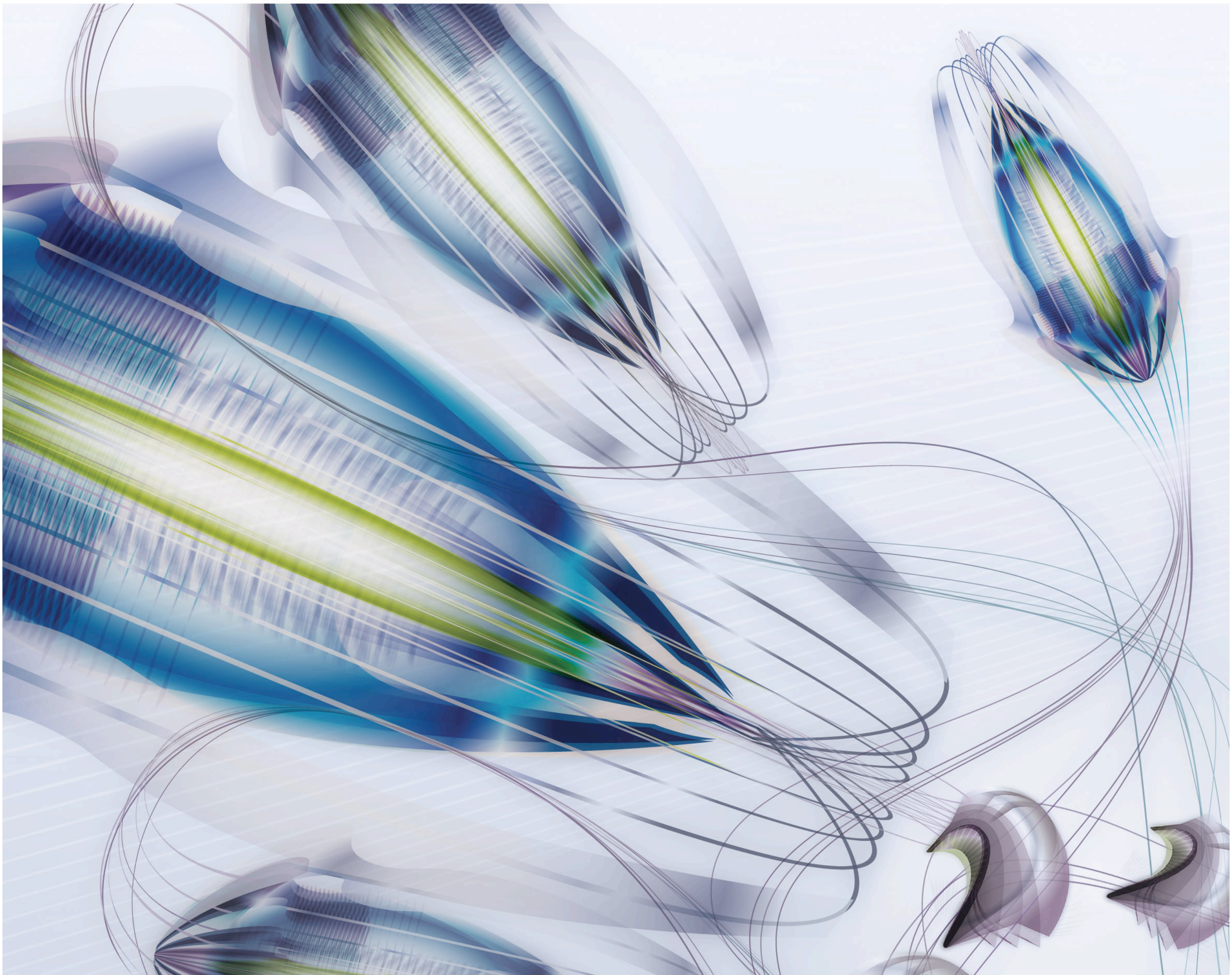
CLiPS 1

COMPUTATIONAL LINGUISTICS & PSYCHOLINGUISTICS
TECHNICAL REPORT SERIES, CTRS-001, JULY 2010

WWW.CLIPS.UA.AC.BE/CTRS

Machine Learning Approaches to Sentiment Analysis using the Dutch Netlog Corpus

Sarah Schrauwen



Universiteit
Antwerpen

COMPUTATIONAL LINGUISTICS &
PSYCHOLINGUISTICS
RESEARCH CENTER CLiPS

CTRS-001

28 July 2010

MACHINE LEARNING APPROACHES TO SENTIMENT ANALYSIS USING THE DUTCH NETLOG CORPUS

Sarah Schrauwen

MACHINE LEARNING APPROACHES TO SENTIMENT ANALYSIS USING THE DUTCH NETLOG CORPUS[1]

Sarah Schrauwen
CLiPS, University of Antwerp
Lange Winkelstraat 40-42, 2000 Antwerp (Belgium)
Sarah.Schrauwen@student.ua.ac.be

ABSTRACT

Sentiment analysis deals with the computational treatment of opinion, sentiment and subjectivity. We constructed and manually annotated a corpus, the Dutch Netlog Corpus, with data extracted from the social networking website Netlog. This corpus was annotated on three levels: 'valence' (expressing the opinion of the writer; we distinguish between 'positive', 'negative', 'both', 'neutral' and 'n/a') and additionally language performance, which is divided into two areas: 'performance' ('standard' versus 'dialect') and 'chat' ('chat' versus 'non-chat'). We tackle sentiment analysis as a text classification task and employ two simple feature sets (the most frequent and the most informative words of the corpus) and three supervised classifiers implemented from the Natural Language ToolKit (the Naïve Bayes, Maximum Entropy and Decision Tree classifiers). The highest obtained accuracy score for valence classification with the entire data set is a 65.1%. We suggest three factors leading to errors in valence classification. First, the nature of the data affects results, since most of the corpus is made up of dialect and chat language, which is more difficult to predict. Second, the number of classes to predict from is larger for valence classification (five classes) than for performance or chat classification (three classes for both), and is therefore also more difficult to process. Third, the skewed class distribution of the corpus probably has the biggest influence on the results. Some classes in the corpus are very well represented, while others are relatively rare, which leaves less training data for the classifiers to work with. We suspect that more training data will solve these current problems. The highest accuracy result for performance classification is 77.6%; for chat classification, it results in 84.2%.

[1] This is an abbreviated version of the Master's thesis *Machine Learning Approaches to Sentiment Analysis Using the Dutch Netlog Corpus*, written for obtaining the degree of Master in Computational Linguistics at the University of Antwerp in 2010.

Table of contents

Preface

Imagining a world without the World Wide Web seems quite impossible for the average contemporary man or woman. It would mean no more 'instant' or 'free' anything: no latest e-mails or headlines, no up-to-date weather forecast, no accurate stock market news, no googling, no quick information check-up on Wikipedia, and so on.

Statistics depicting the Internet are often shocking: in 2009, the Web had a staggering amount of 1,802,330,457 users worldwide, which accounts for 26.6% of the entire world population[2]. The most popular activities online are communication via e-mail and looking for information on goods and services[3]. The latter involves reading customer reviews and feedback for an honest evaluation of a product (unlike glorified product summaries provided by the selling party); about 81% of the Internet users have done online research on a product at least once (Pang & Lee 2008) and between 73% to 87% report that these reviews had a significant influence on their purchase (Pang & Lee 2008). Another popular activity on the Web is using social networking websites: 73% of the teenagers and 72% of the young adults use them worldwide[4]. In the future, the amount of Internet users will grow exponentially. The European Council even aims to achieve 100% coverage of the European population by 2013[5].

Looking at Internet statistics in Belgium reveals that in 2007 about 69% of the Flemish population used the Internet[6]. Comparing age and level of education leads to notable percentages: 95% of the people between 16 and 24 years old use the Internet, against 21% of the people between 65 and 74. Higher educated people are almost omnipresent with 92%, against 45% of the people with a lower educational background[7]. 40% of the Belgians are active on at least one social networking website, and about 20% of the population is active on two or more social networking websites.

Keeping these numbers in mind, it is imaginable what vast amounts of data are gathered in this 'one' place called the Internet. In July 2010, the indexed Web contains a minimum of 28.3 billion web pages[8]. The Dutch indexed Web is accountable for at least 408.18 million pages of them[9]. All of these web pages are filled with valuable and sought-after information: *how people think and feel about things*. Whether they adore or hate something, and reasons why, can be very useful information when you want to form an opinion about that something as well. Buying cell phones, dishwashers and cars in this day and age seldom goes without first consulting some form of online review or customer feedback. Automatically extracting these data leads to huge corpora of information waiting to be researched and analyzed, eventually leading to groundbreaking applications in a number of fields.

