present the clusters to the user. This presentation is done by visualizing the affect in the clusters. For each cluster, the most relative frequent opinionated words are extracted. They are presented in a graphical way to the user.

In the sections below, the three modules are described in detail.

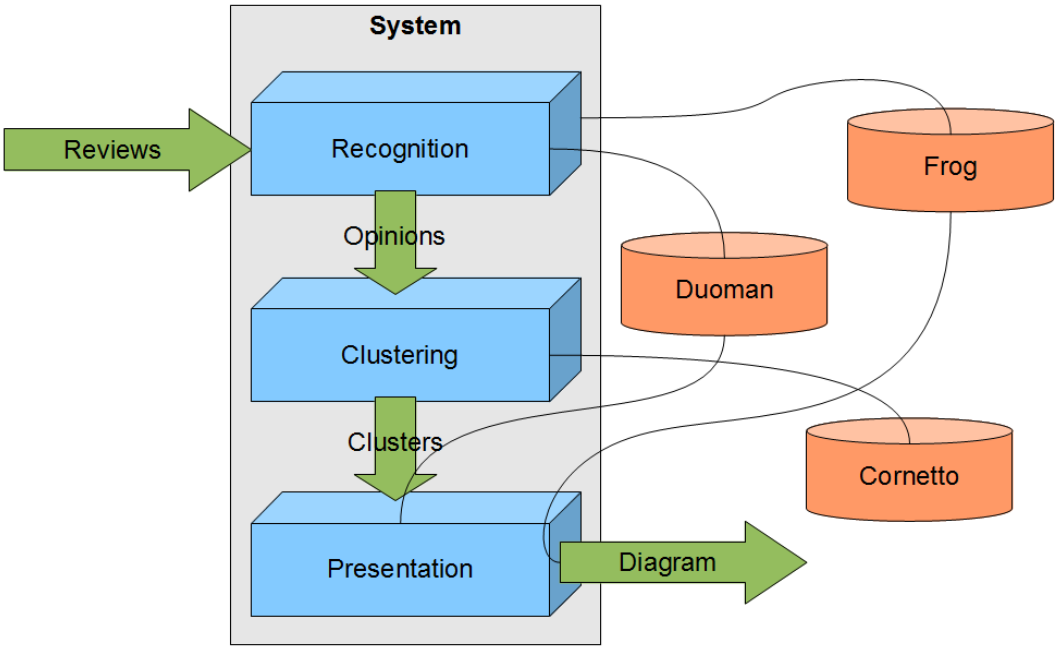


Figure 5.1: Overview of Model

5.5 Recognition

The first module of the system is the Recognition Module. The task of this part is to recognize the opinions in the input dataset and to tag each sentence with the corresponding polarity. A sentence with an opinion has a divergent polarity, we assume. We are interested in divergent polarity because it is plausible that those sentences expresses (striking) opinions. This output can be processed by other systems for further analysis.

As described above, the aim of this model is to use as less as possible manually annotated data. Therefore, we need general purpose data resources. But the question is, how can we recognize sentiment if we don't tell it the system? We don't want to tell the system which sentences contains sentiment, but the model should be able to conclude it on its own. To do so, we need in one or another way knowledge about what sentiment is. This is a very challenging task. We start this task by analysing the data we have. This is described in the next section.

5.5.1 Data Analysis

Our Review Corpus is described in section 4.1 and we have applied an explorative data analysis. Based on those analysis we conclude that the opinions are expressed both textual and numerical. Where textual expression is in the fields title, text, pro and con, the numerical expression is in the field: grade. So, we are interested in those fields. It is obvious that the grade field is a good numerical estimation of the sentiment in a review. A review with a

very low grade contains much negative opinions, and the opposite is also true. So, this field can help us by automatic recognition of sentiment.

Further, it is obvious that there is overlap between the fields since the arguments listed in the fields pro and con go back on the discussion in the text field. It is remarkable that the pro and con fields are more or less lists with key words. The fields don't contain grammatical correct sentences, but mostly only a few words. Another remarkable point is that it isn't always true that the information in the pro and con fields also occurs in the text field. It seems that there is trend that the pro field contains new information when the review is negative. Also, for a general positive review, the con field contains information not covered in the text field.

Finally, if we look to the title field, it can be seen as some kind of summary. In general, titles can be split up into two groups. The first group is characterised by a factual representation and tells only the product name. The other group is characterised by sentimental information and the information in the title contains an advise or sentimental summary. It is clear that the second group contains useful information about the content of the review. The question is: How can we use this information for automatic sentiment recognition?

5.5.2 Preferred Output

For clarity, a visualization of this module is given in figure 5.2. The different steps of the Recognition Module are displayed. The green arrows indicate the training phase of the model, and the yellow arrows indicate the operational mode of the model. Training, which is done once, starts with training data which is put into the Reviews database. After it the reviews are split, sentences are labelled based on those splits and an iterative bootstrapping process starts. This process is meant to build a good classifier.

The preferred output of the model is a set of sentimental sentences, but the above described fields deal with a whole review. Therefore, it is not that easy that we can select the sentences of the review with a high grade, since we cannot make the assumption that all sentences in a review with a high mark are all positive. But there really is a connection between the grade and the sentences in the review. So, the challenge is to induct information on sentence level based on information on review level. This is a knowledge enrichment process.

We have to find a way to utilise the connection between grade and sentences. Our first step to solve this problem is to identify patterns in the used words linked to the ascribed grade. Which words will characterise a high grade, and which terms are specific for a low grade? By answering this question, the common words should be skipped. For instance, a word as *the* is not meaningful.

5.5.3 Approach

There are several possible solutions for this problem. The first solution is to use an adjusted version of the $tf \times idf$ metric. Normally this value is based on one corpus, but we will use three sub corpora. To do so, we divide, based on the grade, the complete dataset into three subsets, namely positive, neutral, and negative. If we would like to know the most

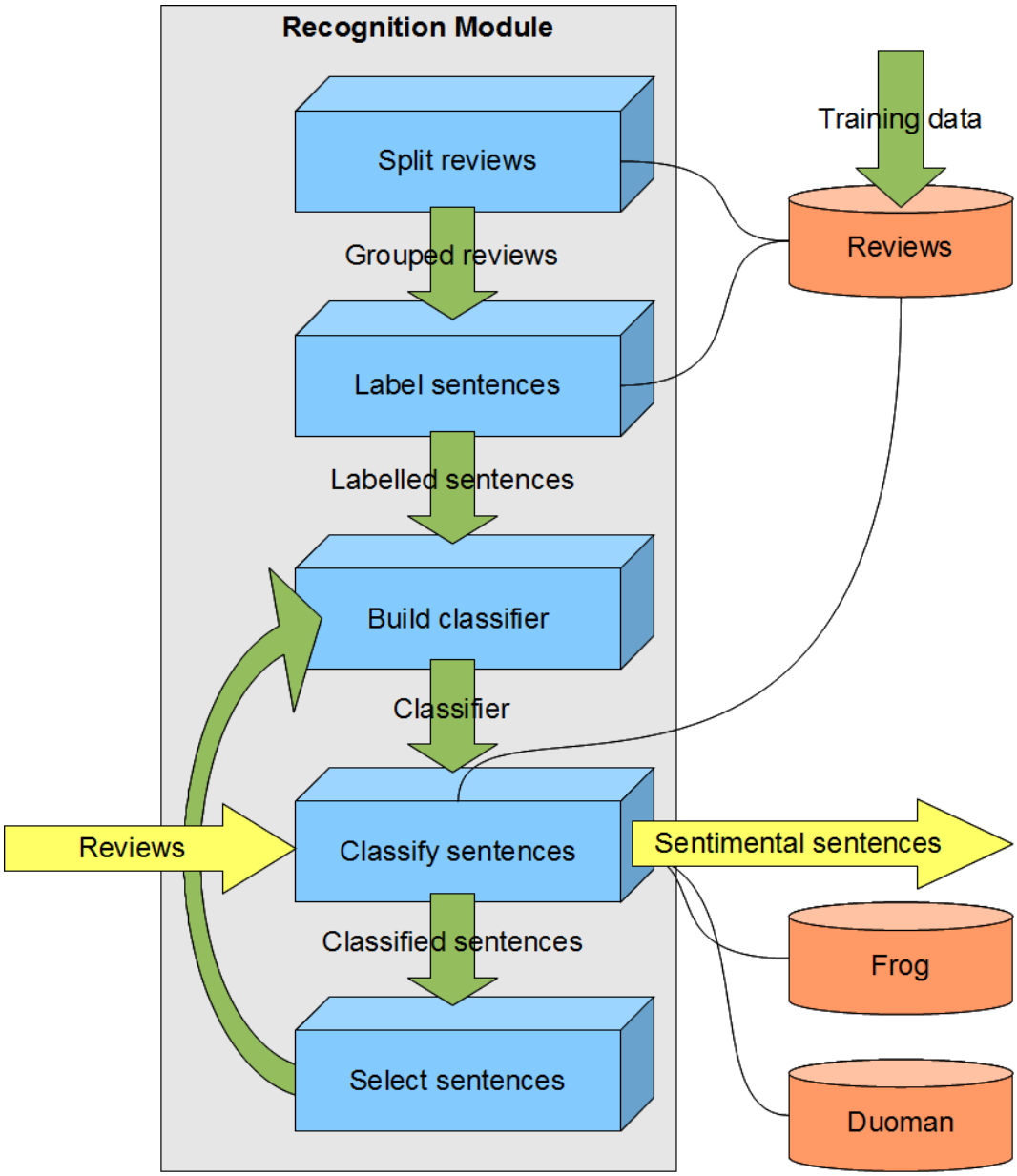


Figure 5.2: Overview of Recognition Module

likely class (positive, neutral or negative) for a certain word, then we compute simply the $tf \times idf$ value for the three sub corpora. The most likely class is determined by the highest value. A certain minimal threshold can be used to reject common words and uncertain cases. This process is displayed in the first two blocks in figure 5.2, namely 'split reviews' and 'label sentences.

If we do this for all the words in a sentence, and select the class with the highest number of occurrences, we are able to classify a sentence by using information on review level. We expect that this procedure works only for the easy sentences, namely the sentences which express pure emotion. Therefore, we use this procedure to select the sentences of which the system is very certain. This set of sentences can be used to analyse the remaining dataset. This is some kind of bootstrapping method.

5.5.4 Bootstrapping

With the above described method, the easiest sentences can be classified, but the goal is to classify all the sentences in the data. Therefore, some extra functionality is needed and we will use the data acquired by the above described method. For the remaining sentences, we will use a machine learning technique, or more precise a pattern recognition technique. A classifier will be used to forecast the labels of the remaining sentences. To build a classifier, labelled training data is needed. For this data we will use the sentences acquired with the above described method. In the next paragraph is described how this dataset will be prepared so that it can be used as training data.

5.5.5 Feature Representation

We should prepare the dataset for use with a classifier. The input of a classifier is always a set of equal length vectors in which each position in a vector represents a property or measurement. Such measurement is named a feature. So, we have to find features for each sentence. Many different features are possible, and the best combination of features is unknown in advance. The best combination can be selected by testing and tuning. In the list below, we present a number of possible features. Some features consist of more than one feature, so it is a feature set. The computation of features is done in the 'classify sentences' block displayed in figure 5.2.

Pure words The most basal feature set we can think of, is to use the words as features. Then, the sentences are compared on word level. Since we use memory-based language processing, this sounds as a good option. Sentences which looks like each other because almost the same words are used, can be found with this feature set.
A problem with this feature set is the varying length of a sentence. To overcome this problem we will cut off long sentences and fill short sentences. A standard length of 25 features per sentence should be enough for almost all sentences. A sentence which contains less then 25 words is filled up with filling symbols. Those filling symbols are recognized by the classifier as an 'empty' symbols. Of course, changing the length is not a problem at all.

An alternative is to use the whole vocabulary as features and to add a number to indicate that a word occurs in the sentence. A consequence of it is that much of the features contain a zero, and only a few are one or higher. This representation is a so called Vector Representation. The problem is that we will got a very long list with features and this could cause performance issues.

Number of negations Since words as 'not', 'no', 'never', and so on, will change the meaning of a sentence completely, a possible feature is to measure the number of those words and add them as a feature. To find those words, we use a list of predefined negations and compare all the words in a sentence with this list.

Number of polarity words Polarity of a sentence is very important for our application. Therefore, the words in a sentence are analysed on their valence. As features we add the number of positive words, negative words, the mean of all the words and both the most negative and positive values.
The values are determined by using the pre-described Duoman lexicon. This lexicon contains a valence value for all the words in the Dutch WordNet. We check each word in a sentence on occurrence in this lexicon. If a word occur, we use the corresponding value. If not, the word is skipped.

Polarity information Information about the polarity of the words in the sentence are added. The polarity of the most positive and negative word in the sentence is computed. An average of the polarity is computed per sentence.

Position A rather easy feature is the position of sentence in the whole review. Since position is important for summary issues we will add this as a feature. The are multiple possibilities, namely the pure number of the sentence, a normalized version between 0 and 1 or the inverted versions of the previous two. For instance, in the last case, a zero corresponds to the end of review.

Grade Each review has a grade. With this feature we add to each sentence the grade of the review.

Binary grade This feature adds an adapted version of the grade, namely a 'N' for neutral and a 'P' for positive. This character is determined on the basis of the grade of the review. All grades above a certain threshold are classified as 'P' and the remaining as 'N'.

Classified grade In the last feature, the set of possible labels consists of two possibilities. For this feature, a third possibility is added, namely the 'O' for neutral. Those features can have a positive effect on the classifier performance since the total number of classes is reduced to two or three. The smaller the number of target classes is, the more accurate becomes the classifier.

The procedure to build a classifier is an iterative process: For each sentence in the set acquired as described above, a number of features are computed and the corresponding label is assigned. Those sentences are used to train a TiMBL classifier. After it, a set with

manually annotated sentences is used to judge the performance of the classifier. If the performance has improved since the last iteration, the best classified sentences are selected as input for the next iteration. This process is visible in the bottom three blue blocks in figure 5.2. The training process is repeated until the results converge.

5.5.6 Classification of Input

In the previous paragraphs, a method is presented to train a classifier. This classification process is done only once, and the resulting classifier is used to classify new input. When new input is inserted to the module, the reviews are split up into sentences. For each sentence, a number of above described features are computed. After it, the sentences with the most extreme values are selected. Those sentences form the output of this module. This is shown in figure 5.2 with the yellow arrows.