[2] www.internetworldstats.com/stats.htm [2010-07-26]
[3] epp.eurostat.ec.europa.eu/portal/page/portal/publications/regional_yearbook [2010-02-03]
[4] www.pewinternet.org/Reports/2010/Social-Media-and-Young-Adults.aspx [2010-02-03]
[5] epp.eurostat.ec.europa.eu/portal/page/portal/publications/regional_yearbook [2010-02-03]
[6] This information comes from the 2007 ICT survey of the *Algemene directie Statistiek en Economische Informatie.*
[7] Idem
[8] www.worldwidewebsize.com [2010-07-26]
[9] Idem

Sentiment analysis (see Section 1.1.2), recently discovered by the academic world as a useful line of research and receiving an increasingly growing interest from the natural language processing community, is definitely a booming business. This growing interest is particularly motivated by the widespread need for opinion-based applications, such as product and movie reviews, entity tracking and analysis and opinion summarization (Banea et al. 2008). There are at least 20 to 30 companies that offer sentiment analysis services in the USA alone (Liu 2010: 1), and even a small country like Belgium already has companies entirely devoted to it (e.g. Attentio[10]). Economic relevance is assured, since many market research firms are waiting to get their hands on structured and analyzed online data informing them about their product being "FTW"[11] or in tomorrow's garbage bin.

This report is based on the Master thesis "Machine Learning Approaches to Sentiment Analysis Using the Dutch Netlog Corpus" for achieving the Master of Computational Linguistics at the University of Antwerp. Supervisor and assessor of this thesis were Walter Daelemans and Roser Morante respectively.

This study deals with sentiment analysis on a small manually tagged corpus (the Dutch Netlog Corpus, see Section 2) containing short messages extracted from the social networking website Netlog (see Section 2.1.). People express how they think and feel about people or things in many diverse subtle and complex ways. Categorising these opinions is often viewed as a challenging classification task; in the case of the Dutch Netlog Corpus, there are three annotation levels: valence (i.e. negative or positive value assigned by a person to another person, event, goal, object, etc.), language performance ('standard' versus 'chat') and chat ('chat' versus 'dialect'). For valence, we opted for a five-way classification into 'positive', 'negative', 'both' (positive and negative in one single message), 'neutral' and 'n/a' classes. The performance and chat annotation levels are divided into threefold classification tasks: 'standard', 'dialect' and 'n/a' for the performance level, and 'chat', 'non-chat' and 'n/a' for the chat level.

The first section of this report introduces the concept of sentiment analysis, its (possible) applications and challenges. Furthermore, it gives an overview of previous work and discusses the goals of this study. The second section is devoted to the corpus that is constructed and used for this study, the Dutch Netlog Corpus (DNC). Methodologies available for sentiment analysis are dealt with in the third section. The experimental setup is discussed in section four, while section five reports on the results of the experiments. The sixth section summarizes the main accomplishments and findings of this study, and reports on some additional experiments. Section seven presents the main conclusions of this study, and makes some proposals for future work.

[10] www.attentio.com [2010-07-26]

[11] In 2010, FTW is an enthusiastic expression frequently used on the Internet (i.e. *netspeak*). It stands for "for the win", which is the same as saying "this is the best" or "this item is awesome, I recommend using it", etc. Mind you, it is also often used sarcastically. Years ago, it had a very negative connotation and was short for "f*ck the world".

Section 1
Introduction: What is Sentiment Analysis?

1.1. Sentiments and Sentiment Analysis

1.1.1. What are Sentiments?

Sentiments can be described as emotions, or as judgements, opinions or ideas prompted or coloured by emotions (Boiy et al. 2007: 350). In Computational Linguistics, the focus is on **opinions** rather than on sentiments, feelings or emotions, and the words 'sentiment' and 'opinion' are often used interchangeably, also in this report.

Textual information can be divided into two types: factual and opinionated information. While facts are objective expressions about entities, events and their properties, opinions are usually *subjective* expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties (Liu 2010: 1). Kim & Hovy (2004: 1367) on the other hand describe an opinion on the basis of four terms: Topic, Holder, Claim and Sentiment. The Holder believes a Claim about a Topic, and often associates a Sentiment, such as 'good' or 'bad', with the belief. They describe a Sentiment as an explicit or implicit expression in text of the Holder's positive, negative or neutral regard toward the Claim about the Topic; they always involve the Holder's emotions and desires.

In this study, we view an opinion as Liu (2010: 1) does: subjective information expressing the author's thoughts, ideas, judgements, or feelings toward someone or something.

1.1.2. What is Sentiment Analysis?

Sentiment analysis (also *sentiment mining, sentiment classification, opinion mining, subjectivity analysis, review mining* or *appraisal extraction*, and in some cases *polarity classification*) deals with the computational treatment of opinion, sentiment, and subjectivity in text (Pang & Lee 2008: 8). It intends to ascertain the attitude or opinion of a speaker or writer with respect to a certain topic or target. The attitude could reflect his/her judgment, opinion or evaluation, his/her affective state (how the writer feels at the time of writing) or the intended emotional communication (how the writer wants to affect the reader). Furthermore, it should be noted that in this context 'subjective' does not mean that something is *not true* (Mejova 2009: 3).

In sentiment analysis, we study **subjective language**: language used to express private states in the context of a text or conversation (Wiebe et al. 2004: 279). Quark et al. (1985) define a **private state** as a general covering term for opinions, evaluations, emotions, and speculations. There are three main types of subjective expressions: references to private states (e.g. “He was boiling with anger”), references to speech (or writing) events expressing private states (e.g. “The editors of the left-leaning paper attacked the new House Speaker”) and expressive subjective elements (see below, e.g. “That doctor is a quack”) (Wiebe & Mihalcea 2006: 1066).

Linguistic elements of private states in context are subjective elements (Wiebe et al. 2004: 280). A **subjective element** is uttered by a source who is either the writer of the text or someone mentioned in it (Wiebe et al. 2004: 280). A **nested source** is a source that is not directly speaking to us, but is being quoted by the writer (Wiebe et al. 2004: 280). Furthermore, this subjectivity is often about or directed toward someone or something, which we have already called the **target** (Wiebe et al. 2004: 280).

Subjective language, or the opinions uttered by a writer, can be categorized into a number of **classes** or categories: e.g. ‘positive’, ‘negative’ and ‘neutral’ (i.e. determining the *valence*); or into an *n*-point scale: e.g. ‘very good’, ‘good’, ‘satisfactory’, ‘bad’, ‘very bad’ (Prabowo & Thelwall 2009: 143); or into a number of emotions: e.g. ‘joy’, ‘sadness’, ‘anger’, ‘surprise’, ‘disgust’ and ‘fear’. In this respect, a sentiment analysis task is seen as a classification task where each category represents a sentiment (Prabowo & Thelwall 2009: 143). When dealing with merely two classes (e.g. “positive” versus “negative” or “good” versus “bad”), we speak of *polarity classification*. In the case of movie/product reviews, rating systems (e.g. stars) or the terms “thumbs up” and “thumbs down” are also frequently used (e.g Turney 2002; Pang, Lee & Vaithyanathan 2002).

The concept of subjectivity is fairly **ambiguous**; consider Pang & Lee’s (2008: 9) definitions of terms closely related to the notion of sentiment or opinion:

- *Opinion* implies a thought out conclusion that is still open to dispute (e.g. “Each expert seemed to have a different opinion.”);
- *View* suggests a subjective opinion (e.g. “Tom is very assertive in stating his views.”);
- *Belief* implies (often) deliberate acceptance and intellectual agreement (e.g. “She has a firm belief in her party’s platform.”);
- *Conviction* applies to a party’s firmly and seriously held belief (e.g. “The conviction that animal life is as sacred as human is what defines Shirley’s character.”);
- *Persuasion* suggests a belief grounded in assurance (e.g. “He was of the persuasion that everything changes over time.”);
- *Sentiment* suggests a settled opinion reflexive of ones feelings (e.g. “Her feminist sentiments are well-known”).

We can add a few definitions[12] to this non-exhaustive list:

- *Emotion* suggests a strong feeling (e.g. “I have difficulty controlling my emotions when something like this happens.”);
- *Feeling* implies an experience of affective and emotional states (e.g. “I like her, and apparently the feeling is mutual.”);
- *Thought* suggests a personal belief or judgment that is not founded on proof or certainty (e.g. “That thought had never occurred to me.”);
- *Impression* is a vague idea in which some confidence is placed (e.g. “Her first impression of him proved to be completely wrong.”).

[12] These definitions are gathered from WordNet (www.wordnet.princeton.edu) [2010-07-26].

One approach to sentiment analysis is to use a lexicon with information about which **words and phrases** are positive and which are negative (Wilson, Wiebe & Hoffmann 2009: 400). For instance, SentiWordNet is a publicly available lexical resource in which each WordNet synset is ascribed three numerical scores describing how objective, positive, and negative the terms in the synset are (Esuli & Sebastiani 2006b: 417). This lexicon can either be manually compiled (e.g. Stone's General Enquirer, Stone et al. 1966) or be acquired automatically. The annotation of lexica or corpora is usually done by hand, and classifiers are then trained with large sets of features to classify a new batch of words or phrases. Other approaches to sentiment analysis focus on the mining of **sentences or entire documents**, rather than to rely on the valence of words. This approach usually works with corpora of text documents. The main problem with document (polarity) classification is that it has to determine the *overall* sentiment properties of an entire document, while the expressed sentiment can be contained in just one sentence or word. In other cases, the sentiment can be expressed implicitly, which makes it even more difficult to detect and classify (see Section 1.3). However, the context surrounding these 'hidden' sentiments can provide very useful information for classifying it. Based on this division of the field of sentiment analysis, we often speak of **word-level**, **sentence-level** and **document-level** sentiment classification.

Another approach is the mining of sentiment **on the web**. Web opinion mining aims to extract, summarize, and track various aspects of subjective information on the Web (Ku & Chen 2007: 1838). This can prove helpful for advertising companies or trend watchers (see Section 1.2).

1.2. Applications

As said in the Preface, sentiment analysis is receiving an increasingly growing interest from the natural language processing community, which is particularly motivated by the wide-spread need for opinion-based applications, such as product reviews, entity tracking and analysis and opinion summarization (Banea et al. 2008).

The existence of the World Wide Web has changed the way that people express their views and opinions (Liu 2010: 1), and has provided researchers with a huge source of *user-generated content*. Wanting to buy a product no longer involves questioning friends or family; wanting consumer opinions about your own product no longer needs to rely on focus groups or external consultants (Liu 2010: 1). However, this huge resource of valuable information, the Web, is unstructured, and sentiment analysis is able to automatically discover opinions and present them in a structured manner.

Sentiment mining has become a useful tool for the commercial activities of both companies and individual consumers. They want to sort out opinions about products, services, or brands that are scattered in online texts such as product review articles or forums (Hiroshi, Tetsuya & Hideo 2004). In the following paragraphs we sum up a few important (future) applications of sentiment analysis.

Sentiment analysis can be used for **determining critics' opinions** about a given product (e.g. a digital camera, movie, etc.) by classifying online product reviews from websites such as Amazon and C|Net (e.g. Dave, Lawrence & Pennock 2003; Hu & Liu 2004), RottenTomatoes.com (e.g. Pang & Lee 2004) and IMDb (e.g. Pang, Lee & Vaithyanathan 2002), and can also prove very helpful for opinion-oriented questions in question answering (Pang & Lee 2008: 12). Tracking the shifting attitudes of the general public toward a political candidate by mining online forums is also a useful application (Esuli & Sebastiani 2006a: 193). It can furthermore be used to alert customer services of dissatisfied customers that utter their frustrations on forums or discussion boards. Tracking (mood) trends of bloggers is also becoming a valued research field, since it can be used for research in trends or consumer preferences.

In other words, applications resulting from sentiment analysis research can help a great deal in **marketing research** (i.e. quality control, automatic information gathering from the Internet instead of bothering customers with surveys (Boiy et al. 2007), etc.), and can consequently be of great service for advertising and market intelligence companies and trend watchers. In this respect, sentiment analysis can contribute to *collective intelligence* research: the study of the combination of behaviour, preferences, or ideas of a group of people to create novel insights (Segaran 2007: 2).

Sentiment analysis can also be helpful for **recommendation systems** (Pang & Lee 2008: 12), since those systems should not recommend something that receives negative feedback, and for the development of new kinds of search engines.

The **detection of "flames"**, overly heated or antagonistic language, in e-mails or on social networking websites will also benefit from sentiment classification. Monitoring newsgroups and forums, where fast and automatic detection of flaming is necessary (Boiy et al. 2007), will also see spectacular improvements.

Another application of sentiment analysis that is becoming hugely necessary is that of **opinion spam detection**. While e-mail and Web spam are quite familiar, opinion spam is still new to the general public. Because of the enormous growth of user-generated content on the Web, it is now a common practice for people to find and read other's opinions. Opinion spam refers to human activities that try to deliberately mislead readers or automated opinion mining systems by giving undeserving positive opinions to some target objects (in order to promote the objects, i.e. *hype spam*) and/or by giving unjust, malicious or false negative opinions to other objects (to damage their reputations, i.e. *defaming spam*) (Liu 2010: 28, 30). These opinions are also called *fake opinions* or *bogus opinions* (Liu 2010: 28). This kind of spam detection can also be considered as a classification problem, i.e. into 'spam' and 'non-spam' categories. The problem of detecting spam opinions will become more critical, because consumers and organizations will increasingly use the Web to search for opinions. However, identifying spam opinion is a very difficult task, even for humans.

A related problem that also has been receiving more attention over the past few years is the determination of the **usefulness, helpfulness or utility of a review**, which determines how helpful a review is to a user (Liu 2010: 29).

A large number of text processing applications have already employed techniques for automatic sentiment analysis (Banea et al. 2008: 127), for example automatic expressive text-to-speech synthesis (Alm, Roth & Sproat 2005), text semantic analysis (Wiebe & Mihalcea 2006; Esuli & Sebastiani 2006a), tracking sentiment timelines in online forums and news (Lloyd et al. 2005; Balo et al. 2006), mining opinions from movie reviews (Hu & Liu 2004), and question answering (Yu & Hatzivassiloglou 2003).

Future related projects are paedophilia and suicide detection on the Internet (e.g. on discussion boards, social networking websites, chat rooms, etc.).

For more applications of sentiment mining in domains such as business and government intelligence, we refer to the overview paper of Pang and Lee (2008: 11ff).

1.3. Challenges

One of the biggest problems in the field of Computational Linguistics is **ambiguity**. There are three areas in which we have to solve ambiguity: semantically, lexically and syntactically ambiguous text. This problem can only be tackled if computational systems possessing some form of *world knowledge* or at the very least a rudimentary dictionary and decision making skills become available. *Semantic ambiguity* has to do with a choice between any number of possible interpretations, and is therefore closely related to vagueness, e.g. idiomatic expressions that rarely or never have well-defined definitions. Lexical and syntactic ambiguity problems always involve semantic ambiguity as well. *Lexical ambiguity* deals with a choice between a finite number of known and meaningful context-dependent interpretations, for example 'bank' (financial institution, e.g. "I want to go to the bank to withdraw some money") and 'bank' (side of river, e.g. "Do you see that river? I'm going to live by its bank"). One way of dealing with lexical disambiguation is to look at other lexical elements frequently occurring in combination with or in proximity of the ambiguous word (money-bank versus river-bank). Aside from these two areas of ambiguity, *syntactic ambiguity* is also a huge obstacle, and deals with grammatical ambiguities; e.g. disambiguating the well-known sentence "Flying planes can be dangerous": is flying planes dangerous, or are flying planes dangerous? Context is also of crucial importance here.

Other problems are **co-reference** and **anaphora resolution**: sentences like "I want that one!" and "Queen Elizabeth II is the Queen regnant of sixteen independent sovereign states, she is politically neutral and by convention her role is largely ceremonial" do not imply what "that", "Queen regnant", "she" or "her" refer to, which makes it even more difficult to define which emotion is expressed and who or what is the target.

Emotions and opinions can be expressed explicitly and implicitly. **Implicitness** is a challenge for computational systems, since even for humans it is not easy to identify and analyze these expressions correctly. This also applies for humour, sarcasm, irony, etc.

Another problem is **inference**, the process of drawing conclusions by applying certain clues (logic, statistics, etc.) to observations or hypotheses. Inference has been a popular field of research, and applications such as expert systems and business rule engines have followed. The aforementioned world knowledge is used by an inference system to resolve ambiguities, and to find and identify implicit relations.

Unlike text categorization, which aims at classifying documents by topic, sentiment classification has relatively **few classes** (such as 'positive', 'negative' and 'neutral') that generalize across many domains and users (Pang & Lee 2008: 16). In fact, opinion classification is the most complicated when it also has to predict 'neutral' (Wilson, Wiebe & Hoffmann 2009).

Unique to sentiment analysis is the **regression-like nature** of strength of feeling, degree of positivity, subtlety, and so on (Pang & Lee 2008: 17). For more information about determining the strength of opinions, see 2.2.3.

Text documents often contain **multiple (contrasting) opinions**. It can therefore be useful to use a category such as 'both', which includes single messages that hold both positive and negative opinions.

Lastly, **domain-independence** is one of the biggest problems in machine learning and classification. A carefully selected feature set can produce very high accuracies for a certain corpus (e.g. newspaper corpus), but perform very poorly when applied to another kind of corpus (e.g. biomedical corpus). Finding innovative and effective approaches to overcome this problem is a valued research field.