5.6 Clustering

The clustering module is meant to automatically build semantic clusters of sentences. Clustering is important because the amount of information could be too large to analyse manually. With clustering, the most important trends can be made visible and the main subjects discovered. This method is in particular useful when analysing a big set with sentences extracted from reviews. The goal is to combine sentences about the same aspect or subject into one group. In the next module, those clusters will be presented to the user.

The input for this module is the output of the Recognition Module and exists of a set with sentences. While the sentences are selected on the basis of affective content, here we are interested in the meaning of sentences.

How can we reason automatically about the meaning of sentences? There are several solutions for this problem. The first one is a shallow approach in which we only take care of the words in the sentence set and identify the words which are relative frequently used. It is an iterative process and in each iteration the most important cluster is found.

In figure 5.3 a schematic overview is given of the clustering module. The different steps of this module are displayed. It is an iterative process and in each iteration a number of computations are applied. Each iteration results in one cluster and the sentences can exist only in one cluster. In the remaining of this section the different steps are explained.

In each iteration the next steps are applied. First, a list is generated with all the words in the sentences. For each word, a $tf \times idf$ value is computed, based on a reference set. This reference set is created by using all the sentences from the text field in the complete Review Corpus. After it, the word with the highest value is selected as current cluster. All the sentences which contain the current cluster word are selected for the first. All the remaining sentences are input for the next iteration. After a predefined number of iterations, the iteration process is stopped and the remaining sentences are clustered as one.

Another more sophisticated method is to include for each word extra semantic relations. This can be done by using a semantic relational database and to search for hypernymy, but it requires that such a database is available. (A hypernym is the opposite of a hyponym and is a word whose semantic field is broader of that of another word, e.g colour is a hypernym

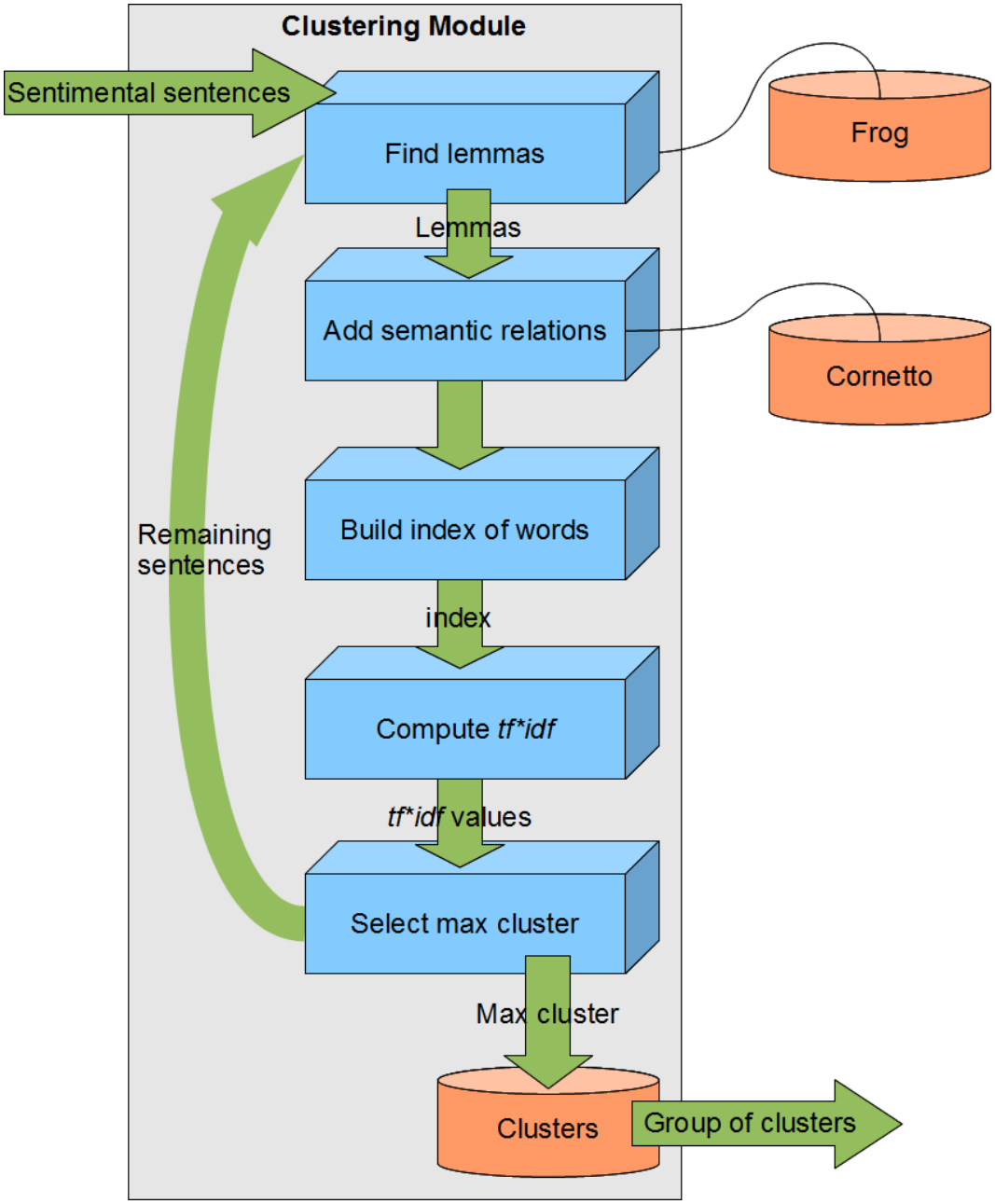


Figure 5.3: Overview of Clustering Module

of red) Fortunately, we have the Cornetto database which can be used for this task. So, pre-described algorithm will change a little bit. In the first step of each iteration, we add also the hypernymy to set with words.

An improvement for both above described methods is to use the lemma instead of the pure word. The lemma could be found by using Frog because it includes a lemmatiser. A more easy approach is to apply a rule based filter to all the words, for instance a snowball filter which removes all the conjugations so that some kind of lemma is returned.

The described method uses the $tf \times idf$ metric, another option is to use a method based on graphs. The idea is to build a graph in which all the nouns, verbs, adjectives and adverbs occurring in the sentence set, are connected. We can use synonyms and hyperoniems to build such a graph. When the graph is build up, the clusters will be identified by using a cluster algorithm for graphs. For instance, the Markov Cluster Algorithm can be used.

5.7 Presentation

Although the three sub modules are equally important for the final result, the third module is most visible for the user. The aim of this module is to present the results in a graphical way to the user. We want some kind of diagram which provides a direct overview of subjectivity per cluster. So, the input for this module is the output of the Cluster Module which consists of grouped sentences. Each group is specified by one abstract term. Under here, a model is presented which create a figure in which each cluster is presented as some kind of cloud. Around each cloud, sentimental words are connected with a line and the thickness of the line indicates how often the word is used in the cluster.

5.7.1 Example

An example will makes clear what is meant with above described idea. We have selected some sentences from our Review Corpus and the reviews where the sentences are took from deals about the Wii-Nintendo game console. So, real data is used and therefore the example is in Dutch. The next sentences are includes in one cluster:

1. De Nintendo WII is een veelzijdige spelconsole.
2. Ik ben echt super tevreden over de wii, het is een geweldig apparaat!
3. Echt geweldig.
4. Echt een aanrader.
5. De WII is in een woord geweldig.
6. Al met al een super console maar net als alle andere duur en voor blijvend plezier zul je er nog meer geld in moeten steken.

The given sentences belongs to the cluster *Conclusie* (Conclusion) and are manually clustered. A graphical sentimental representation of this very small cluster is given in figure

5.4. It is not presented as the only possibility but as an indication for the final result. We see that the cluster name is presented in the middle as a cloud. Around this cloud, some words are presented. We see that the word *geweldig* has a fat line, which indicates that this words is used more frequently than the other words. The color indicates the polarity of the words. A presentation of a set of clusters can contain multiple clouds.

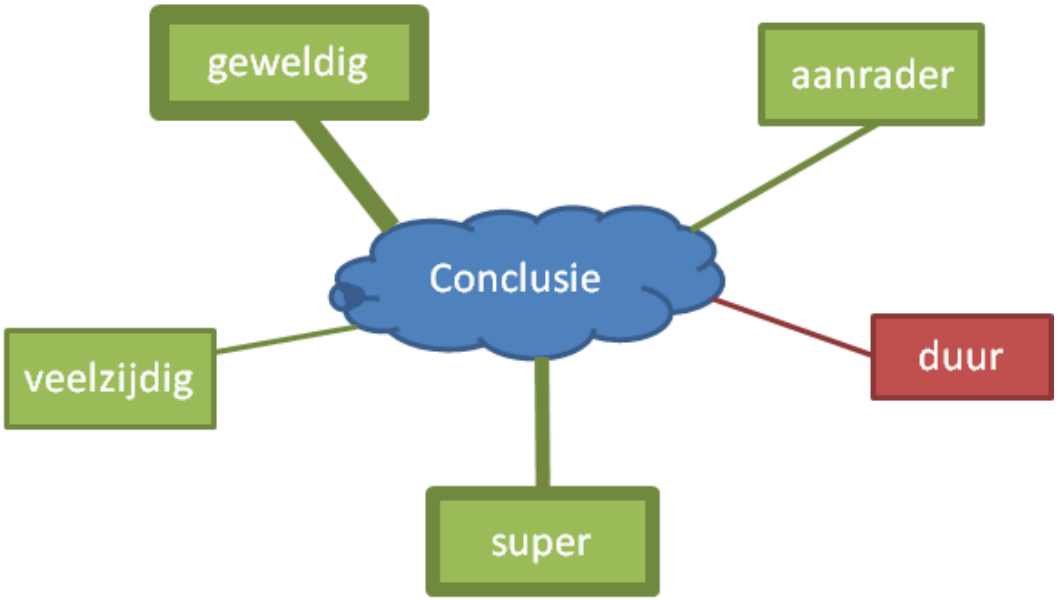


Figure 5.4: Presentation of cluster with main concept 'Conclusie'

5.7.2 Background

The idea as given in figure 5.4 is inspired by a number of observations. First, customers are not willing to read big parts of text, this is visible in the click-through data of review suppliers. Only the first set of reviews will be read and only a few customers will click to the next set. So, we conclude, customers are not willing to read much text.

Second, humans have very robust imagination abilities. It has been demonstrated that numerous benefits can be achieved by applying visual mapping techniques that foster the graphic re-construction of knowledge (Eppler, 2006). Several different types of visualization formats have been suggested, including concept maps, mind maps, conceptual diagrams and visual metaphors. We will discuss those formats one-by-one and use the definitions from Eppler (2006). A concept map is a top-down diagram showing the relationships between concepts, including cross connections among concepts, and their manifestations (examples). It shows systematic relationships among sub-concepts relating to one main concept. This diagram is constructed by starting with the main concept and end with examples at the bottom. It can be used to summarize key topics or clarify the elements and examples of an abstract concept.

A mind map is a multi-coloured and image-centred, radial diagram that represents semantic or other connections between portions of learned material hierarchically. It shows the sub-topics of a domain in a creative and seamless manner and is mostly used for personal note taking. A mind map should be read center-out and constructing starts also in the center. Sub-topics are branched out, pictograms and colours are used to add additional meaning and text is added above the branches.

A conceptual diagram is a systematic depiction of an abstract concept in pre-defined category boxes with specified relationships, typically based on a theory or model. It is used to analyse a topic of situation through a proven analytic framework.

A visual metaphor is a graphic structure that uses the shape and elements of a familiar natural or man-made artefact or of an easily recognizable activity or story to organize content meaningfully and use the associations with the metaphor to convey additional meaning about the content. It is mostly used as text book illustration, summary of presentation to novices.

5.7.3 Knowledge Sharing

It has been showed that the above described mapping techniques provide advantages for knowledge sharing. It depends on the application which method is most suitable, and therefore it is recommended to combine the strong points of the four methods Eppler (2006).

Our question is, can we use (a combination of) those visualizing methods to present a summary? To find an answer on this question, we first tries to identify which are not suitable for our purpose. We would like to create automatically a visualization. Automatic creation of visual metaphors will be too hard nowadays, since much analogical reasoning is needed. There is too much background knowledge assumed and a misunderstood may trigger wrong associations and will lead to a wrong summary. Therefore, we conclude that a visual metaphor is not suitable for automatic summarization.

So, there are three methods left. A conceptual diagram is used to depict an abstract concept with the help of a number of pre-defined categories. It is typically based on a theory of model and the level of difficulty is medium to high. The advantage of this method is that is provides a concise overview, structures each topic into systematic building blocks. Disadvantages are however, that it can be difficult to understand without knowledge of category meanings, and it is not applicable to each topic at hand. Since our presentation should be understood by almost every internet user, it seems that it is not so applicable for our purpose.

So, there are only two methods left: concept maps and mind maps. Can we use the strong points for our situation? The strong points of concept maps are rapid information provision and high understandability by others. However, the level of difficulty is medium to high. The advantages of mind maps are that they have low level of difficulty, and they are easy to extend and add further content. Both methods have points which are desirable for our situation.

The rapid information provision property of concept maps lies in the fact that it makes clear the existing relationships in a consistent way. However, the focus is on the structure of relationships, and colors are not used to express it. The low level of difficulty of mind

maps lies in the fact that the format is not specified into detail. The user is very free to use and apply its own method, such as colors and shapes. But the withdrawal of this is that it is hard for others to understand.

Therefore, if we combine mind maps and concept maps in such a way, so that a consistent and easy to understand visualization arises, we have done a good job. Therefore we will apply the systematic approach of concept maps to mind maps, so that we can use the colouring and radial properties of mind maps in a systematic manner.

5.7.4 Approach

The input of this module is a set of clusters which consists of a set of sentences and a main concept. Our goal is to find the sentimental words are present those around the cloud with the main concept. Our approach is described in the next paragraphs.

First, we extract the adjective words from all the sentences. This is done by parsing all the sentences with the Frog parser. After parsing, all part-of-speech tags are rejected with the exception of the adjective tags. For all adjective words, the lemma is determined again by using Frog.

Second, a index is build of all adjective lemma's. We compare the frequency with the normal frequency and use only words which occur relative frequently.

Third, for each relative frequent words, the polarity is determined. For this, we use the Duoman lexicon to find the corresponding polarity values.

Forth, a graphic is created with the main concepts and the relative frequent words. The lines which connects the relative frequent words with the clouds are coloured based on the polarity determined in the third step. Very negative words are coloured red and very positive words are coloured green.