1.4. Previous Work

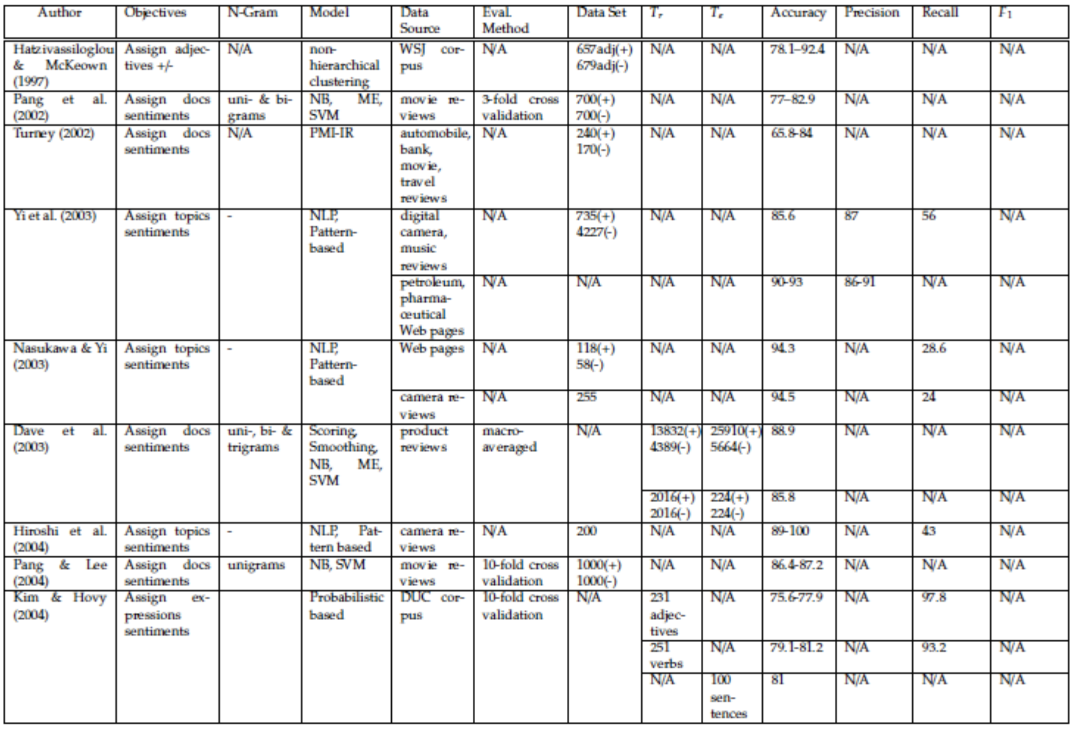
Over the past couple of years, many papers, books and dissertations have been written about opinion mining. While some researchers focus on more specific tasks, such as finding the sentiments of words (e.g. Hatzivassiloglou & McKeown 1997), subjective expressions (e.g. Wilson et al. 2005), subjectivity clues (e.g. hapax legomena and collocations, Wiebe et al. 2004: 278), subjective sentences (e.g. Pang & Lee 2004), topics (e.g. Yi et al. 2003), and extracting sources of opinions (e.g. Choi et al. 2005), other researchers focus on assigning sentiments to entire documents (e.g. Pang, Lee & Vaithyanathan 2002; Pang & Lee 2004).

All sorts of data sets have been used in these studies, for instance movie and product reviews from websites such as Amazon (e.g. Dave, Lawrence & Pennock 2003; Hu & Liu 2004), IMDb (e.g. Pang, Lee & Vaithyanathan 2002) and RottenTomatoes.com (e.g. Pang & Lee 2004), customer feedback from discussion boards and forums (e.g. Gamon 2004), the Multi-Perspective Question Answering (MPQA) corpus (e.g. Wiebe, Wilson & Cardie 2005; Choi et al. 2005; Wilson, Wiebe & Hoffmann 2009) and the Wall Street Journal (WSJ) corpus (e.g. Hatzivassiloglou & McKeown 1997; Yu & Hatzivassiloglou 2003). Others have shown an interest in mining political opinions from online forums (e.g. Esuli & Sebastiani 2006a: 193) or tracking (mood) trends on blogs (e.g. Mishne & Glance 2006).

Researchers of sentiment analysis have applied various approaches to 'automatically' predict the sentiments of words, expressions or documents. These are Natural Language Processing (NLP) and pattern-based techniques (e.g. Yi et al. 2003), machine learning algorithms such as Naïve Bayes, Maximum Entropy and Support Vector Machines (e.g. Joachims 1998), and unsupervised learning (e.g. Turney 2002); the majority of these approaches are discussed in Section 3.2.3.

Work in sentiment analysis has been subdivided in a number of ways. Godbole, Srinivasaiah & Skiena (2007) divide previous work in the context of their specific task (sentiment analysis for news and blogs) into two categories: one that relates to techniques for automatically generating sentiment lexica and one that relates to systems that analyze sentiment for entire documents. Esuli & Sebastiani (2006a) on the other hand, divide related work in two other categories: one deals with determining term orientation and the other deals with determining term subjectivity. These divisions only refer to research on term-level classification, and not document-level classification.

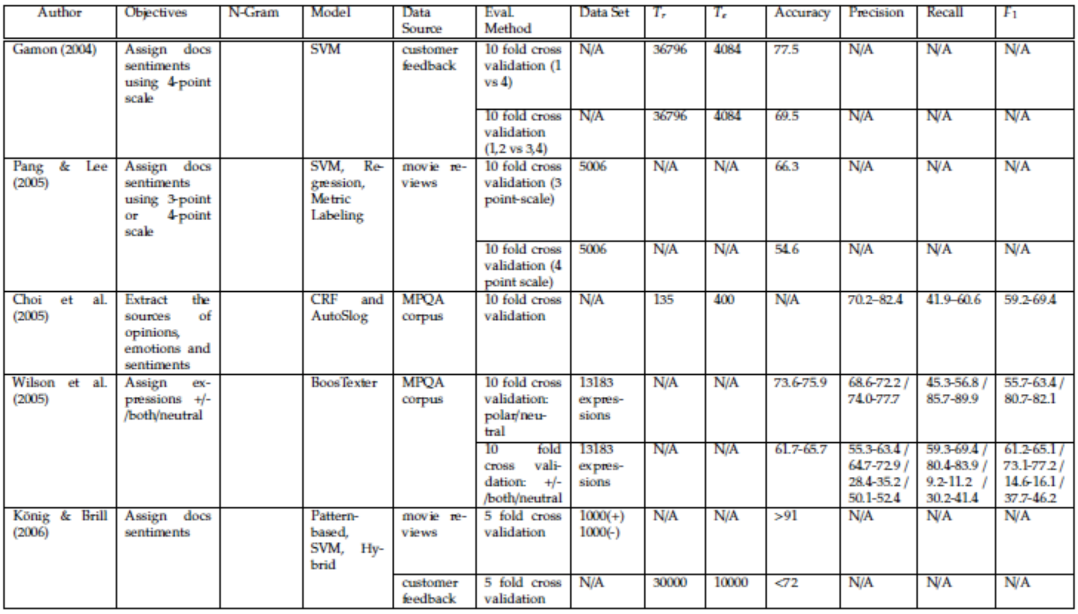
Prabowo and Thelwall (2009) assembled Tables 1 and 2[13], to present an overview of the literature on sentiment analysis until 2006.



Author	Objectives	N-Gram	Model	Data Source	Eval. Method	Data Set	T_r	T_e	Accuracy	Precision	Recall	F_1
Hatzivassiloglou & McKeown (1997)	Assign adjectives +/-	N/A	non-hierarchical clustering	WSJ corpus	N/A	657adj(+) 679adj(-)	N/A	N/A	78.1–92.4	N/A	N/A	N/A
Pang et al. (2002)	Assign docs sentiments	uni- & bi-grams	NB, ME, SVM	movie reviews	3-fold cross validation	700(+) 700(-)	N/A	N/A	77–82.9	N/A	N/A	N/A
Turney (2002)	Assign docs sentiments	N/A	PMI-IR	automobile, bank, movie, travel reviews	N/A	240(+) 170(-)	N/A	N/A	65.8-84	N/A	N/A	N/A
Yi et al. (2003)	Assign topics sentiments	-	NLP, Pattern-based	digital camera, music reviews	N/A	735(+) 4227(-)	N/A	N/A	85.6	87	56	N/A
				petroleum, pharma-ceutical Web pages	N/A	N/A	N/A	N/A	90-93	86-91	N/A	N/A
Nasukawa & Yi (2003)	Assign topics sentiments	-	NLP, Pattern-based	Web pages	N/A	118(+) 58(-)	N/A	N/A	94.3	N/A	28.6	N/A
				camera reviews	N/A	255	N/A	N/A	94.5	N/A	24	N/A
Dave et al. (2003)	Assign docs sentiments	uni-, bi- & trigrams	Scoring, Smoothing, NB, ME, SVM	product reviews	macro-averaged	N/A	13832(+) 4389(-)	25910(+) 5664(-)	88.9	N/A	N/A	N/A
							2016(+) 2016(-)	224(+) 224(-)	85.8	N/A	N/A	N/A
Hiroshi et al. (2004)	Assign topics sentiments	-	NLP, Pattern based	camera reviews	N/A	200	N/A	N/A	89-100	N/A	43	N/A
Pang & Lee (2004)	Assign docs sentiments	unigrams	NB, SVM	movie reviews	10-fold cross validation	1000(+) 1000(-)	N/A	N/A	86.4-87.2	N/A	N/A	N/A
Kim & Hovy (2004)	Assign expressions sentiments		Probabilistic based	DUC corpus	10-fold cross validation	N/A	231 adjectives	N/A	75.6-77.9	N/A	97.8	N/A
							251 verbs	N/A	79.1-81.2	N/A	93.2	N/A
							N/A	100 sentences	81	N/A	N/A	N/A

Table 1: Existing work in sentiment analysis

[13] A list of the (model) abbreviations used in these tables: Naïve Bayes classifier (NB), Maximum Entropy classifier (ME), Support Vector Machines (SVM), Point-wise Mutual Information and Information Retrieval (PMI-IR), Natural Language Processing (NLP), Conditional Random Field (CRF), Probabilistic Term Frequency and Inverse Document Frequency (PrTF-IDF) and Hidden Markov Models (HMMs). A few of these techniques are discussed in Section 3.