Fifth, we makes the graphic interactive so that it is possible to click on the relative frequent words. By clicking, a list is shown with all sentences which contain that words. We think that it is essential for a graphical summary that the uses could know which sentences are the bases of each part of the graphic.

5.8 Conclusion

In this chapter, a model is presented which is meant to provide the user an overview of the opinions in a given dataset. To do so, the model is divided into three sub modules, each with its own task. The first module is the Recognition Module. This module selects the most opinionated sentences from a set with reviews. To enable this selection process, a bootstrapping method is applied to train a classifier.

The second module groups the set with opinionated sentences based on its subject. This is done with a semantic database with functionality to select more abstract terms. The output of this module is a set with clusters.

The third module presents the set with clusters to the user. This is done graphically. The subject of a cluster is presented in the middle, and around the subject, subjective keywords are presented.

This model enables the user to get an overview of the opinions in the given dataset. The presented information is directly related to the input reviews. The output of the model is an easy to understand graphic. Those last three sentences match the goals as defined at the first section of this chapter.

Chapter 6

System

In the previous chapter the model is presented. This chapter describes the implementation of this model. The design choices are discussed and the key features are described.

The model is divided into three sub modules, all with its own task. The first module is meant to recognize sentences with a high affective content, the second module groups those sentences into clusters, and the third module presents a visualization of those sentences to the user. An overview of the system is given in figure 6.1.

As mentioned in the introduction of this thesis, the focus of this graduation project is on sentiment analysis and summarization. Sentiment analysis is mainly done in the first module of this system. That module recognizes the sentiment in the sentences. The sentiment in the sentences is compared with the sentiment in other sentences. This analysing process selects the sentences with above-averaged polarity. The first part of the focus is implemented in this module.

The summarization process is mainly done in the third module. This module selects affective keywords for each cluster with sentences. The keywords are analysed and the polarity is determined. The keywords which occur more often in the given cluster in comparison to other clusters are selected for the visualization. Those keywords are used to build a graphical summary which is presented to the user.

Since the focus of this project is mainly realized in the first and third module, we have decided to skip the implementation of the second module. This decision is also driven by time limits. The second module is a Natural Language Processing task and has nothing to do with sentiment analysis. This requires much extra effort which go beyond the focus of this project. We have tried to find an usable cluster implementation, but we were not able to find it.

The chapter is structured as follows. In section 6.1 the technical requirements to run our system are presented. In the third section, the implementation of the recognition module is described. Finally, the presentation module is discussed.

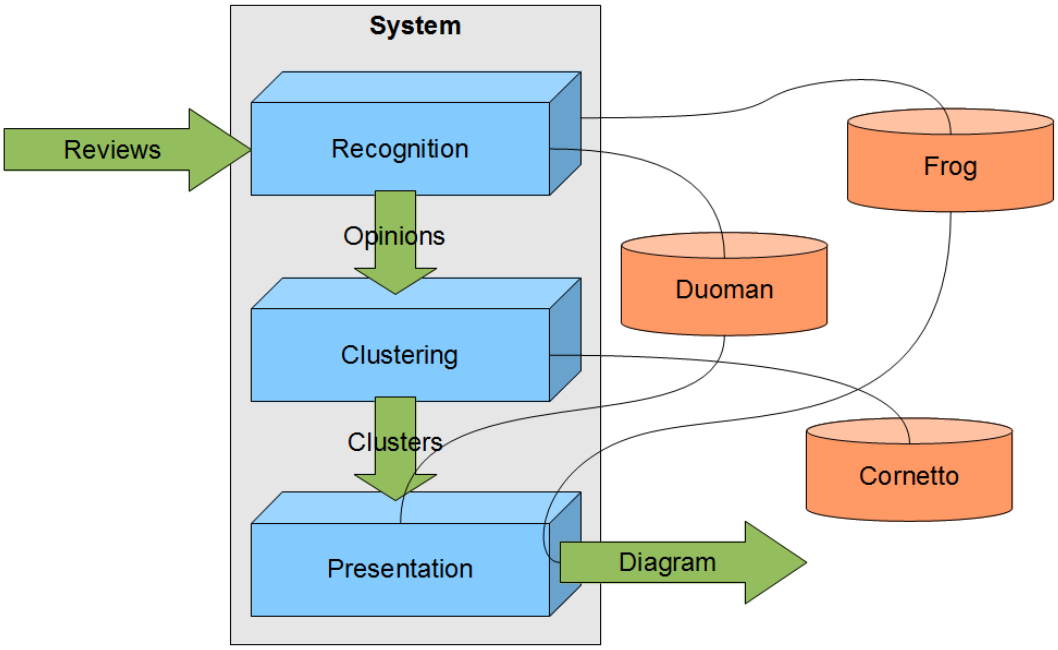


Figure 6.1: Overview of System

6.1 Technical Requirements

This section describes the technical requirements needed to run our software. The choices are explained.

6.1.1 Programming Language

For the implementation we have used the programming language Python. This language is used for many Natural Language Processing problems, and therefore we can reuse functionality of existing toolkits. For instance, the Natural Language Toolkit (NLTK) contains much basic functionality such as text splitting and standard data structures (Loper and Bird, 2002).[1] Furthermore, Python can be used for both scripting and Object Oriented programming. This makes it very useful for a broad spectrum of applications and specially for our case because small independent sub tasks can be done with a script. For example, we have generated datasets by using a single script. On the other hand, the biggest part of the system is object oriented and hence, the code is easily reusable.

Besides those mentioned pros, one can argue that a language like C++ gives programs that run faster than Python. Probably, this will be indeed the case, and when the model will be implemented for a real world application another language can be a better choice. But this system is only meant as scientific research and for this research, performance issues are not that important. For our case, it does not matter if running the program takes a

[1] http://www.nltk.org

few minutes more or less. Our expectation is that the time needed for implementation will increase too much if a language like C++ is used.

6.1.2 Operating System

The system is developed on a computer with the Linux distribution Ubuntu 11. We have taken this decision since some of the used tools are developed for Linux. The TiMBL package, for example, needs Linux. Python programming is also more easy in Linux, since there exists multiple free open source programming environments.

6.1.3 Software Needed

To run our system, some software packages are needed. First, a running Frog server is needed. (See section 4.5.) This server should run on port 2222 on the localhost. Second, TiMBL software should be installed. For both Frog and TiMBL , packages for Ubuntu exist. The third requirement is a Python binding for TiMBL . This software can be downloaded from `http://ilk.uvt.nl/~sander/software/python-timbl.html`. Unfortunately, this software is not direct usable, a small manual change is needed. For all functions in timbl.h and timbl.cc, the namespace 'TIMBL' should be added. After it, the software can be compiled and TiMBL functionality should be available in Python.

6.2 Recognition Module

The implementation and design choices for the Recognition Module are presented in this section. In figure 6.2 an overview of this module is given. Each lightblue block in this overview will be discussed separately in a subsection. For some main functionalities, blocks with code are given. In appendix A, class diagrams are given for the first and third module. In the subsection below, references to classes are indicated like *class*.

6.2.1 Group Reviews

The first step of the Recognition Module is to group the reviews based on their grade. This is done with the code in code 6.1. The number of output groups is determined by the length of the input parameter splitvalues. A review can be assigned to multiple groups.

The parameter 'splitvalues' is important for the final performance and should be selected carefully. Also, it determines how many output labels the classifier can ascribe. It depends on the used dataset what the best parameters are. For the Review Corpus, several different settings has been investigated. It turns out that for three output labels, overlapping split values provide the best results. Overlapping of half a point gives the best results.

```
1 ''' This function splits the given reviews dataset into a number
2     of sub sets, based on the values given in splitvalues. The grade
3     field of the reviews is used for this split.
4     Splitvalues should be format as follows: [[min max] [min max] ... ]
5     The number of groups is determined by the number of SplitValues
          values.
```

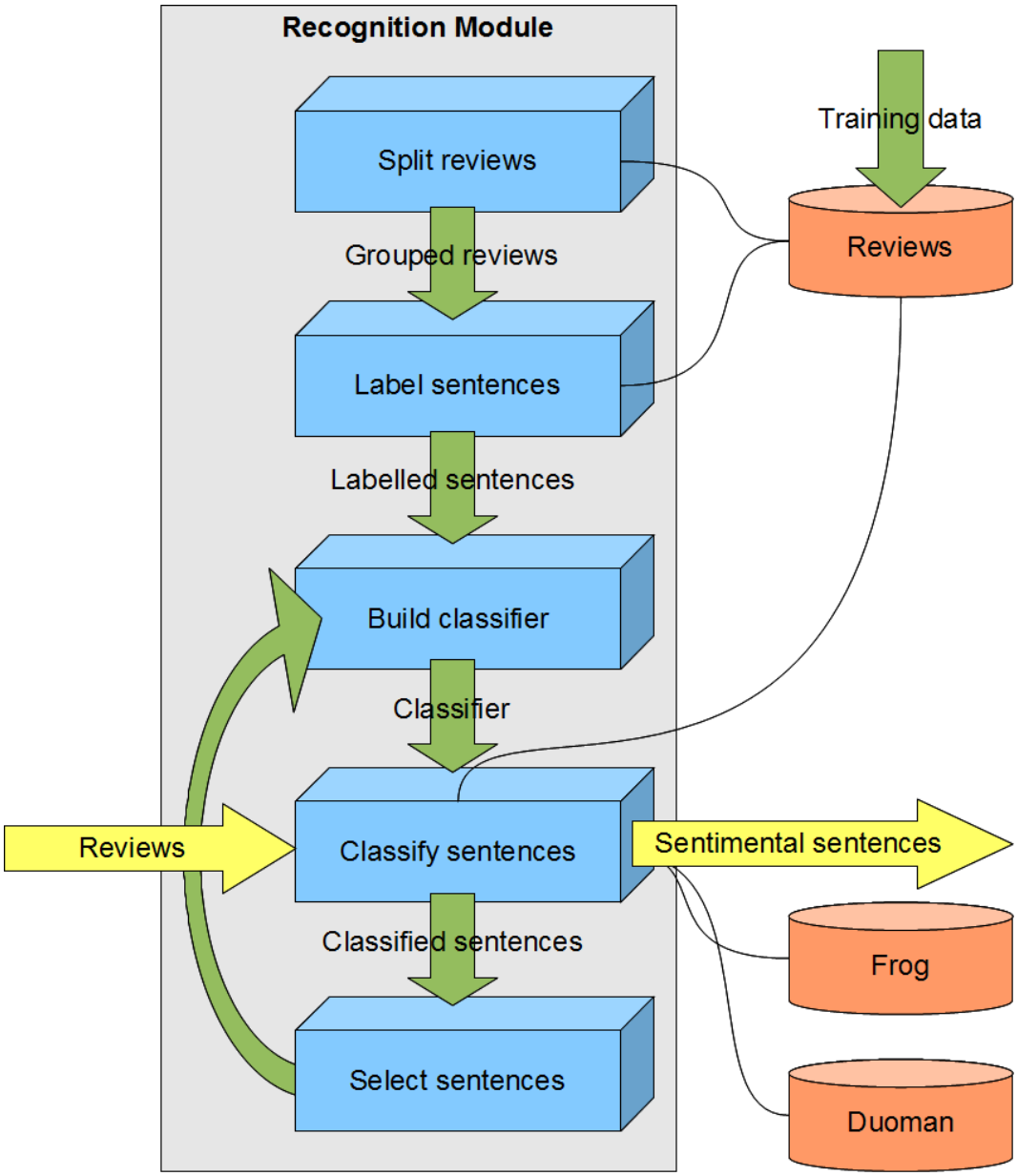


Figure 6.2: An overview of the Recognition Module

```
'''
def split_on_grade(self, reviews, splitvalues):
    # Create empty result list with length equal to length splitvalues
    result_sets = [[] for row in range(len(splitvalues))]

    # Add reviews to the right class
    for rev in reviews:
        my_grade = rev.getGrade()
        set_id = 0;
        # iterate over all splitvalue pairs
        for (minVal, maxVal) in splitvalues:
            # if found, add review to result set.
            if ((my_grade > minVal) & (my_grade < maxVal)):
                result_sets[set_id].append(rev)
            set_id+= 1
    return result_set
```

Code 6.1: Code used to group reviews, based on their grade

6.2.2 Label Sentences

The next step of the system is to label all the sentences in the Review Corpus. To that end, the inverse document frequencies (which are part of $tf \times idf$) are computed. This is done for all the groups obtained in previous step of this module. In our system, one document is equal to one single sentence.

To compute a $tf \times idf$ value for each sentence, the text of reviews should be split into single sentences. This is done with the code in code 6.2. We have used the sent_tokenize function from NLTK. Each sentence is wrapped into a *LineInstance* object and those *LineInstance* objects are appended to a *InternReview* object. This is done in the wrap_reviews function of *DatasetUtils*

```
''' This function wraps the reviews into InternReviews. All those
    intern reviews contain one LineInstance per sentence.
'''
def wrap_reviews(self, reviews):
    result_ireviews = []
    for rev in reviews:
        my_irev = InternReview(rev)
        # Split text into sentences
        sentences = sent_tokenize(rev.getText())
        for sen in sentences:
            # Split sentence into words
            words = word_tokenize(sen)
            my_irev.add_instance(LineInstance(words))
        result_ireviews.append(my_irev)
    return result_ireviews
```

Code 6.2: Split text of a review into sentences

Once the reviews are split into sentences, for each sentence a label is obtained by computing a $tf \times idf$ value for each group constructed previously. The label of the group for

which the $tf \times idf$ value is maximum and exceed a defined threshold is assigned to the sentence. This is visible in code 6.3 which is a function of DatasetUtils.