Author	Objectives	N-Gram	Model	Data Source	Eval. Method	Data Set	T_r	T_e	Accuracy	Precision	Recall	F_1
Gamon (2004)	Assign docs sentiments using 4-point scale		SVM	customer feedback	10 fold cross validation (1 vs 4)	N/A	36796	4084	77.5	N/A	N/A	N/A
					10 fold cross validation (1,2 vs 3,4)	N/A	36796	4084	69.5	N/A	N/A	N/A
Pang & Lee (2005)	Assign docs sentiments using 3-point or 4-point scale		SVM, Regression, Metric Labeling	movie reviews	10 fold cross validation (3 point-scale)	5006	N/A	N/A	66.3	N/A	N/A	N/A
					10 fold cross validation (4 point scale)	5006	N/A	N/A	54.6	N/A	N/A	N/A
Choi et al. (2005)	Extract the sources of opinions, emotions and sentiments		CRF and AutoSlog	MPQA corpus	10 fold cross validation	N/A	135	400	N/A	70.2–82.4	41.9–60.6	59.2-69.4
Wilson et al. (2005)	Assign expressions +/-/both/neutral		BoosTexter	MPQA corpus	10 fold cross validation: polar/neutral	13183 expressions	N/A	N/A	73.6-75.9	68.6-72.2 / 74.0-77.7	45.3-56.8 / 85.7-89.9	55.7-63.4 / 80.7-82.1
					10 fold cross validation: +/-/both/neutral	13183 expressions	N/A	N/A	61.7-65.7	55.3-63.4 / 64.7-72.9 / 28.4-35.2 / 50.1-52.4	59.3-69.4 / 80.4-83.9 / 9.2-11.2 / 30.2-41.4	61.2-65.1 / 73.1-77.2 / 14.6-16.1 / 37.7-46.2
König & Brill (2006)	Assign docs sentiments		Pattern-based, SVM, Hybrid	movie reviews	5 fold cross validation	1000(+) 1000(-)	N/A	N/A	>91	N/A	N/A	N/A
				customer feedback	5 fold cross validation	N/A	30000	10000	<72	N/A	N/A	N/A

Table 2: Existing work in sentiment analysis (continued)

These tables provide a good reference for researchers of sentiment analysis, but much new and innovative work has been produced since 2006. Therefore, we assembled Table 3, an overview of some past literature on various domains of sentiment analysis. This table is not exhaustive.

Author	Objectives/goals	Model(s)	Data source	Data set
Esuli & Sebastiani (2006)	Solving problem of determining subjectivity and orientation	NB, SVM, Rocchio, PrTF-IDF	General Inquirer Lexicon	4,206 subjective (+ 1,915, - 2,291), 5,009 objective
Banea et al. (2008)	Prove that machine translation can be used to generate a subjectively-annotated corpus and to effectively train tools for subjective analysis	SVM, NB	Machine Translation (MT) of manually annotated corpora (MPQA corpus), MT of source language training data (subset of SemCor), MT of target language training data (subset of SemCor), MT of target language test data (subset of SemCor)	9,700 subjectively annotated sentences (MPQA), 11,000 sentences (subset of SemCor)
Benamara et al. (2007)	Assign a number from -1 to +1 to denote the strength of sentiment on a topic in a sentence or document, based on the score assigned to the applicable adverb-adjective combinations found in sentences	Three adjective-adverb combinations: Variable scoring, Adjective Priority Scoring, Adverb First Scoring	Annotated set of documents selected randomly from a set of popular news sources	200 documents
Boiy et al. (2007)	Compare own results with past literature: what are best machine learning techniques and best features?	SVM, NB, ME	Pang & Lee's movie review corpus, Corpus gathered from blogs, discussion boards and other websites	+ 1,000 and - 1,000 (Pang & Lee), + 759 and - 205 (own corpus), 1,965 neutral, 1,562 junk examples

Kobayashi, Inui & Matsumoto (2007)	Build a computational model to extract opinions from Web documents in a form such as "*Who* feels *how* on *which aspects* of *which subjects*"	Various classifier induction algorithms for tree-structured data, software package called BACT	Set of weblog posts in four domains: restaurant, automobile, cellular phone and video game	395 weblog posts in restaurant domain from opinion-annotated corpus for experiments
Wilson, Wiebe & Hoffmann (2009)	Automatically distinguish between prior and contextual polarity, with a focus on understanding which features are important for this task	BoosTexter, TiMBL, Ripper, SVM	MPQA corpus with added contextual polarity annotations	19,962 subjective expressions from MPQA annotations
Balahur et al. (2010)	Test relative suitability of various sentiment dictionaries and attempt to separate positive or negative opinion from good or bad news	(often user-defined) Boolean search word expressions or by using lists of search words with varying weights	Sentiment dictionaries (EMM, SentiWordNet, WordNet Affect, MicroWNOp), newspaper articles	1,592 quotes (reported speech) from newspaper articles in English
Pak & Paroubek (2010)	Show how to automatically collect corpus for sentiment analysis from Twitter and build document-level sentiment classifier for 'positive', 'negative' and 'neutral' sentiments	Multinomial NB, SVM, CRF	Corpus gathered from Twitter	300,000 very short English text posts from Twitter
Rentoumi et al. (2010)	Improving sentiment analysis for figurative language by joining machine learning and rule-based methods	HMMs, FigML (machine learning), PolArt (rule-based)	Affective Text Task of SemEval '07	1,000 polarity annotated headlines

Table 3. Overview of some previous work since 2006.

From Tables 1 to 3 we can make up that most of the previous research has focused on particular aspects of the opinion mining problem, e.g. research with subjective words or adjectives. In this study, we propose a more general and basic approach to the problem: by introducing a new dataset with user-generated Dutch messages posted on a social networking website (the Dutch Netlog Corpus) and employing straightforward feature sets (most frequent and most informative words of the corpus).

Two of the three classifiers used in this study, the Naïve Bayes and Maximum Entropy classifiers, occur frequently in the tables above. Research from Boiy et al. (2007) proves that while the ME classifier yields better results, it might be advantageous to use the NB classifier, because it is considerably faster. The third classifier we use, the Decision Tree classifier, does not occur in these tables but has proven to be a useful classifier in other domains (e.g. part-of-speech tagging).

1.5. Goals of this Study

The first objective of this research is to create an annotated corpus comprising at least 5,000 short messages (minimum 5 and maximum 50 words) posted on the social networking website Netlog. These messages contain only direct speech; reported speech and quotations (i.e. text with a nested source) are discarded from the corpus. For a complete overview of what is included in and excluded from the corpus, see Section 2 and the annotation manual in Appendix.

The second objective is to determine what classifiers and features produce the best results for this particular sentiment classification task. We classify the DNC data by experimenting with three supervised classifiers (i.e. Naïve Bayes, Maximum Entropy and Decision Trees); they will determine whether messages are 'positive', 'negative', 'both', 'neutral' or trash ('n/a'). In addition, we will also report on classification results for messages tagged on the performance ('standard', 'dialect' or 'n/a') and chat ('chat', 'non-chat' or 'n/a') levels of the DNC data. The experiments are executed in two stages: the first stage uses feature sets containing the most frequent words of the corpus, and the second stage uses feature sets with the most informative words of the corpus, as measured using an information-theoretic approach (see Section 4.2.1). We hypothesize that dialect and chat uses of Dutch will be more difficult to classify (i.e. produce lower accuracy results) than the standard and non-chat data. Standard and non-chat language is mostly uniform, which means that it uses the same orthography for a word every single time. Dialect and non-chat language on the other hand are anything but uniform. Another hypothesis is that the first approach (i.e. with feature sets consisting of most frequent words) will produce lower results than the second approach (i.e. with feature sets containing the most informative words), since a frequent word is not necessarily a relevant feature for opinion classification.

Section 2
The Dutch Netlog Corpus (DNC)

Supervised machine learning (see Section 3.2.3.1) for opinion classification tasks requires an annotated corpus to train and test a classifier. Available corpora are for example product reviews (e.g. digital camera's, cars, movies; for instance Blitzer's corpus (Blitzer et al. 2007)), movie reviews (Pang & Lee 2002), the Wall Street Journal corpus, etc. These corpora are useful for research on sentiment analysis and classification, but in this context, we thought it would be more instructive to construct our own corpus with a language and medium that have not been used frequently for this purpose (i.e. (colloquial) Dutch and user posts and comments on Netlog).

In the DNC, we gather 5,500 manually annotated short messages and comments, composed of minimum 5 to maximum 50 words, that are uttered on the social networking website Netlog.

In this section, we will first discuss the source of the data: the social networking website Netlog. Next, the concepts of subjectivity and polarity are clarified. In the third section, the process of annotating the corpus is discussed, and in the fourth section, the annotator agreement study is reported. We bring this section to an end with some notes on text classification and chat language.

2.1. Netlog

Netlog is a Ghent-based social networking website that was founded in 2003 by Lorenz Bogaerts and Toon Coppens. Its primary users are aged 10 to 16, and content on Netlog is free for every subscriber to see and comment on.

Netlog has approximately 67,916,961 users worldwide[14], and supports more than 25 languages, with the highest number of Spanish users, followed by users who speak English, French, Turkish and Portuguese. Dutch has 3,187,392 users and is the 8th most used language on Netlog[15]. Google AdPlanner estimates that Netlog reaches 12.2% surfers in Belgium, and 1% worldwide[16] (to compare: Facebook reaches 51.1% Belgian and 34.3% worldwide users[17]).

We thank Netlog for supplying the raw data needed for constructing this corpus, and Mollom (especially Benjamin Schrauwen) for making this possible.

[14] nl.netlog.com/go/about/statistics [2010-07-26]
[15] Idem
[16] www.google.com/adplanner [2010-07-26]
[17] Idem

2.2. Data Distribution

Of the 5,500 messages, Netlog provided us with metadata of 4,563 users (82.96% of the data). This metadata includes the gender, age, country and region of the users.

Concerning the gender of the users, we determine that there are more female users (59.5%) than male users (40.5%). Age-wise, the majority of the users in the corpus are teenagers (72.2% of the users are younger than 20 years old), while only a small part of the users is older than 20 (24.8%). For a more detailed distribution of the age of the DNC users, see Figure 1.

The messages we collected are (supposed to be) written in Dutch, so the provenance of most users will be in Dutch-speaking countries. Almost all of the users in the corpus (91.5%) listed Belgium as their country of origin, while a minority is from the Netherlands (7.1%). A small percentage of the users (1.5%) listed another country as their home. Among these countries are the United States, Turkey, Italy, the United Kingdom, Canada, Cuba, France, etc. Because we focus on Dutch-speaking countries and take the different dialects in account in this corpus, we looked at the Belgian and Dutch regions that the users of the DNC listed as their region of origin. Most of the Belgian authors originate from Antwerp (28%), East-Flanders (22.4%) or West-Flanders (19.7%). Most of the Dutch authors come from North-Holland (1.1%), South-Holland (1.1%) or North-Brabant (1.1%). For a full overview of the regions, see Figures 2 and 3.

Figure 1: Age distribution of DNC

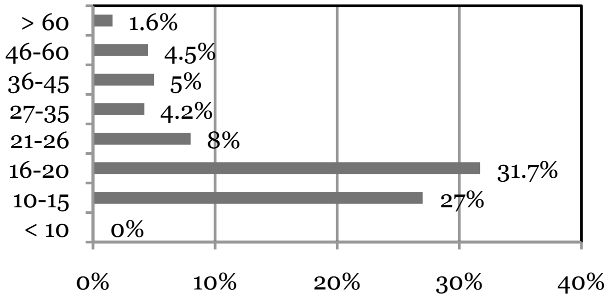


Figure 2: Belgian region distribution of the DNC

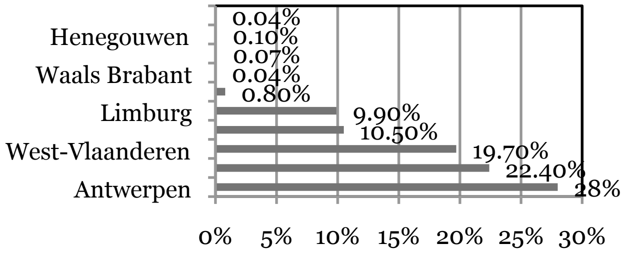
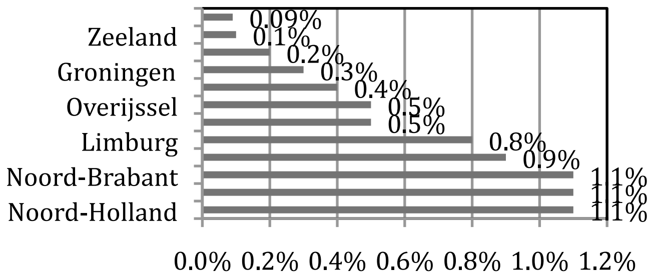


Figure 3: Dutch region distribution of the DNC



2.3. Subjectivity and Polarity

Before discussing the annotation process of the corpus, a note on subjectivity and polarity is in order. These two concepts are similar, but far from the same. Both are used to conduct the annotation in the corpus. Short messages mined from a social networking website are not always subjective, unlike customer reviews that (almost) always express an opinion about a product, and that therefore need no document subjectivity determination before deciding on polarity (e.g. Gamon 2004: 841). Hence, it is useful for sentiment classification with our corpus to also determine document subjectivity, instead of only document polarity. There is also a third subtask of sentiment analysis, determining the strength of document polarity or orientation, which will be discussed below, but is not used in this study.

2.3.1. Determining Document Subjectivity

Determining the subjectivity of a document is about determining whether it has a factual nature (i.e. it describes a given situation or event without expressing a positive or negative opinion) or expresses an opinion on its subject matter (Esuli & Sebastiani 2006a: 193). This leads to binary categorization into the categories ‘objective’ and ‘subjective’.

In our corpus, ‘objective’ will be translated to ‘neutral’ (i.e. not containing opinions of any kind) and ‘subjective’ will immediately be determined by the presence of document polarity (see Section 2.3.2).

2.3.2. Determining Document Polarity

Document polarity, or document orientation, involves deciding whether the ‘subjective’ text expresses a ‘negative’ or a ‘positive’ opinion on its subject matter (Esuli & Sebastiani 2006a: 193).

Aside from the ‘negative’ and ‘positive’ classes, we also use a ‘both’ class in the corpus, for documents that contain multiple and contrasting negative and positive opinions.

2.3.3. Determining the Strength of Document Polarity

Deciding on the strength of a document is determining whether the ‘positive’ opinion expressed by a text is ‘weakly positive’, ‘mildly positive’ or ‘strongly positive’ (Esuli & Sebastiani 2006a: 193). This also applies for ‘negative’ opinions.

Since the current categorization into five classes is already a difficult task for current classification techniques, this subtask of opinion mining is not annotated in the corpus.

2.4. Annotating the Corpus

The DNC is annotated with three different kinds of tags: one denoting the valence of the text and two describing the language performance of the user (standard versus dialect Dutch and chat versus non-chat Dutch). The latter distinctions are made because the language use of Netlog users is often highly colloquial. Previous work has not focused on classification with chat language before, so we thought it would be useful to distinguish between standard and dialect, and chat and not-chat language use in the corpus. With this subdivision, we can check our hypothesis about whether opinion classification is easier for non-chat and standard Dutch than for chat and dialect Dutch.

2.4.1. What is Considered Trash in the Corpus?

Since the corpus deals with expressions of opinion, only directly stated speech is allowed. In other words, messages with any kind of nested source or reported speech (i.e. not expressed by the author himself) are excluded (tag ‘not applicable’ or ‘n/a’).

A next step is asking the question “How does the writer feel about someone/something?” If this question is not applicable to or not answerable by the text, it does not have opinionated content and is regarded as being factual or objective (tag ‘neutral’).

Some messages comprise only words in other languages than Dutch (due to a language classification error), and these messages are discarded (tag ‘n/a’). However, if at least 50% of the words in a message are Dutch words, it is allowed. Using foreign languages (usually English) is a natural process in the evolution of (chat and dialect) language, so it cannot be entirely ignored. In the same vein, if more than 50% of the characters of a text are emoticons or punctuation marks, it is excluded (tag ‘n/a’).

One last category of messages that is excluded from the corpus is incomprehensible messages (whether the cause be extreme dialect or extreme chat language, tag ‘n/a’).

2.4.2. Valence Tags

For tags describing the valence of a message, we use five different categories: ‘positive’, ‘negative’, ‘both’, ‘neutral’ and ‘not applicable’.

2.4.2.1. Neutral or No Valence

Neutral or objective messages display no opinion(s) toward someone or something, neither positive nor negative, and are therefore tagged as **'neutral'**.

These messages consist of general information or facts like announcements or statements, e.g. a meeting place and/or time, a to do list, information about the user, random conversation, etc. The question "How does the writer feel about someone/something?" applied to these messages can therefore not be answered.
Wishing somebody a happy birthday, merry Christmas, happy new year, etc. or welcoming someone is considered to be objective conversation.

2.4.2.2. Positive valence

Subjective messages that provide a positive answer to the question "How does the writer feel about someone/something?" are tagged **'positive**'. This includes expressions of friendship, love, joy, excitement, etc. Usually, positive messages are not hard to detect because of typical semantic expressions ("Ik hou van jou", "Wat is het mooi", etc.). Sometimes however, words with a positive value can form a message that is actually neutral or even negative toward the topic discussed. This is why the validating question "How does the writer feel about someone/something?" is relevant.