Based on tests, we have concluded that the $tf \times idf$ is not a very good indication for the best group. It turns out that the use of only the frequency is a better indication. For example, the word 'cool' has a high $tf \times idf$ value for the negative classes, but it is word which is an indication for a positive sentence. If we use the frequency, it turns out that the frequency of 'cool' is highest for the positive groups with reviews. For that reason, we have change the _compute_tf_idf function so that it returns the frequency in place of the $tf \times idf$.

```
''' Label all sentences in the set with internReviews. It computes
    a tf_idf value for each sentence and determines on the basis of
    the given tf_idfs (multiple possible) the most appropriate class.
    This class is returned as the index of tf_idfs.
    The threshold is used as to select or reject a sentence.
    A new list with i_reviews is returned, and those internReviews
    contain only the LineInstances of which the threshold is high enough.
    The labels of those LineInstances are set.
'''
def label_sentences(self, i_reviews, tf_idfs, labels,threshold=0.5):
    result = [] # List with InternReviews
    for i_rev in i_reviews:
        found = False
        new_rev = InternReview(i_rev.get_review()
        for li in i_rev:
            words = li.get_words()
            label = ""
            max_value = -9999.999
            for (i, tf_idf) in enumerate(tf_idfs):
                value = self._compute_tf_idf(tf_idf, words)
                if value > max_value:
                    label = labels[i];
                    max_value = value
            # if the found value is higher than the given threshold
            # set the label of the sentence to the label of the
            # group and add the sentence to the interReview.
            if max_value > threshold:
                found = True
                li.set_label(label)
                new_rev.add_instance(li)
        if found:
            result.append(new_rev)
    return result
```

Code 6.3: Assign a label to each sentence

The threshold parameter is used to select the sentences which appear frequent enough, since we are looking for sentences specific for a group with reviews. The range of this parameter is between 0 and 1. If the parameter is one, all sentences are selected. If the parameter is 0, no sentences are selected. Based on evaluation, we conclude that the best threshold depends on the number of output labels. The more output labels, the lower the threshold.

6.2.3 Build Classifier

Once the sentences contain a label, they can be used to build a TiMBL classifier. The labelled sentences could come from either the 'label sentences' phase or the 'select sentences' phase. (Which will be discussed later on.)

In the 'build classifier' phase, a file is constructed which is used to start a TiMBL instance. The file used to train TiMBL contains per line an instance with the features. In figure 6.3 a few lines are given of a possible train file. The different features are separated by a comma, and all lines contain an even amount of features. The last item on a line is the desired label.

```
20 erg, heel, prijzig, =, =, 1, -0.004, 2, -0.210, 2, 0.176, 0, None,n
21 echt, waardeloos, =, =, =, 2, 0.002, 2, -0.206, 3, 0.212, 0, None,o
22 heel, ontevreden, =, =, =, 3, -0.064, 2, -0.203, 1, 0.017, 0, None,o
23 goed, =, =, =, =, 4, 0.070, 1, -0.003, 2, 0.206, 0, None,p
24 persoonlijk, goed, echt, link, =, 5, 0.049, 5, -0.203, 4, 0.229, 0, None,p
25 nijden, sal, slecht, =, =, 0, -0.032, 7, -0.223, 4, 0.239, 3, None,o
26 laatst, volledig, =, =, =, 0, -0.026, 7, -0.209, 1, 0.007, 1, None,p
```

Figure 6.3: A few lines of a TiMBL training file

We have written a function which enables us to compute a number of features for a set of *InternReviews*. It iterates over a given list with *InterReviews* and for each *LineInstance* in those *InternReviews*, a number of features are computed. The implementation of the *Features* is described in the 'classify sentences' phase. The process to generate a TiMBL training file is displayed in code 6.4. This function can also be used to generate a test file, the only difference is that in that case no label is added. The function is part of the *TimblUtils* class.

```
1  ''' This function can be used to create a file which can be used as input
2      to timbl.
3  '''
4  def build_timbl_file(self, iReviews, featureSet, filePath, add_labels):
5      with open(filePath, 'w') as f:
6          for iRev in iReviews:
7              # Compute all the features for this InternReview
8              for feat in featureSet:
9                  feat(iRev)
10             # Write for each LineInstance in iRev a line to the output
                   file.
11             for li in iRev:
12                 s = ""
13                 if add_labels:
14                     s = ", ".join(li.get_features()) + ",%s\n"
                           %li.get_label()
15                 else:
16                     s = ", ".join(li.get_features()) + "\n"
17                 f.write(s)
```

Code 6.4: Function to generate a TiMBL training file

6.2.4 Classify sentences

A TiMBL classifier is trained with the file generated in the previous phase. In 'classify sentences' phase, this classifier is used to find an appropriate label for input sentences. To do so, the same features are computed as in the previous phase. In the section below, the implementation of the features is discussed. For each features is described how it is computed and which metric is used to find the distances between sentences. The distance between sentences is used to find similarity between sentences. For numerical features, it is easy to compute the distance, namely as the sum of the differences between feature value pairs. This is indicated with a 'N' But for text based features, it is quite harder. For that reason, in TiMBL several options are possible.

The most used option is to compute the distance between sentences with the Modified Value Difference Metric. This metric computes the distance in the phonetic domain. It can use the information that 'b' and 'p' are more similar than 'a' and 'b'. The computation is based on co-occurrences of feature values with the target class. To compute the difference between two values v_1 and v_2 of a feature, they compute the difference of the conditional distribution of the classes C_i for these values.

$$\delta(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)|$$

In table 6.1 is presented how features are computed and which metric is used for that feature.

6.3 Presentation Module

The task of the presentation module is to create a visualization for clusters of sentences.

The input of the presentation module consists of a Comma Separated Value (CSV) file. A few lines of an example are given in figure 6.4. The field contains a identifier to the sentence id. The second field is a identifier for the cluster. The last field contains the text of the sentence.

The information in the file is converted to an internal representation. After it, for each sentence the subjective keywords are determined. We have followed the next procedure to determine subjective keywords.

The first step of this process is to find all the part-of-speech tags of the complete Review Corpus. This is done using of a Frog server. A frog client is build to create a socket connection to this server. This client has a function 'process' to passes a sentence to the server and to return the resulted tags.

We have tagged all the reviews from the Review Corpus, which consists of more than 700,000 sentences. This tagging procedure has taken many hours. After tagging, all the words with the tag 'adjective' are selected. Those words are used to build a $tf \times idf$ base. With this base, we can compute a $tf \times idf$ value for new sentences.

The second step is to determine the part-of-speech tags for each input cluster. This in done in the same way as the complete corpus. For all the adjective words in each cluster, the $tf \times idf$ value is computed by use of the pre mentioned base. This results in a list with

Feature name	Computation	Nr feat.	Metric
Pure words	Simply add each word as a feature, and use the '=' symbol if the number of words is smaller than the number of features	25	MVDM
Number of negations	Check each word in the sentence for occurence in a list with negation words. Add the number of occurences.	1	N
Number of polarity words	Check each word in the sentence with Duoman, and check if it is positive of negative. Add the number of positive words and the number of negative words as features.	2	N
Polarity information	Check each word with Duoman and add the mean, minimal polarity value and maximal polarity value as features.	3	N
Position	Add the number of the line as a feature	1	N
Binary grade	Add a 'N' if the grade is smaller than 5.5 or a 'P' if greater than 5.5.		MVDM
Classified grade	Add a 'N' as the review grade is smaller than 4.33, a 'O' if the feaure between 4.33 and 7.67 or a 'P' if the grade is higher than 7.67.	1	MVDM
Grade	Add for each sentence, the grade of the review as a feature	1	N

Table 6.1: This table presents how the features are computed.

(word, frequency) pairs. The word is the adjective word, the frequency is the $tf \times idf$ value of the corresponding word.

The third step is to find the polarity value for all those adjective words. This is done by use of the Duoman corpus. For this task the code in code 6.3 is used. It returns a list with the word, the frequency and the corresponding affect.

```
''' Check the given list with (word, frequency) pairs for existence
in Duoman Lexicon. Returns a new list with (word, frequency, affect)
    tuples.'''
def _check_with_duoman(self, tuples):
    results = []
    for (word, frequency) in tuples:
        try:
            affect = self.duoman.classify_word(word)
        except WordNotFoundError:
            affect = 0
        results.append([word, frequency, affect])
    return results
```

The next step is to analyse the found frequencies and polarity values. Only adjective words which occur frequent enough are selected to present in the final graphic. We have

```
44 840;2;Al spelend kun je jezelf een hele workout geven.
45 372;2;Het is een multifunctioneel apparaat.
46 80;2;De Nintendo WII is een veelzijdig spelconsule.
47 666;2;Makkelijk in de bediening, neemt weinig ruimte in en de mogelijkheden nemen nog steeds toe.
48 28;2;Een echte aanrader.
49 373;2;Echt geweldig.
50 2;2;Echt een aanrader.
51 623;2;Ik ben echt super tevreden over de wii, het is een geweldig apparaat!
52 569;3;Wii is voor wat betreft het gamen en bewegingen compleet nieuw en daarnaast erg aantrekkelij
53 149;3;De Nintendo Wii is een leuke console die vooral bedoeld is om met vrienden of familie speler
54 912;3;Waar ik normaal gesproken helemaal niet zo'n fan ben van games geeft het spelen met vrienden
55 138;3;Vanwege de vele mogelijkheden eindelijk een console waar we met de hele familie wat aan hebb
56 724;3;Het is een fantastisch familie spel.
57 531;3;"Ze spelen met en tegen elkaar; soms zijn vriendjes de tegenstander."
58 643;3;ik heb samen met mij dochter (4jaar) veel plezier, zeker met de mat die we erbij hebben gekc
59 574;3;Leuk voor volwassenen maar ook voor kleine kinderen.
60 893;3;Een systeem leuk voor man , vrouw en kinderen
61 382;3;Ik heb geen kinderen, maar tijdens feestjes en bij bezoek wordt hij meestal wel tevoorschijn
62 255;3;Deze spelconsole is leuk voor alle leeftijden
```

Figure 6.4: This screenshot of a file shows a few lines of a input file for the Presentation Module.

chosen a threshold of 0.15, but this threshold could be adjusted by the customer when he is analysing a set with reviews. But for now, this parameter is set to 0.15, and this is done based on the number of words selected for the presentation. By using this parameter, the final graphics contain maximal 11 adjective words. By adding more words, the customer will loose the overview.

The frequency is used for the thickness of the lines around the words. The most frequent words gets the most thick line.

The background color of the adjective words is determined by the affective value. A positive value causes a green background, a negative value a red background.

A graph create with the information gathered with the above described process is given in figure 6.5.

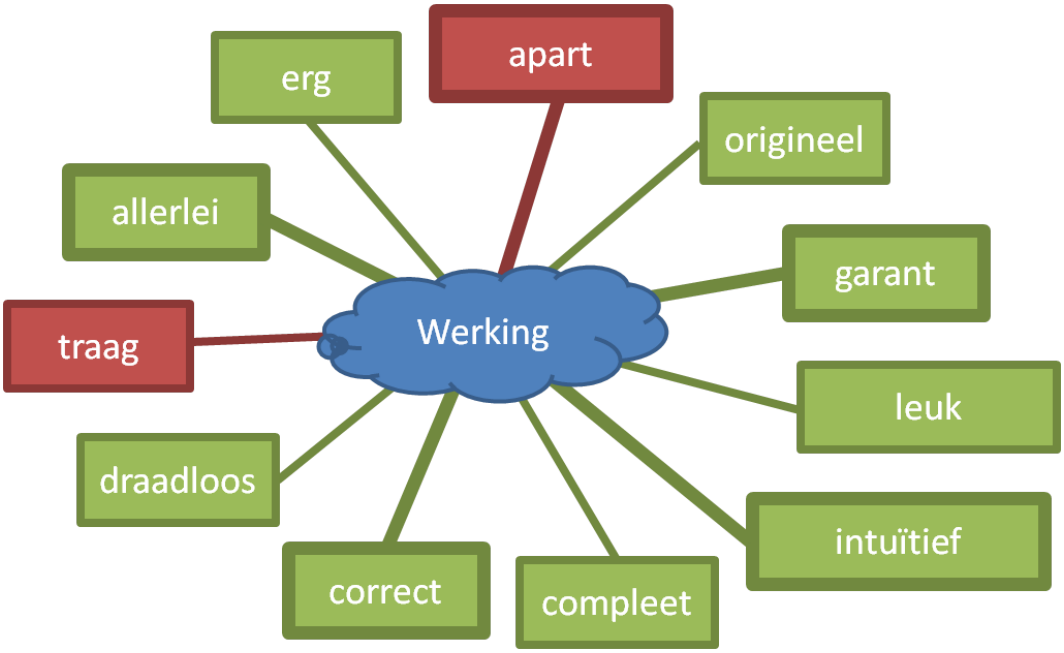


Figure 6.5: This is a cluster generated with the Presentation Module.

Chapter 7

Evaluation

In the previous chapter, the implementation of the Recognition Module and Presentation Module is described. In this chapter those modules are evaluated. The Recognition Module is evaluated using a Gold Standard test set. The Presentation Module is evaluated with an user experiment.

In the next section the evaluation of the Recognition Module is described. First, the used method is given. Next, the used software tool is presented. After it, an evaluation of the Gold Standard test set is given. In subsection 7.1.4 is described how the Gold Standard is used to evaluate the Recognition Module. After it, some feature settings are given.

The evaluation of the Presentation Module starts in section 7.2. After it, both the agreement between raters and the results are discussed. This chapter ends with a conclusion.

7.1 Recognition Module

The evaluation of opinion extraction systems is mostly done by using an evaluation corpus. Our recognition module can be seen as such an opinion extraction system since its goal is to find the most opinionated sentences. Therefore, we decided to evaluate our system with an evaluation corpus. Unfortunately, there wasn't such a corpus available for Dutch and therefore we have decided to create our own. First, the method used to create such a corpus is described and after it the used software tool is presented. After it, the reliability of the results are discussed. Finally, the used parameters of the recognition module are optimized and the resulting system is tested with the created test corpus.

7.1.1 Method

For the development of our test corpus, we have used the method suggested by Ku et al. (2007) and referred by Pang and Lee (2008). They have introduced a method for developing opinion corpora involving multiple annotators. First, a corpus is annotated by multiple human annotators, and second, a gold standard is generated based on the agreement of annotators. There are two metrics used to determine the agreement, namely both a strict and lenient method. The strict metric is different from the lenient metric in the agreement of annotations. For the strict metric, sentences with annotations agreed by all three annotators

are selected as the testing collection; for the lenient metric, the sentences for which at least two annotators agreed are selected.

If we want to judge our systems at the level of human beings, Ku et al. have suggested to enhance systems with strict, high agreed, and substantial consistent testing collections.

We have selected randomly 750 sentences from the Review Corpus. Those sentences are annotated by two annotators and the sentences for which the two annotators agreed are selected as gold standard, so we have used the strict metric. Annotating is done using a specially developed annotating tool. This tool is meant to enable annotators to annotate as quick as possible a big set of sentences. The tool is described in the next subsection. The annotation guideline which is provided to the annotators is included in appendix B.

7.1.2 Annotation Tool

The annotation tool is small Java application which can read a text file from disk. This text file should have the right format, namely two fields per line and the fields should be separated by a semicolon. The first field is an identifier, compound of the review identifier and the sentence number. The second field contains the text of a sentence. When the file is selected by the user, the lines are presented one by one to the user. As one can see in figure 7.1, the progress is shown at the top, below, the sentence is displayed and at the bottom a button is presented. On the slider above the button, the annotator can indicate the polarity of the sentence. Left indicates very negative, right indicates very positive. The middle is used to indicate a neutral sentence. On the slider, there are marking points to indicate the edges between positive and neutral, and neutral and negative. Therefore, the rater was able to indicate very precise the polarity of the sentence. The use of the slider is motivated by its easiness of use. The position of the slider is converted to an integer value between 0 and 100. If the annotator has indicated the polarity, the result is written to a new text file.

This easy to use tool makes it possible for almost everyone to annotate sentences in a graphical way. Very less to none training is needed and the tool is platform independent.

7.1.3 Evaluation of test collection

In total, three annotators have annotated (a part of) the test collection by use of the presented tool. Although we use only the sentences for which the two annotators agree, it is good to have insight into the evaluations. We will use Kappa values to assess the annotated test corpus (Fleiss et al., 1969). Kappa value gives a quantitative measure of the magnitude of inter-annotator agreement. The formula to compute Kappa is:

$$\kappa = \frac{f_0 - f_c}{N - f_c}$$

where f_c) is the number observed agreement and f_c represents the number of items that would be expected to be coded the same way by chance alone. N is the number of items coded by each rater.

Table 7.1 shows a commonly used scale of the Kappa values. This table originates in Landis and Koch (1977).

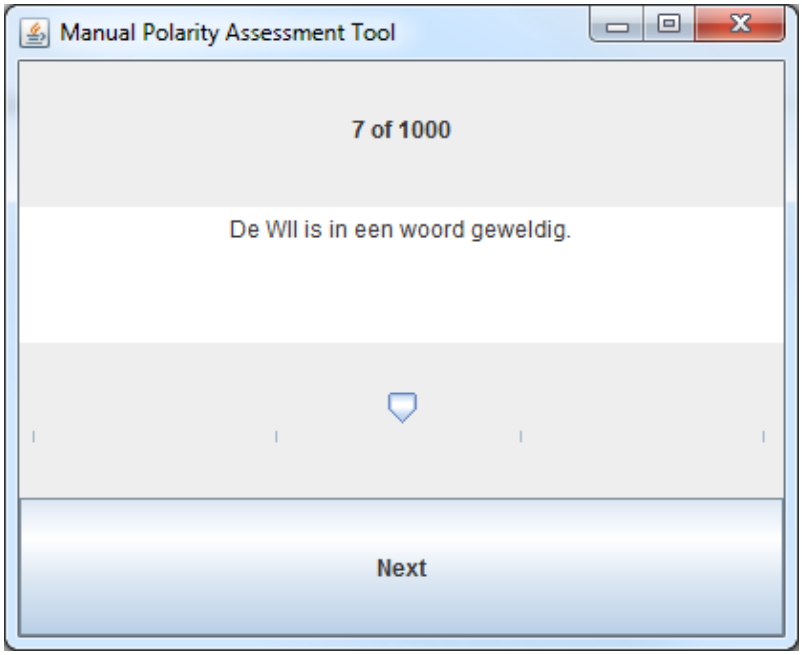


Figure 7.1: Screenshot of the tool used for manual annotation of test set

Kappa value	Interpretation
<0	less than change agreement
0.01-0.20	slight agreement
0.21-0.40	fair agreement
0.41-0.60	moderate agreement
0.61-0.80	substantial agreement
0.81-0.99	almost perfect agreement

Table 7.1: Interpretation of Kappa value

As mentioned before, there are three annotators, named 'Rater 1', 'Rater 2' and 'Rater 3'. Rater 1 has annotated the complete set, and Rater 2 and 3 both a different part. Since the goal is to classify a sentence as 'negative', 'neutral' or 'positive' instead of a numerical values, we have converted the numerical values given by annotators to those three classes. In addition to it, we have converted the list to only 'negative' or 'positive' by classifying ratings above 50 as positive and the remaining as negative. So, we have to find the agreement for both sets.

The Kappa values are given in table 7.2. We see that the values for only positive of negative (bi) indicates substantial agreement, although the values are on the boundaries of that category. The agreement for the division in three classes is at least 0.49 which indicates

	Rater 2		Rater 3	
	bi	tri	bi	tri
Rater 1	0.80	0.68	0.61	0.49

Table 7.2: Kappa values

moderate agreement. Those values corresponds to values found by others for similar tasks (Gamon et al., 2005; Schrauwen). We conclude that the results are reasonably good.

We have selected the sentences for which both annotators agree. This results in second test sets, one with two classification labels, and one with three classification labels. The first set includes 675 labelled sentences and the second 569. Since those sentences are randomly selected from the complete Review Corpus, it is a good representation.

7.1.4 Evaluation of Recognition Module

In the previous sections, the development and evaluation of a test corpus is described. We have used that test corpus to evaluate the developed Recognition Module. This evaluation is described here. Both the test sets with two and three category labels are used. For those evaluations, we have trained a classifier with all the 50 thousand reviews from the Review Corpus. We have used the most optimal parameter settings. The results for the two test sets are described separately.

Three Category Labels

To test the performance of the Recognition Module with the 'tri test set', a system is trained with all the reviews. The parameters used for this test are as follows. The parameter used to split the Review Corpus into three subsets, based on the grade of the reviews, is $[[0, 5.5], [5, 7], [6.5, 10]]$ for respectively the negative, neutral and positive set. We have tested multiple settings, and this own appears to provide the best results. This set is used to find the initial feeds. It means that reviews with a grade between 0 and 5.5 are selected for the negative set, reviews with grade between 5 and 7 for the neutral set and reviews with a grade between 6.5 and 10 for the positive set. We see that there is an overlap in these sets. Reviews with a grade between 5 and 5.5 are included in both the negative and neutral sets. In the same manner, reviews with a grade between 6.5 and 7 are included in both neutral and positive sets. Overlapping settings appear to be more optimal than non overlapping.

For all the sentences of the Review Corpus, we have computed which subset (negative, neutral or positive) matches the best with the words of the sentence. This match is computed as the relative frequency of the words of the sentence. The relative frequency is a number between 0 and 1, where 1 indicates that the words of the subset are exactly the same as the words in the sentence. A relative frequency of 0 means that no word of the sentence match with words in the subset. The sentence gets the label of the set for which the relative frequency is highest and goes beyond a threshold. This threshold is a parameter of our system. Based on several tests, it turns out that a threshold of 0.01 performs the best.

		Assigned label			
		P	O	N	Total
Label	P	194	25	28	247
	O	139	53	46	238
	N	22	21	41	84
	Total	355	99	115	

Table 7.3: Confusion matrix for three destination classes

The last parameter used for this system is the set of features. We have used the adjective words, the position of the sentence in the review, the number of negations, polarity information and the number of polarity words. For the adjective words we have used the Modified Value Distance Metric to compute the distance between two instances. The other features used the numeric metric.

The system trained as described, is evaluated with the 'tri test set'. A confusion matrix is given in table 7.3. The label 'O' stands for neutral. A remarkable result is that the system has a preference of either negative or positive. Although many neutral sentences exists, only a small amount is actually recognized as neutral. This indicates how hard it is to classify a sentence as neutral.

In 50% of the cases, the correct label was assigned. The corresponding F_β score of the system is 0.53 with $\beta = 1$. The F_1 score can be interpreted as a weighed average of the precision and recall, where it reaches its best value at 1 and worst score at 0. to The recognition of positive sentiment is most easy for the system. The recall of the positive sentences is 0.79 and the corresponding precision 0.54. This precision is like the precision of the neutral sentences, but the recall is higher than 0.22 for neutral sentences. For negative sentences, the precision and recall are 0.36 and 0.49 respectively. So, the results for this 'tri test set' are not very high. Therefore, we will examine our system with an easier task, namely to classify the sentences into two groups.

Two Category Labels

This test is conducted in the same manner as the previous system. We have used the next parameters to train our system: The parameter used to split the Review Corpus is $[[0,5],[5,10]]$. The threshold for the relative frequency is 0.2. We have used adjective words, the position of the sentence in the review, the number of negations, polarity information, the number of polarity words and the grade of the review as features. For distance computation, all distances are computed with the Modified Value Distance Metric, except for the grade feature which is numerical computed.

The system, trained with the given parameters, is evaluated with the 'bi test set'. A confusion matrix is given in table 7.4. For the positive class, a precision of 0.83 and a recall of 0.91 is reached. For the negative class, the precision and recall are respectively 0.43 and 0.25. The F_β score of the complete system is 0.76 with $\beta = 1$. We see that the performance of the system will increase to respectable values with classifying into two

		Assigned label		
		P	N	Total
Label	P	492	46	538
	N	103	34	137
	Total	595	80	

Table 7.4: Confusion matrix for two destination classes

groups. The recognition of negative sentences is quite hard for the system, this is caused by the skewness of the data. The amount of positive reviews is higher than the negative reviews.

7.2 Presentation Module

This section describes the evaluation process applied to determine the performance of the third module. The goal of the third module is to present the essence of the clusters graphically. A cluster is compound of a 'cloud' in the middle and affective term around it. To determine the performance of this module, we want to know how well the affective terms are selected. Also, we want to know if the chosen presentation agrees with the expectation of the user.

We have conducted an user experiment to find an answer on those questions. To answer the question about the right affective terms, users are asked to fill in a questionnaire about an automatically generated visualization. But before we could generate a visualization, we need a set with clustered sentences.

Since the second module is not implemented, we have manually created this clustered set of sentences. Therefore, we have selected the product which contains the most reviews. This resulted in about 100 reviews about the Nintendo Wii console.[1] In total there are 950 different sentences in this review set. Out of those 950 sentences, we have selected 100 sentences at random. We have divided the sentences into 6 different clustered groups. The number of clusters is determined by the subjects and the subjects are 'target group', 'working of product', 'time of ownership', 'comparison with other game consoles', 'playing', and 'conclusion'. The clusters include 9 to 26 sentences. This set with clusters is used as input for the Presentation Module.

First, we have generated a visualization with the created cluster set. This results in a set with pictures, for each cluster one. An example of such a cluster is presented in figure 7.2. This information is output of the Presentation Module and is not manually adjusted. While only one cluster is given here, there are six different clusters generated.

For the second step of the evaluation, we have selected the sentences which are used create the clusters. Both, the sentences and the visualization are provided on paper to the participants. See appendix C for an example. The participants are asked to underline the key words in the sentences. They are instructed to underline only the words which are key

[1] http://en.wikipedia.org/wiki/Wii

Figure 7.2: This figure shows the output of our Presentation Module for the cluster with sentences which contain a comparison with other products. The background colour is an indication for the polarity. The thickness of the line indicates how relevant the word is for the cluster.

words for the whole set of sentences. The participants are told what the cluster name is. This is done so that they know which subject is important. In sum, the participants are asked to achieve the same task as the system automatically will do.

In the second part of the evaluation, the participants are asked to rate the output of the system. The created visualization is shown and the participants are asked to indicate their agreement with three statements about each affective keyword. The first is about the relevance, the second and third about the presentation. So, we end up with three values of agreement for all the key words in the visualization. The participants are asked to indicate their agreement on a 5-point Likert scale. It is a widely used approach to scaling responses in survey research. The items of the Likert scale range from strongly agree to strongly disagree.

The evaluation is carried out under 5 participants, and they have all evaluated the same three clusters.

7.2.1 Agreement

An interrater reliability analysis using the Kappa statistic was performed to determine consistency among raters. Since the agreement is measured with a 5-point Likert scale, the weighted Kappa statistic is used (Cohen, 1968). Weighted Kappa is used to count disagree-

	1	2	3	4	5
1	-	0.279	0.494	0.396	0.400
2	0.279	-	0.213	0.035	0.178
3	0.494	0.213	-	0.427	0.482
4	0.396	0.035	0.427	-	0.468
5	0.400	0.178	0.482	0.468	-

Table 7.5: This table presents the Weighted Kappa values to indicate the agreement between raters.

ment for ordered levels. The formula for the weighted Kappa is:

$$\kappa = 1 - \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}x_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}m_{ij}}$$

where $k =$ number of codes (5 in this case), w_{ij}, x_{ij}, and m_{ij} are elements in the weight, observed and expected matrices, respectively. The observed matrix contains on the indices (i, j) a count that rater 1 and rater 2 have assigned an i and j respectively. The expected score is based on chance agreement. The weight matrix has on the diagonals zeros and it represent total agreement. Off-diagonal cells contain weights indicating the level of disagreement and it runs from zero to one.

The Weighted Kappa values are given in table 7.5. Those values are computed for the three statements. On average, the Kappa value is 0.337 and can be interpreted as fair agreement.

The agreement of the statement about relevance is further analysed. The reason for it is that relevance of the affective keywords is one of the most important factors of the Presentation Module. Again, the Weighted Kappa values are computed. The values are higher than the Weighted Kappa values for the three statements together. On average, the value is 0.373, this can be interpreted as fair agreement. Notice that the maximum score indicates substantial agreement.

The level of agreement is not very high and this indicates that relevance assessment isn't an easy task. In literature, we found that this is the same for comparable evaluation tasks. Hori et al. (2003) have evaluated automatic summaries of text generated through sentence of word extraction. They reported Kappa values of 0.35 and 0.39. In addition, the Kappa values reported by Radev et al. (2003) are even lower. Therefore, although there is low agreement, the results are comparable with other evaluations.

7.2.2 Results

The results of the user evaluation are analysed. In figure 7.3 a boxplot is given. We have taken the median of the five ratings. We see that the performance of the system for choosing a background color is very good. In at least 75% of the cases, the median rating is 'agree' or 'strongly agree'. In only three cases (those cases are shown as outliers and are numbered) the median of the ratings is 'disagree' or 'strongly disagree'.

	1	2	3	4	5
1	-	0.277	0.513	0.307	0.500
2	0.277	-	0.106	0.033	0.277
3	0.513	0.106	-	0.399	0.657
4	0.307	0.033	0.399	-	0.658
5	0.500	0.277	0.657	0.658	-

Table 7.6: This table presents the Weighted Kappa values to indicate the agreement between raters on the Relevance aspect.

The second point is about the *line* aspect. We see that the median of the median is 'neutral'. If we take the mean of the means, we see that the mean is also between zero. It indicates that it is very hard for the raters to give their opinion about the thickness of the lines. We conclude that it is a too vague concept and that raters in the majority of the cases are not able to provide a sensible meaning. Therefore, we will not further analyse this aspect.