2.4.2.3. Negative valence

A negative answer to the question "How does the writer feel about someone/something?" leads to the tag **'negative**'. This includes hate, racism, profanity, sadness, bullying, etc. As with positive messages, negative messages are usually not hard to detect because of typical semantic expressions ("Ik haat je", "Wat een lelijke jas", etc.). Sometimes words with a negative value can form a message that is actually neutral or even positive toward the topic discussed.

2.4.2.4. Positive and negative valence

In some cases, both positive and negative opinions are uttered in a single subjective expression. This happens quite frequently; for example, by first saying a good thing about a person followed by a negative comment about someone else or about that same person. The tag **'both'** applies here.

2.4.2.5. Not applicable

The not applicable or **'n/a'** tag is relevant for the previously mentioned cases of reported speech, messages with more than half of the words in foreign language(s) and/or emoticons and punctuation marks, and incomprehensible text messages. This data is considered to be too noisy for successful classification tasks, and is therefore separated from the useful data. However, this category can be very useful for classification if the accuracy results for these messages are high (see Section 5.1.1.2.).

2.4.3. Language performance tags

Since the Internet is a great place to observe the rapid evolution of our language and because its use is largely performance-based, some elementary tags concerning the language performance of Netlog users can be prove to be very useful. This data is especially relevant in combination with information about the age, gender and location of the authors (e.g. computational stylometry, sociolinguistics). There are two dimensions in this category of tags and therefore two phases in the annotation process: standard versus dialect Dutch and chat versus non-chat language.

Language on the Internet makes use of a specific vocabulary (e.g. “overmooi”) and writing style or orthography (e.g. “fotoow”) that varies greatly between users (e.g. subcultures, age and peer groups, etc.) or even between users themselves (e.g. depending on mood). This phenomenon is often referred to as youth language and is a hybrid of dialect (vocabulary) and chat (orthography) characteristics. Messages exhibiting these words or word forms are tagged dialect in the first phase and chat in the second phase.

Messages that were tagged as ‘n/a’ in the first annotation phase, will also receive an ‘n/a’ tag in both of the annotation stages for language performance.

2.4.3.1. Standard versus non-standard Dutch

In the first phase of the annotation process of language performance tags, we aim to determine whether a text message is written in standard or non-standard Dutch.

Standard Dutch refers to messages that are written in proper Dutch; this means that spelling, grammar and orthography are consistent with the standard conventions and formal rules of Dutch (i.e. Het Groene Boekje, Van Dale woordenboek, etc.). However, a minor spelling error that occurs frequently and often unconsciously in Dutch (e.g. dt-errors), or in any language (e.g. typo), is ignored. However, when more than 25% of the words in the text are incorrectly spelled, or when the same error occurs twice, or when obvious deliberate spelling errors are made (e.g. frequent n-deletion at the end of verbs or deletion of characters in words), the text is categorized as ‘non-standard’ or ‘dialect Dutch’ (e.g. “wa” instead of “wat” and “lope” instead of “lopen”) or sometimes even as chat (see below).

Non-standard or **dialect Dutch** is inconsistent with standard Dutch (e.g. using “gij” instead of “jij” or “tis” instead of “het is”) and is distinguished by its vocabulary, grammar, orthography and pronunciation. Dialect can be geographically (regiolect) or socially (sociolect) different from standard Dutch. Regiolects in Flemish Dutch are roughly *West-Vlaams, Oost-Vlaams, Antwerps, Limburgs* and *Brussels*. Regiolects in Holland Dutch are numerous and comprise among many others *Zeeuws, Fries, Twents,* and *Gronings*. An example of a regiolect is “k zin ik” [West-Vlaams] versus the standard Dutch form “ik ben”. Sociolects can be ‘caused’ by differences in social status, education/occupation, ethnicity, gender, or age; for example the use of jargon (education/occupation), slang (social status, age), youth language (age) and certain slight differences in vocabulary and grammar (gender, e.g. research in computational stylometry). A text message is considered to be dialect if even only one word in the message is dialect (e.g. “gij” instead of “jij” or “drij” instead of “drie”).

In the rest of this study, the annotation process between standard and dialect messages will be referred to as the ‘performance’ level.

2.4.3.2. Chat versus non-chat language

The second phase of the language performance tagging process distinguishes between chat and non-chat language.

Chat language takes on many forms and is very distinct from non-chat language. It transforms dialect and standard Dutch by writing in a more phonetic way, by deleting characters or adding repetitive ones, by not using any punctuation marks or exceedingly much of them, by using emoticons or certain abbreviations (e.g. “omg” for “oh my god”, “btw” for “by the way”), by using a wide range of orthographic varieties, etc.

Non-chat language conforms to the writing style of standard formal Dutch, which means not using excessive punctuation marks and capitalization, not using emoticons, etc. Only one occurrence of the above mentioned characteristics of chat language is not enough to label a text as chat language, because this could be a typo or a more general way of expressing opinions through symbols (e.g. using an extra question mark when surprised). At least two occurrences are necessary (e.g. a contraction such as “kzie” instead of “ik zie” combined with the adding of repetitive characters such as “jeeee” instead of “je”).

In the rest of this study, the annotation process between standard and dialect messages will be referred to as the ‘chat’ level.

2.4.4. Tag codes

Writing tag codes instead of entire tag names, such as ‘positive’ or ‘negative’, is very timesaving. Table 4 shows the tag codes for valence tags and Table 5 presents the tag codes used for the language performance tags.

Tag code	Tag name
+	positive
-	negative
&	both
_	neutral
/	n/a

Table 4. Tag codes for valence tags.

Tag code	Tag name
s	standard
d	dialect
c	chat
n	non-chat
/	n/a

Table 5. Tag codes for language performance tags.

For more detailed information and some examples of the tagged data, see the annotation manual in that is included in Section 9.

2.5. Inter-Annotator Agreement Study

To validate the accuracy of the annotation process as well as the explicitness and completeness of the annotation guidelines (see Appendix), the agreement between two annotators has been calculated using Cohen's kappa coefficient (a statistical measure for inter-annotator agreement which takes chance agreement into account). Two annotators (Claudia Peersman and the author of this report) labelled the same set of 100 Netlog messages with the tags described above.

The Percentage of Agreement (PA) on message valence is 86%, while Cohen's kappa statistics (*k*) results in 0.79 or 79%. The latter percentage is lower than the former one, because the Percentage of agreement Expected by chance (PE) is taken into account in Cohen's kappa coefficient. The PA for chat versus non-chat tags is 90% and *k* is 0.66 or 66%. Tagging of standard versus non-standard messages results in a PA of 87% and *k* of 0.71 or 71%.

	PA	PE	*k*
Valence	86%	34%	**0.79**
Standard vs. dialect	87%	55%	**0.71**
Chat vs. non-chat	90%	71%	**0.66**

Table 6. Annotator agreement for annotating the DNC.

Table 6 shows that the PA of chat versus non-chat messages is considerably higher than that of valence or standard versus dialect messages, which can be explained by the large number of messages tagged as 'chat' and the relatively few messages tagged as 'non-chat' (see Section 2.6). The PE of valence is much lower that those of the other categories, because there are five possible tags for valence (instead of only three tags for the other categories). However, this low PE for valence does result in the highest percentage of *k*. It is clear that the annotators agreed very often on whether messages were positive, negative, both, neutral or n/a, but seem to have different opinions as to whether messages contain chat or non-chat language. It is also possible that the annotation manual is not yet optimal for annotating chat versus non-chat.

In the literature, a *k* of approximately 80% is considered to represent highly reliable annotator agreement. We can therefore say that the inter-annotator agreement for valence is accurate and reliable. The k of chat versus non-chat and standard versus dialect are not as high, but still high enough to be acceptable and useful.

2.6. Class Distribution

How much of the data in the DNC is subjective, and how much is objective? How much is considered to be trash? Is 'standard' language use more congruent with 'non-chat' or with 'chat'? This section will give an answer to these questions, and more, in an overview of the class distribution of the DNC.

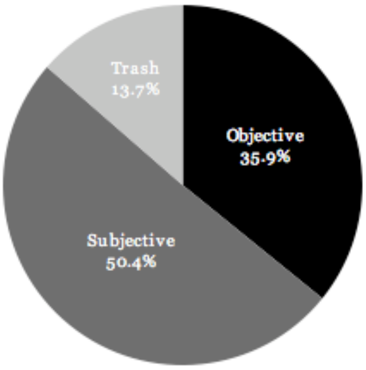


Figure 4. Class distribution of the DNC

Figure 4 shows that half of the data in the DNC is tagged as being subjective (i.e. 'positive', 'negative' and 'both'; 50.4%), while 35.9% is tagged as being objective (i.e. 'neutral') and 13.7% is considered trash (i.e. 'n/a'). The latter means that more than one message in ten was not useful for the corpus, because it comprised only foreign language, or too much punctuation marks, emoticons, etc. This class is built to reduce the amount of noisy data in the corpus, but it can be useful for classification if the classifier accuracy is high (see Section 5).