The third point of interest is the aspect 'Relevance'. We see that more than half of the median of the raters is 'agree' or 'strongly agree'. This means that most of the automatic determined affective keywords are rated as relevant. This is quite a good performance. To get more insight into the ratings given by the participants, figure 7.4 is generated which includes all the ratings, instead of the medians. It is remarkable that the some raters seems to like the 'agree' rating while others are more likely to choose 'strongly agree', for example raters 2 and 4.

As mentioned before, the participants have rated three different clusters. We have investigated the differences of the clusters. The results are shown in figure 7.5. All ratings are used and not the mean or median. It is remarkable that the ratings for cluster 5 are very divergent. Since the participants have also manually underlined the relevant words, we can compare the automatically created visualization with the words chosen by the participants. Cluster 5 is about the time that the Wii is in use by the review author or how often the Wii is used. The participant have underlined words or phrases like 'about 4 week', 'half a year', and '1 month'. Those words are not included in the automatically generated visualization. This is caused by the selection process, which only selects adjective words. Therefore, we conclude that selecting only adjective words is not enough. We need to select also nouns and phrases should be included in the output.

7.3 Conclusion

In this chapter the implemented modules where evaluated. The Recognition Module is evaluated using a Gold Standard test set and the Presentation Module is evaluated with an user experiment. The Gold Standard is created by manually annotating sentences on polarity. In total 750 sentences are annotated with two annotations. There exist substantial

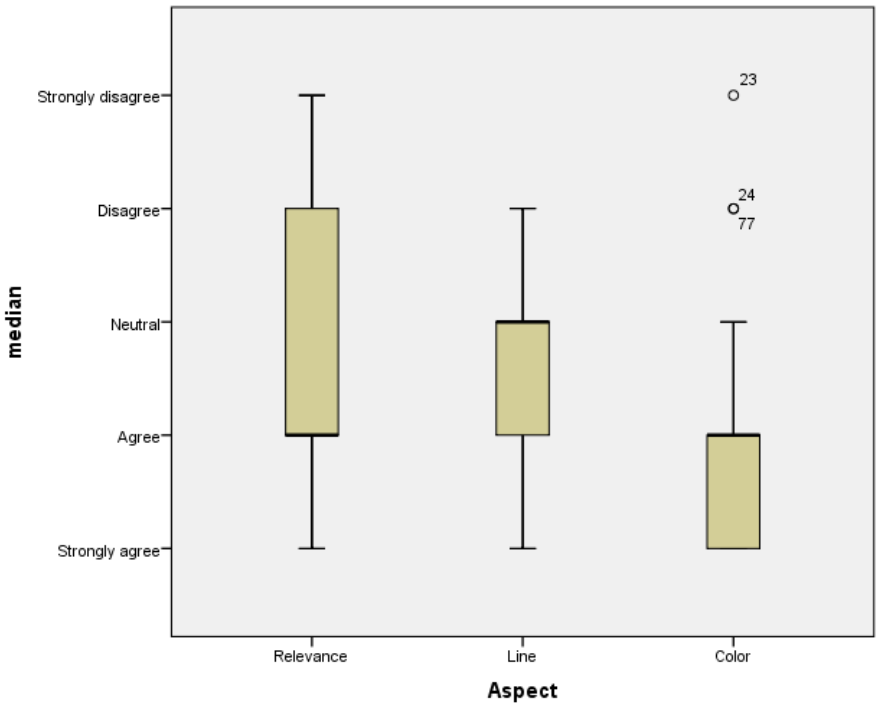


Figure 7.3: In this figure, boxplots are given about three aspects. On the vertical axis, the median is given of the ratings of the participants. It shows that the median of the raters agree with the given affective keywords. The median of the thickness aspect is 'neutral'. In about 75% of the cases, the median rating for background color is 'agree' or 'strongly agree'.

agreement between annotators. The sentences for which both annotators agrees are included in the Gold Standard.

The Gold Standard is used to evaluate our Recognition Module. The best performance is reached by classifying sentences into two groups and the corresponding F_β score is 0.76 with $\beta = 1$.

The Presentation Module is evaluated with an user experiment. Five participants are asked to identify key words for three clusters of sentences. Besides this, they are asked to indicate their agreement about the relevance and presentation of automatically determined keywords on a 5-point Likert scale. The agreement between the 5 raters is comparable with similar experiments reported in literature.

In more than 75% of the cases, the raters agreed with the automatically determined background color of the automatically determined keywords. 56.2% of the ratings indicates that the automatically determined keywords are relevant.

In the current model, only adjective words are included in the keywords found by the Presentation Module. The experiments indicate that including phrases and nouns improves the accuracy of the system.

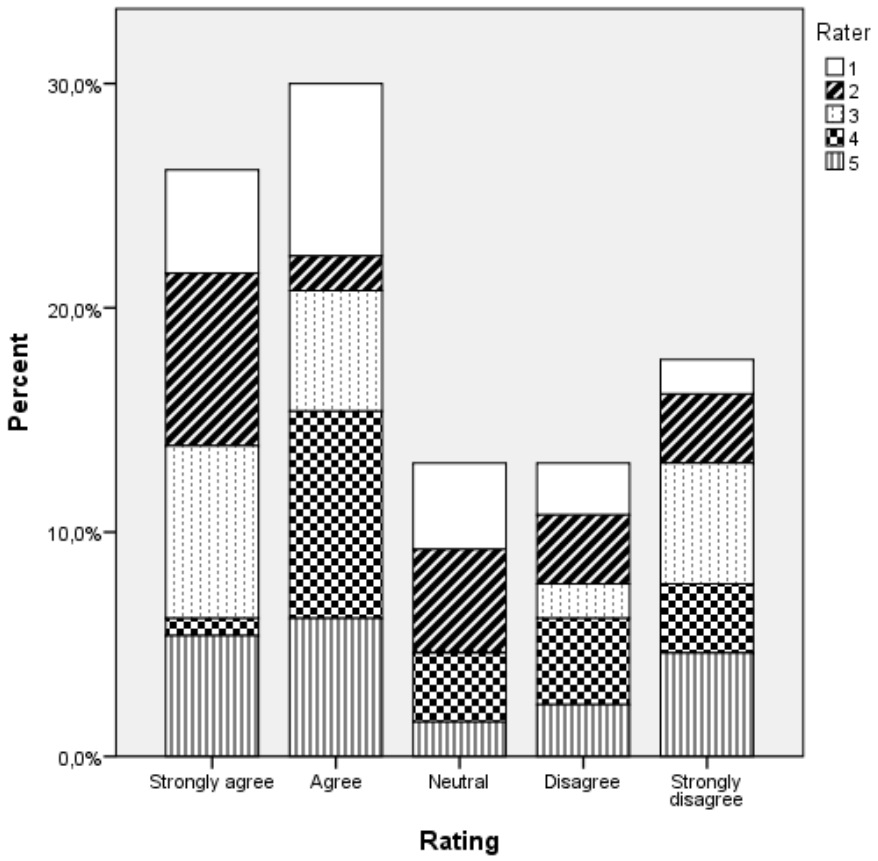


Figure 7.4: This figure shows all the ratings given by the five participants. More than 50% of the ratings agrees with the relevance of an affective keyword.

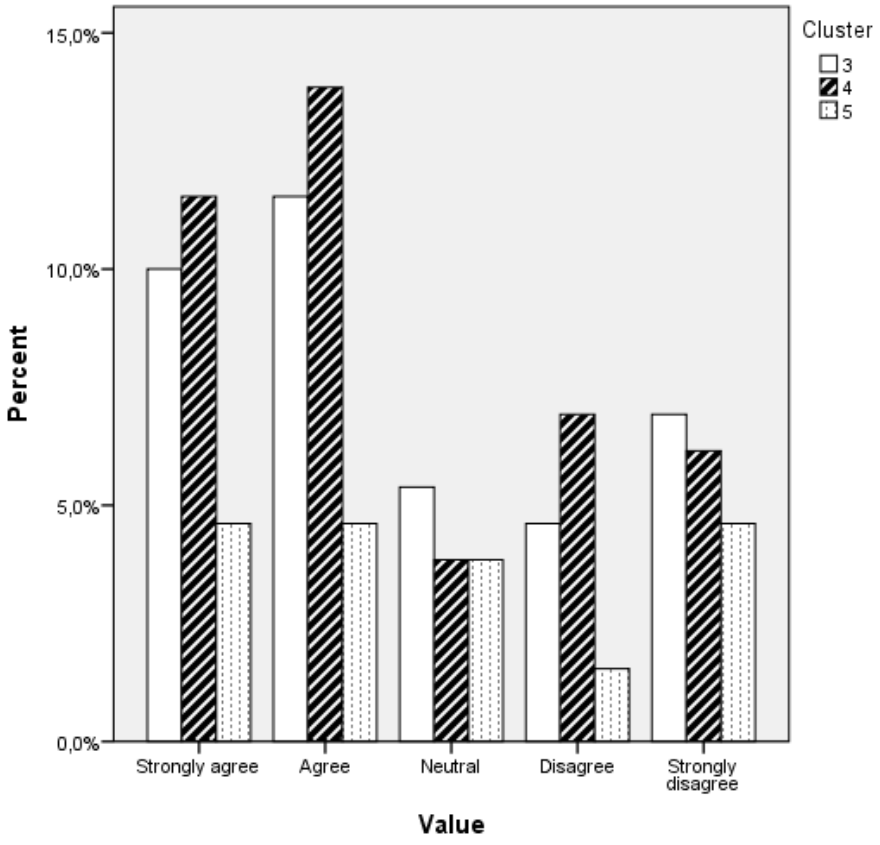


Figure 7.5: This figure shows the ratings given on the relevance aspect for different clusters. It is remarkable that cluster 5 seems worse than the clusters 3 and 4.

Chapter 8

Conclusion

The focus of this graduation project is on summarization of the affective content in product reviews. The goal is to create a visual summary of the opinions in those product reviews. A model is developed for the recognition, grouping and presentation of the opinions. In this chapter, a summary and conclusion are given about the results of this project.

8.1 Model

The goal of our model is to recognize the affective states in a text. Those states are clustered and finally the clusters are presented to the user. So, the model is divided into three modules. The first for recognition, the second for clustering and the third for presentation. First, the three modules are globally discussed.

The recognition of the affective states in a text is done using a classifier. In this implementation the valence of sentences is recognized, i.e. positiveness or negativeness. The input of the system consists of complete, unannotated product reviews, the output is a set of sentences with an above-average valence value.

The classifier is trained with an automatically selected set of sentences. Those sentences are labelled based on the relative word frequency which is computed based on multiple sets of reviews. The sentences selected are used as initial feeds for a bootstrapping algorithm. In this algorithm the set with initial feeds is extended and the final classifier is build.

The built classifier is used to find sentences with an above-average valence value. Those sentences are the input for the second module of the model: the module for clustering. In this module, the set with sentences is grouped into multiple sets based on the words in the sentences. First, the lemmas are determined, after it, semantic relations are added by use the *Cornetto* database. After it, $tf \times idf$ values are determined and the maximum cluster is extracted. The above described process is repeated until all sentences are clustered, or a certain threshold is reached.

The third module is meant to visualize the results to the user. The first step is to find a name for the cluster. To do so, the words in the cluster are enriched with more abstract words, which are obtained by use of the semantic database *Cornetto*. After it, the affective words are identified. The last step of this module is to create a graphic which exists

of 'clouds' with the name of the cluster. Around this 'cloud' the affective words are presented. Colours are used to display the polarity, and the thickness of the line indicates the occurrences of the affective term.

8.2 Evaluation

Parts of model as described in brief above are implemented and evaluations are conducted. More than 50 thousand reviews are used to train the recognition module. This module is evaluated by use of a manually annotated test set. Several people are asked to classify up to 750 sentences. The annotators agreed on a moderate scale. The sentences for which both annotators agreed are used to evaluate the recognition module. Classifying sentences as 'negative', 'neutral' or 'positive' is correctly done in 51% of the test cases. Better results are reached when classifying sentences as either 'negative' or 'positive'. In that case the overall accuracy is 81%. The corresponding F_β scores are 0.53 and 0.74 respectively with $\beta = 1$.

The best results are obtained by using the next features: Adjective words, number of polarity words, number of negations and polarity of sentence. Pre classification of the grade into a label seems not to improve the performance of a classifier. Also, including the grade of the review doesn't lead to better performance. It is remarkable that the number of negations improves the performance of a classifier. This information is also included in the pure words feature, but for some reason that information is not noticed by the classifier.

The accuracy of the system is reasonable good, but not very high. We think that this is caused by the unsupervised training method, and more specific, the selection procedure for the initial feed sentences. However, the results obtained are comparable with the results of similar research (Gamon et al., 2005; Popescu and Etzioni, 2005).

The Presentation Module is evaluated using an user experiment. Participants are asked to indicate their agreement about the relevance and presentation of automatically determined keywords. In 56 % of the cases, raters indicate that the keywords are relevant. The background color of the keywords is in more than 75 % of the ratings indicated as corresponding with the polarity.

The current Presentation Module uses only adjective words, as defined in the model. The experiments indicate that including nouns improves the accuracy of the system. Another result is that one word is too less to indicate aspects meaningful. Therefore, including phrases will improve the accuracy of the system.

8.3 Delivered Products

In the above two sections, the most important parts of this graduation project are described. However, there are more products delivered during this project. All the delivered products are described in sum in the list below. For an extended discussion, the reader is referred to other parts of this thesis.

- Review Corpus
  A new corpus is constructed. This corpus consists of more than 50 thousand product

reviews, downloaded from the website `www.beslist.nl`. All reviews are written in Dutch. This review corpus is used to train our system. See section 4.1.

- Annotated sentence set
  A randomly selected set of sentences are selected from the Review Corpus. Those sentences are rated by annotators on a scale between very negative and very positive. This test set contains 750 sentences and could be used to either test or train classifiers. See section 7.1.

- Tool for polarity assessment of sentences
  This tool is created to enable people to annotated quickly a set with sentences. It is used to annotate the above described sentence set. See section 7.1.

- Cluster Dataset
  For the input of the third module, which is responsible for the representation, a sample dataset is manually created. This dataset consists of 100 sentences which are clustered into 7 groups. Since the clustering module is not implemented yet, this work around is used.

- Model
  A complete model is developed, as described in section 8.1.

- Recognition Module
  The recognition module of the model is implemented. It can be used to extract sentences with above averaged polarity values.

- Presentation Module
  The presentation module of the model is implemented. In the presentation module, a cluster is analysed and presented as a graphic.

- Literature survey
  The theoretical background about Sentiment Analysis and Summarization is given in the chapters 2 and 3.

8.4 Research Questions

In the previous sections, all the work performed during this graduation project is described. Now it is time to provide answers to the research questions as defined in section 1.2. First, the questions are repeated one by one and follow with an answer.

1. What is state of the art of sentiment analysis and summarization?

This thesis started with two chapter with a theoretical background. In those chapters, the questions is answered and the findings are summed up as follows. The field of sentiment analysis is a sub field of affective computing and deals with the computational treatment of opinion, sentiment and subjectivity in text. Different applications are suggested such as review- and opinion-aggregation websites.

Three theoretical approaches of emotion are discussed, namely discrete theories, which emphasize a small set of fundamental emotions. Dimensional theories describes emotion in terms of a small set of dimension. Most multi-dimensional models include valence, arousal and dominance. The third type of theories are the component process theories which emphasize a set of domain-independent features of the situation and the appraising agent.

Several approaches of affect detection from text are discussed, including corpora-based, lexical and semantic approaches. Although many approaches have been suggested, not so many fully automatic applications exist.

Automatic creation of summarizations is a field with a long history. It started with surface approaches in which location information (e.g. header, first sentence, last sentence) and statistical information such as word frequency are used to build summaries. More sophisticated approaches make use of the discourse structure in a text. Relations are filtered out and used to build a summary. It is still a challenging task to create summaries like humans do.

2. What are the key concepts of a model that can reason about the affective states in a text?

The first step toward a model that can reason about the affective states is to recognize the affective states. Once it is recognized, it can be further analysed. It depends on the used datasets what kind of reasoning is possible. Good data is indispensable for reasoning. With good data a dataset is meant in which words or sentences are combined with an affective state. (See chapter 2 and 5.)

3. Which features can be used to analyse affect?

In section 5.5.5 different possible features are described. Based on the experiments in chapter 7 we conclude that, for our model, the combination of next features preformed the best: Adjective words, number of polarity words, number of negations and polarity of sentence.

4. Which affect representation is the most appropriate one for an affect analyser?

In the theoretical part, several different theories are described to represent emotion. This question asks which of those theories is most suitable for the analysis of affect. Based on literature, we found that most of the developed systems make use of dimensional representation methods (Pang and Lee, 2008).

We were not able to carry out an experiment to find the best representation method. The reason for it is that the needed data resources don't exist. As mentioned before, there aren't so many Dutch data resources and we were forced to create our own. Since generation of such data resources is very time consuming, we were not able to do it for different representations.

5. What kind of model can be used to graphical represent a summary of reviews?

In the model as described in chapter 5, a summary is generated by clustering the most opinionated sentences. For each cluster a name is given and the sentimental words are determined. Those information is combined in a graphic which enables users to get an overview of the reviews. 56 percent of the asked panel marks the presented information as relevant.

8.5 Future Work

As mentioned before, this graduation project is, as far as we know, the first investigation in the Dutch language field with the focus on summarization of reviews. Therefore, we recommend to work out this concept.

8.5.1 Corpora

A basic need for Natural Language Processing is the availability of corpora. Those corpora should contain much data which is representative for the language field it is meant for. Besides this, the data should be annotated with annotations specific for the mentioned usage. For example, for the development of a part-of-speech tagger, we need data annotated with the part-of-speech. In the same manner, for an opinion analysis system, data annotated with sentimental information is needed.

Unfortunately, such kind or corpora are not available for Dutch. Only the Duoman corpus contains sentimental information on the word level, but this corpus is created using an English corpus which is translated and further processed. Therefore, we need a large Dutch corpus annotated with sentimental information. Since sentiment analysis depends strongly on the domain, this corpus should include a broad spectrum of domains.

Most corpora nowadays available (for English) contain sentimental information based on polarity. In chapter 2 different representation methods for emotions are discussed. Based on those theories, we conclude that the use of polarity annotations only, not all emotions can be specified. Therefore, corpora with polarity annotations only covers a (small) part of the emotional spectrum. Not all the information in the text can be used. For that reason, we suggest to annotate the new Dutch corpora on multiple dimensions. For example, the suggested valence, arousal, and dominance scale could be used.

8.5.2 Representation

The third module in our model is the presentation module. This module creates a graphical summary representation. We recommend further investigation of the most convenient representation method. The current representation is theoretical founded, but not tested on a large panel. We suggest to apply a broad scale customer survey with different representations. The goal is to find the key points which are important for customers. We have to find answers on questions like “Is subject clustering valuable?”, “Are adjectives good indicators of the opinions in a text?”, and “Which colour scheme is the most convenient?”. This customer survey will be a initial step for further research.

8.5.3 Output phrases

In the current model, only single adjective words are used as output of the Presentation Module. Based on user experiments we have concluded that single adjective words are not expressive enough to be always meaningful. Therefore, investigation is needed to adapt the model to analyse phrases. A good starting point is to use bi-grams and tri-grams of words. This means that two or three words are analysed instead of one single word. In fact, the current model is a mono-gram model.

8.5.4 Bootstrapping feeds

The Recognition Module applies a bootstrapping method to train a classifier. The initial feeds of this bootstrapping algorithm are selected based on the relative frequency of words in the sentences. Further investigation is needed to optimize this method since the initial feeds are rather important for the performance of the final classifier. If we are able to neglect sentences which are not key sentences of the dataset, the performance should increase. A suggestion for a method is to build a 'stop-word' list. This list could include (but is not limited to) words used to build complex or compound sentences, for example: except, but, however. In most cases, such sentences are used to present more than one point.

8.5.5 Extra classification layer

Based on related research (Wiebe and Riloff, 2005; Hu and Liu, 2004), we may hypothesize that the accuracy of our Recognition Module could be improved by adding an extra classification layer. In the current model, the sentences of reviews are directly classified on polarity. Perhaps, this is a too challenging task. Therefore, we can first divide the sentences into a group of facts and a group of opinions. In the next step, the opinions are classified as either positive or negative.

Bibliography

Barzilay, R. and Elhadad, M., "Using lexical chains for text summarization," *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, 17(1):10–17 (1997).

Bloemfield, L., *Language*, Holt, Rinehard and Winston, New York, NY (1933).

Boguraev, B. and Kennedy, C., "Salience-based content characterisation of text documents," in I. Mani and M.T. Maybury, (editors) "Proceedings of the ACL97EACL97 workshop on Intelligent scalable text summarisation," pp. 3–9, The MIT Press (1997).

Bradley, M. and Lang, P., *Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1, Gainesville, FL*, Ph.D. thesis, University of Florida (1999).

Bradley, M.M. and Land, P.J., "Affective norms for English Text (ANET): Affective ratings of text and instruction manual." (2007).

Calvo, R.A. and D'Mello, S., "Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications," *IEEE Transactions on Affective Computing*, 1(1):18–37 (2010).

Carbonell, J. and Goldstein, J., "The use of MMR, diversity-based reranking for reordering documents and producing summaries," *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336 (1998).

Cohen, J., "Weighed kappa: Nominal scale agreement with provision for scaled disagreement or partial credit," *Psychological Bulletin*, 70(4):213–220 (1968).

Daelemans, W. and Van den Bosch, A., *Memory-based Language Processing*, University Press, Cambridge (2005).

Das, D. and Martins, A.F.T., "A Survey on Automatic Text Summarization," *Literature Survey for the Language and Statistics II Course at CMU*, 4:192–195 (2007).

De Saussure, F., "Cours de linguistique générale," in "Paris: Payot. Edited posthumously by C. Bally A. Sechehaye, and A. Riedlinger. Citation page numbers and quotes are from the English translation by Wade Baskin, New York: McGraw-Hill Book Company, 1966," (1916).

De Vries, S., *Samenvatten van Nederlandse teksten volgens Rhetorical Structure Theory*, Master's thesis, Universiteit van Tilburg (2005).

Edmundson, H., "New methods in automatic extracting," *Journal of the ACM (JACM)*, 16(2):264–285 (1969).

Ekman, P. and Others, "An argument for basic emotions," *Cognition and emotion*, 6(3/4):169–200 (1992).

Eppler, M.J., "A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing," *Information Visualization*, 5(3):202–210 (2006).

Evans, D., Ku, L., Seki, Y., Chen, H. and Kando, N., "Opinion analysis across languages: An overview of and observations from the NTCIR6 opinion analysis pilot task," *Applications of Fuzzy Sets Theory*, pp. 456–463 (2007).

Fellbaum, C., *WordNet: An Electronic Lexical Database*, *Language, speech, and communication*, volume 71, MIT Press (1998).

Ferré, P., "Advantage for emotional words in immediate and delayed memory tasks: could it be explained in terms of processing capacity?" *The Spanish journal of psychology*, 5(2):78–89 (2002).

Fleiss, J., Cohen, J. and Everitt, B., "Large sample standard errors of kappa and weighted kappa," *Psychological Bulletin*, 72(5):323–327 (1969).

Foong, O., Oxley, A. and Sulaiman, S., "Challenges and Trends of Automatic Text Summarization," *International Journal of Information and Telecommunication Technology*, 1(1):34–39 (2010).

Gamon, M., Aue, A., Corston-oliver, S. and Ringger, E., "Pulse : Mining Customer Opinions from Free Text," pp. 121–132 (2005).

Greenway, D., Sandoz, E. and Perkins, D., "Potential applications of relational frame theory to natural language systems," *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 6:2955–2958 (2010).

Hofmann, K., Maks, I. and Vossen, P., "The Design of the Cornetto Database," Technical Report August (2008).

Hori, C., Hirao, T. and Isozaki, H., "Evaluation Measures Considering Sentence Concatenation for Automatic Summarization by Sentence or Word Extraction," in "Proceedings of the ACL-04," pp. 82–88 (2003).

Horrigan, J.B., "Online Shopping," *Pew Internet & American Life Project Report* (2008).

Hu, M. and Liu, B., "Mining and summarizing customer reviews," *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, p. 168 (2004).

Hudlicka, E., *Affective Computing: Theory, Methods and Applications* (2009).

Jezek, K. and Steinberger, J., "Automatic Text Summarization (The state of the art 2007 and new challenges)," *Znalosti*, Bratislava:1–11 (2008).

Jijkoun, V. and Hofmann, K., "Task-based Evaluation Report: Building a Dutch Subjectivity Lexicon," Technical Report September (2008).

Jijkoun, V. and Hofmann, K., "Generating a Non-English Subjectivity Lexicon: Relations That Matter," in "Computational Linguistics," April, pp. 398–405 (2009).

Jin, X., Li, Y., Mah, T. and Tong, J., "Sensitive webpage classification for content advertising," (2007).

Ku, L., Lo, Y. and Chen, H., "Test Collection Selection and Gold Standard Generation for a Multiply-Annotated Opinion Corpus," *Computational Linguistics*, (June):89–92 (2007).

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

Larsen, R.J. and Diener, E., "Promises and problems with the circumplex model of emotion," *Review of Personality and Social Psychology*, 13(13):25–59 (1992).

Lin, C., "Rouge: A package for automatic evaluation of summaries," in "Proceedings of the workshop on text summarization branches out," pp. 25–26 (2004).

Loper, E. and Bird, S., "NLTK : The Natural Language Toolkit," in "Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics," pp. 63–70 (2002).

Luhn, H.P., "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development*, 2(2):159–165 (1958).

Lund, K., "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods* (1996).

Lutz, C. and White, G.M., "The Anthropology of Emotions," *Annual Review of Anthropology*, 15(1):405–436 (1986).

Mani, I. and Maybury, M.T., *Advances in Automatic Text Summarization*, volume 26, MIT Press (1999).

Mann, W. and Thompson, S., "Rhetorical structure theory: Toward a functional theory of text organization," *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281 (1988).

Marcu, D., "From discourse structures to text summaries," *Proceedings of the ACL*, 97:82–88 (1997).

Nosofsky, R., "Attention, similarity, and the identification-categorization relationship," *Journal of Experimental Psychology*, General(15):39–57 (1986).

Ono, K., Sumita, K. and Miike, S., "Abstract generation based on rhetorical structure extraction," *Proceedings of the 15th conference on Computational linguistics* -, p. 344 (1994).

Osgood, C.E., Suci, G.J. and Tannenbaum, P.H., *The measurement of meaning* (1971).

Ounis, I. and Macdonald, C., "On the TREC blog track," *Proceedings of the International Conference on Weblogs and Social Media* (2008).

Ozsoy, M.G., *Text Summarization using Latent Semantic Analysis*, Ph.D. thesis, METU (2011).

Pang, B. and Lee, L., "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, 2:1–135 (2008).

Panksepp, J., *Affective neuroscience: The foundations of human and animal emotions*, volume 4, Oxford University Press, USA (2004).

Pantic, M. and Rothkrantz, L.J.M., "Toward an affect-sensitive multimodal human-computer interaction," *Proceedings of the IEEE*, 91(9):1370–1390 (2003).

Pennebaker, J.W., Francis, M.E. and Booth, R.J., *Linguistic Inquiry and Word Count*, Lawerence Erlbaum Associates (2001).

Picard, R.W., *Affective Computing*, 321, MIT Press (2000).

Popescu, A.M. and Etzioni, O., "Extracting product features and opinions from reviews," *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pp. 339–346 (2005).

Radev, D., Teufel, S., Saggion, H. and Lam, W., "Evaluation challenges in large-scale document summarization," *Proceedings of the 41st*, pp. 375–382 (2003).

Riloff, E. and Phillips, W., "Exploiting Subjectivity Classification to Improve Information Extraction," *Computing*, pp. 1106–1111 (2005).

Russell, J.A., "A circumplex model of affect," *Journal of Personality and Social Psychology*, 39(6):1161–1178 (1980).

Russell, J.A. and Lanius, U.F., "Adaptation level and the affective appraisal of environments," *Journal of Environmental Psychology*, 4(2):119–135 (1984).

Salton, G. and Buckley, C., "Term-weighting approaches in automatic text retrieval," *Information Processing*, 24(6):713 (1988).

Samsonovich, A.V. and Ascoli, G.A., "Cognitive map dimensions of the human value system extracted from natural language," in "Proc. AGI Workship Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms," pp. 111–124 (2006).

Scherer, K., "Emotions as episodes of subsystem synchronization driven by nonlinear appraisal processes," *Emotion, development, and self-organization: Dynamic systems approaches to emotional development*, pp. 70–99 (2000).

Schrauwen, S., "CTRS-001," *Machine Learning* ().

Schuemie, M.J., Kors, J.A. and Mons, B., "Word sense disambiguation in the biomedical domain: an overview." *Journal of computational biology : a journal of computational molecular cell biology*, 12(5):554–65 (2005).

Seki, Y., Eguchi, K., Kando, N. and Aono, M., "Multi-document summarization with subjectivity analysis at DUC 2005," in "Proceedings of the Document Understanding Conference (DUC)," Citeseer (2005).

Smith, E. and Medin, D., *Categories and concepts*, Harvard University Press, Cambridge, MA (1981).

Sparck Jones, K., "Automatic summarizing: factors and directions," in I. Mani and M.T. Maybury, (editors) "Advances in automatic text summarisation," pp. 1–14, MIT Press, Cambridge, MA (1999).

Sparck Jones, K., "Automatic summarising: The state of the art," *Information Processing & Management*, 43(6):1449–1481 (2007).

Steinberger, J., "Using latent semantic analysis in text summarization and summary evaluation," *Proc. ISIM*, 39(04) (2004).

Stone, P., "The General-Inquirer," *http://www.wjh.harvard.edu/˜inquirer* (2000).

Strapparava, C., Valitutti, A. and Stock, O., "The affective weight of lexicon," in "Proceedings of the 5th International Conference on Language Resources and Evaluation LREC," pp. 423–426 (2006).

Taboada, M., "Applications of Rhetorical Structure Theory," *Discourse Studies*, 8(4):567–588 (2006).

Tatemura, J., "Virtual reviewers for collaborative exploration of movie reviews," in "Proceedings of the 5th international conference on Intelligent user interfaces," pp. 272–275, ACM (2000).

Terveen, L., Hill, W., Amento, B., Mcdonald, D. and Creter, J., "A System for Sharing recommendations," *Communications of the ACM*, 40(3):59–62 (1997).

Ward, J., "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association* (1963).

Watson, D. and Tellegen, A., "Toward a consensual structure of mood." *Psychological Bulletin*, 98(2):219–235 (1985).

Wiebe, J. and Riloff, E., "Creating Subjective and Objective Sentence Classifiers from Unannotated Texts," pp. 486–497 (2005).

Wiebe, J., Wilson, T. and Cardie, C., "Annotating Expressions of Opinions and Emotions in Language," *Language Resources and Evaluation*, 39(2-3):165–210 (2006).

Yu, H. and Hatzivassiloglou, V., "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in "Proceedings of the 2003 conference on Empirical methods in natural language processing-Volume 10," pp. 129–136, Association for Computational Linguistics (2003).

Appendix A

Class Diagrams

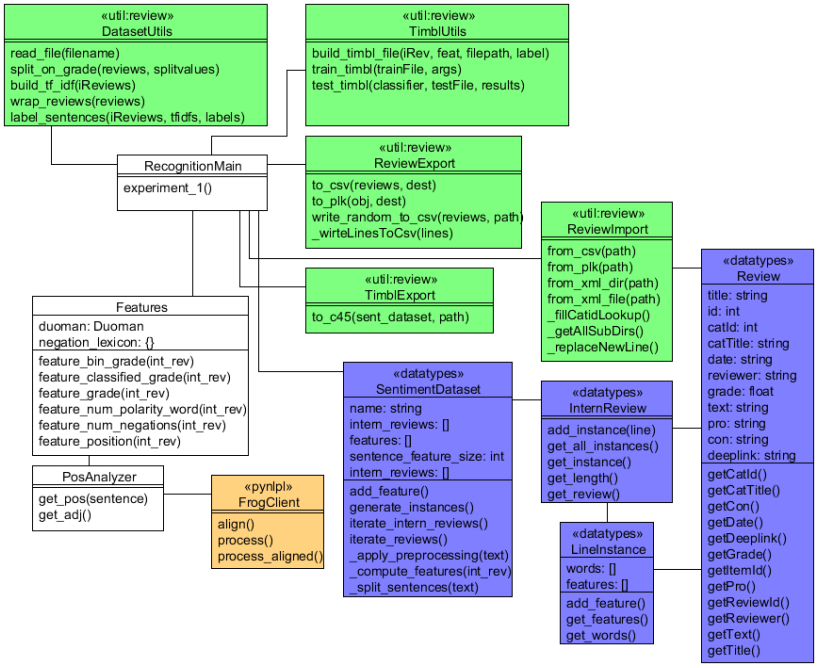


Figure A.1: The class diagram for the Recognition Module. All the classes are written within the framework of this thesis project. The different background colours indicate the package to which the class belongs. The classes are divided into different packages. The classes with the purple background belong to the *datatypes* package. This package is meant to contain all classes used to represent data structures. The green background indicates that the class belongs to the *util* package. This package contains all kind of helper classes, including import and export functionality. The orange background contains classes which are meant for the link with TiMBL functionality.

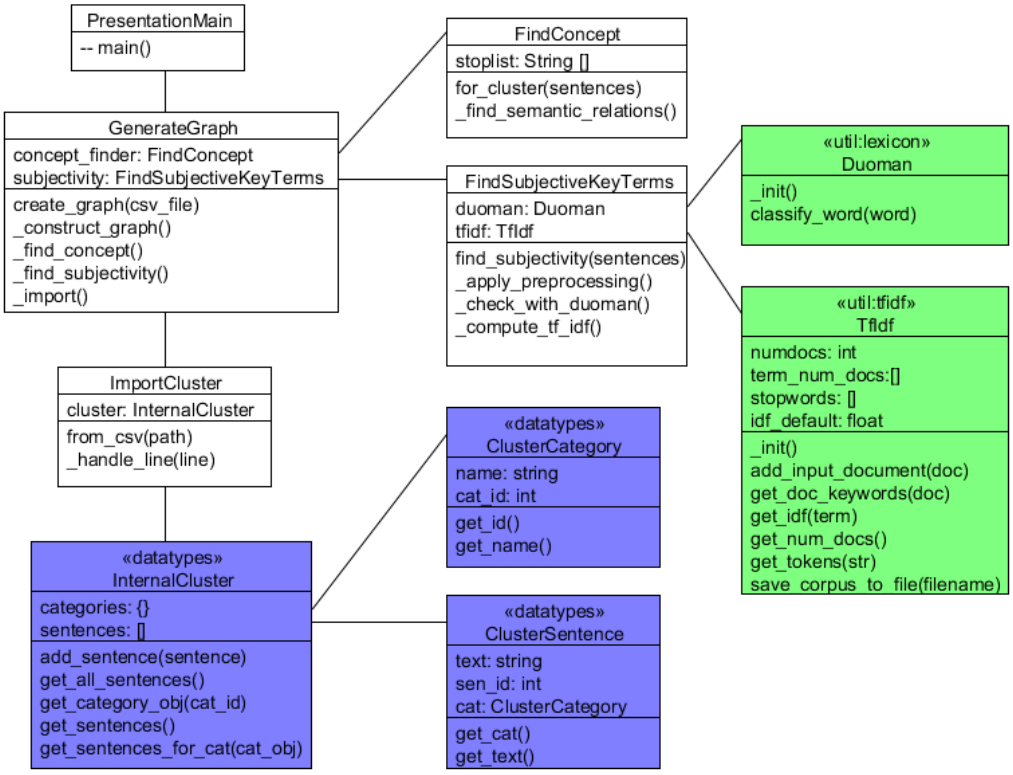


Figure A.2: Class diagram of Presentation Module

Appendix B

Handleiding voor classificeren

Je wordt gevraagd om de sentimentele lading van een set met zinnen te classificeren. Hiervoor gebruik je een computerprogramma dat de zinnen een voor een laat zien. Hierbij geef je per zin aan in hoeverre een zin positief of negatief bedoelt is.

Als je een zin beoordeelt, probeer dan te bedenken welk sentiment de schrijver van de zin bedoelt. Verwar dit niet met je eigen interpretatie.

In figuur B.1 zie je een screenshot van het programma. Bovenaan is te zien met welke zin je bezig bent en hoeveel er in totaal zijn. Eronder wordt de zin weergegeven. Daaronder is een 'schuifregelaar' weergegeven waarop je aan kunt geven hoe positief of negatief de zin is. Het sentiment van een zin kun je aangeven op een schaal van 'heel negatief' tot 'heel positief', respectievelijk helemaal links en helemaal rechts. Middenin geeft aan dan een zin 'neutraal' is en dus geen sentiment bevat. Als je naar een volgende zin wilt gaan, klik je op de onderste knop, het is niet mogelijk om terug te gaan. Tip: Als je met de rechtermuisknop op een positie in de schuifregelaar klikt, wordt de zin geclassificeerd en ga je direct door naar de volgende zin. Hierdoor kun je erg snel door de set heen.

Figure B.1: Screenshot van de Classificatie Tool

Een zin is ‘heel negatief’ als een negatief woord in een zin benadrukt wordt, en ‘heel positief’ als een positief woord in een zin benadrukt wordt. Voorbeeld: “Ik vind dit echt heel gaaf!” is maximaal positief (dus helemaal rechts op de schuifregelaar), maar een zin als “Dit product moet je echt nooit kopen!” is juist heel erg negatief (dus helemaal links op de schuifregelaar). Een neutrale zin bevat helemaal geen sentiment, bijvoorbeeld: “Vorige week heb ik een computer gekocht.” of “Ik ben nu thuis.”.

Veel succes!

Appendix C

Evaluatie Presentation Module

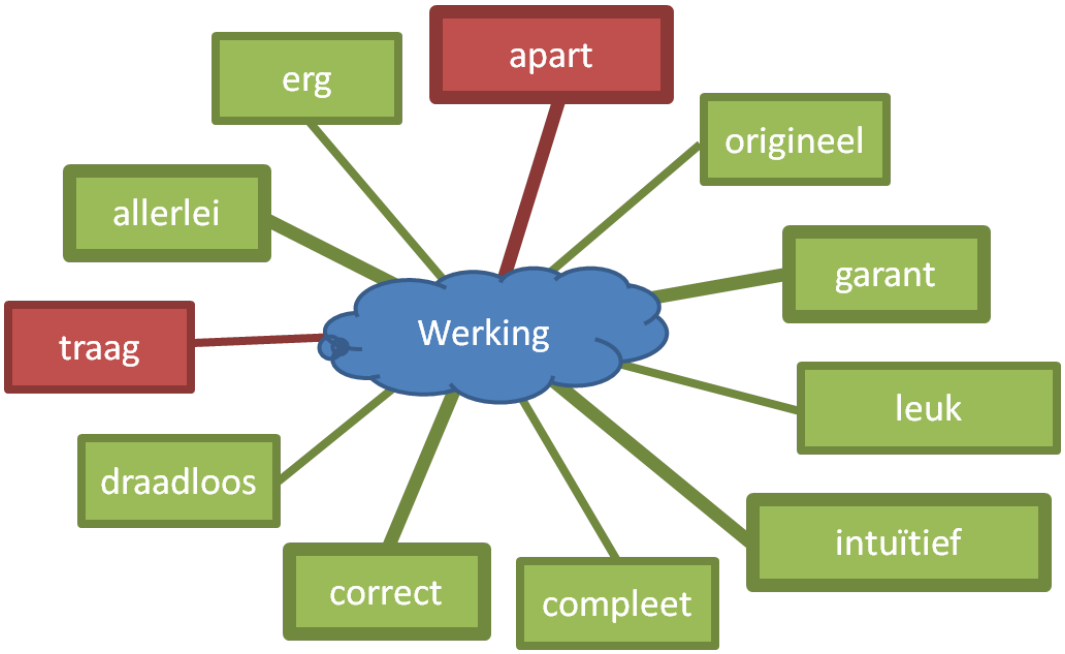
Cluster 4, Werking

Instructie: Lees eerst de zinnen hieronder door. Lees ze daarna nogmaals door en onderstreep de woorden die relevant zijn voor de complete tekst. Als je dit gedaan hebt kun je doorgaan met het tweede onderdeel van de evaluatie. Al deze zinnen gaan over de nintento Wii, een populaire spelcomputer. Ook zouden ze allemaal over n onderwerp moeten gaan.

1. Het bijgeleverde Wii Sports staat al garant voor uren speelplezier, maar kan je ook zeker aanraden een 2e console aan te schaffen in combinatie met Wii Play.
2. De besturing is heel erg origineel.
3. het vernieuwende aan de wii is de aparte besturing, dit gaat door middel van het bewegen van je controler met je arm/ hand.
4. Je kan allelei uitbreidingen krijgen in de vorm van guns en rackets en sticks, om het nog leuker te maken.
5. De internetverbinding maakt het helemaal compleet.
6. Draadloze controllers en een mooi design.
7. Alleen de afstandsbedieningen werken niet altijd helemaal correct, zijn soms wat traag.
8. Alleen vergeten ze de batterijen wel eens op te laden.
9. Door gebruik te maken van een wii-wheel is de besturing namelijk erg intuitief.
10. Je kunt er veel leuke en aparte aspecten bij kopen voor de meeste spelletjes, zoals badjes, racket en stuur.
11. Vooral heel erg leuk zodra je meerdere controllers hebt om samen met vrienden te spelen.

Instructie: Nu begin je met het tweede deel van de evaluatie. Dit betreft het evalueren van de uitkomst van een computer programma. Hieronder zie je een afbeelding die een samenvatting weergeeft van de zinnen uit het eerste gedeelte. Het middelste woord geeft het onderwerp van de zinnen aan. Een groene achtergrond geeft een positief woord aan, een rode achtergrond geeft aan dat het woord negatief bedoeld is. De dikte van de lijn is een indicatie voor hoe belangrijk het woord voor de tekst is.

Geef aan in hoeverre je het eens bent met een stelling. De eerste stelling gaat erover of het woord relevant is voor de samenvatting van deze groep zinnen. De tweede of de dikte van de lijn overeen komt met de mate van relevantie. De derde stelling gaat over de achtergrondkleur.



	Het woord is relevant voor de tekst.						De lijndikte geeft mate van relevantie aan.						De kleur komt overeen met gevoelslading.				
	Helemaal mee eens	Eens	Neutraal	Oneens	Helemaal mee oneens		Helemaal mee eens	Eens	Neutraal	Oneens	Helemaal mee oneens		Helemaal mee eens	Eens	Neutraal	Oneens	Helemaal mee oneens
erg																	
apart																	
origineel																	
garant																	
leuk																	
intuïtief																	
compleet																	
correct																	
draadloos																	
traag																	
allerlei																	