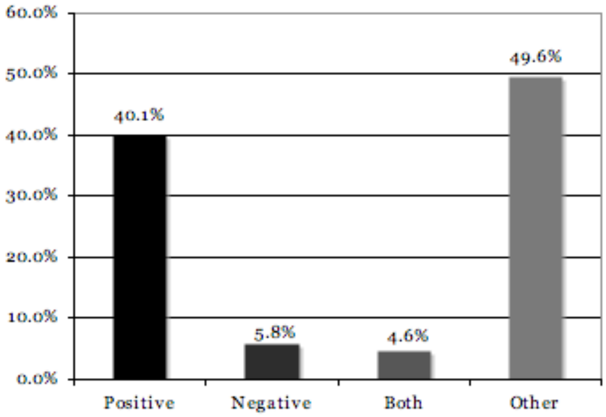


Figure 5. Class distribution of 'valence' class in the DNC

Looking at subjective valence tags in Figure 5, we see that 'positive' messages are the most frequent (40.1%). 'Negative' tags apply to 5.8% of the data, and 'both' is least frequent with merely 4.6% of the data. It is clear that messages posted on Netlog are mostly positive, and rarely negative. The category 'other' contains the objective messages and noisy data (i.e. 'n/a').

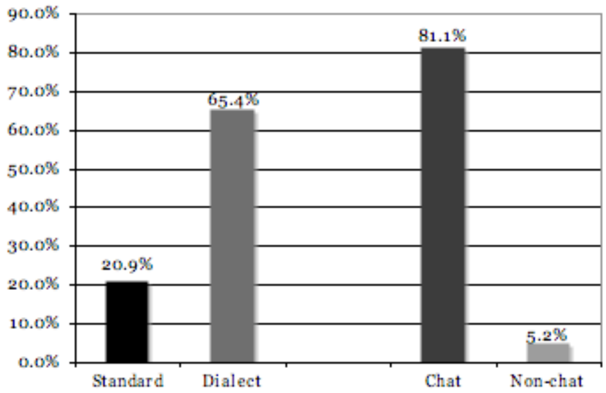


Figure 6. Class distribution of the 'performance' and 'chat' classes of the DNC

Language performance tags were divided into ‘standard’ versus ‘dialect’ and ‘chat’ versus ‘non-chat’. Figure 6 shows that dialect Dutch is very frequent (65.4%) in comparison to standard usages (20.9%). Chat language is present in a staggering 81.1% of the messages, while the label ‘non-chat’ applies to only 5.2% of the data. These percentages are not really surprising: teenagers and adolescents (the prominent users of Netlog) are more prone to using dialect forms of a language, and the Web is, next to text messaging, the biggest source of and outlet for chat language.

Figure 7. Class distribution of ‘positive’ combinations

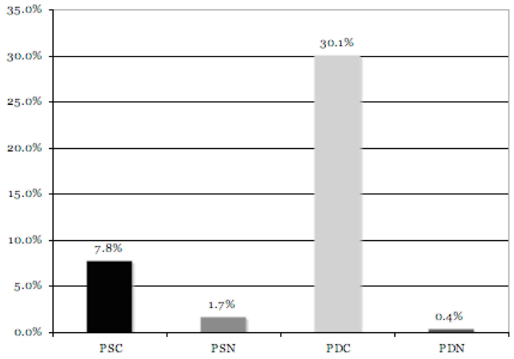


Figure 8. Class distribution of ‘negative’ combinations

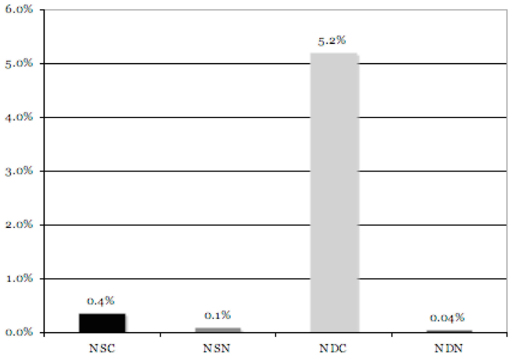


Figure 9. Class distribution of ‘both’ combinations

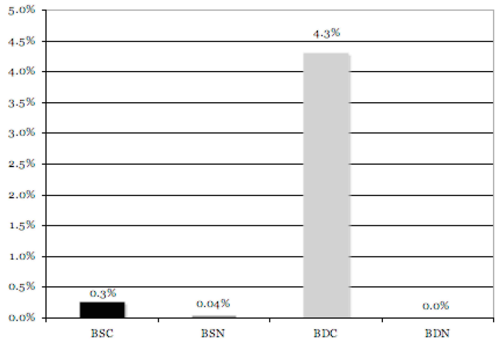
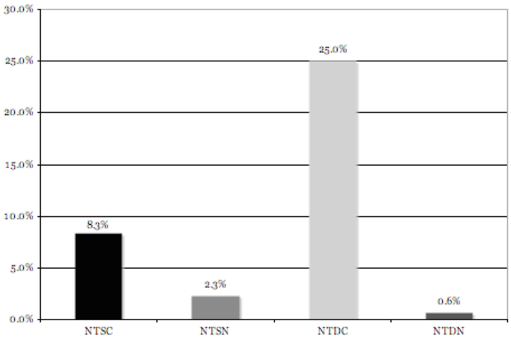


Figure 10. Class distribution of ‘neutral’ combinations



Tag combinations are also very interesting to look at. In Figures 7 to 10, we present the following combinations: positive/standard/chat (PSC), positive/standard/non-chat (PSN), positive/dialect/chat (PSC), positive/dialect/non-chat (PDN) in Figure 7, and the same partitions for negative (N) in Figure 8, both (B) in Figure 9 and neutral (NT) in Figure 10. All graphs show the same patterns: the co-occurrence of ‘dialect’ and ‘chat’ tags protrude in every graph, which means that it is by far the most frequent combination of tags, followed by the often-occurring combination of ‘standard’ with ‘chat’, and rare co-occurrence of ‘standard’ and ‘non-chat’. The co-occurrence of ‘dialect’ and ‘non-chat’ tags is close to non-existent in every graph. Again, this is not surprising given the subjects and the medium.

2.7. Chat language and text classification

Computational text classification can be more difficult for messages written in chat language, as opposed to texts containing non-chat language. The main reason is that chat language is anything but uniform. Everyone has his or her own writing style and habits, and ‘trends’ among Internet users change every hour. For example, in chat language, words are often written without spaces (e.g. “iloveyou”), in abbreviations (e.g. “BFF NWLY”), with deleted (e.g. “kzn u mrg”) or added (e.g. “tommyyyyyy gaann we morge ietss ete???”) characters and/or punctuation marks, using a wide variety of emoticons (e.g. “:)”, “(:”, “xD”), etc. These forms can vary interpersonally and even intrapersonally, depending on the mood of the author, peer group, forum, etc. Furthermore, capitalization is often varied as well. Normally, chat language can be distinguished from non-chat language in the lack of using capitalizations, but it has become a trend to either vary capitalizations (e.g. “mOoOoiIiI”) or to write words entirely in capitals (e.g. “MOOOOI IS DAT”). In classification, chat language will either have no capitals at all, or more than usual.
There is not one way of writing something (e.g. “mooooi”, “mOoOoiIiI”) and all these variations in orthography can refer to the same meaning of a word, or not. This variety of occurring styles makes it harder for the classifier to find helpful features in messages: frequency distribution will not be as detailed as with standard Dutch (e.g. the occurrence of the word “eigenlijk” in two messages will not be added to the frequency with an orthography such as “eigelijk” or “eiglijk”), and lemmatization is not possible for non-standard and chat Dutch (e.g. “bloempje” and “bloemen” are lemmatized to one lemma “bloem”; with chat language forms such as “blmpje” or “bloeme”, it will render incorrect lemmas).

Because of this lack in uniformity, classification will be hindered. A word might be written as “moooi” one time, but as “mOoOoiIiI” the other time, while in standard and non-chat Dutch there are only two formal ways of writing this: “mooi” or “Mooi”. Human annotators or readers know that this word is, aside from the different writing style, one and the same. For a computer, and more specifically for a classifier, this is very far from being the same word. New techniques must be found to get around this problem.

Section 3
Methodology

In general, it is considered that there are two approaches to sentiment analysis: using symbolic methods and using machine learning techniques. Symbolic techniques are based on manually crafted rules and lexica, while machine learning techniques use unsupervised, weakly-supervised and supervised learning (Boiy et al. 2007: 351).

3.1. Symbolic Techniques

As explained, **symbolic techniques** are based on manually crafted rules and lexica. Boiy et al. (2007: 352) argue that symbolic techniques make use of lexicon-based techniques, and they give two examples: using a Web search and using WordNet. Web searches can provide useful statistical information about word frequency, term distribution, co-occurring words, etc. that can be employed as useful features for classifiers. WordNet on the other hand can be used to gather information about links and relationships between words, such as synonymy and antonymy.

3.2. Machine Learning Techniques

Machine learning is a subfield of Artificial Intelligence dealing with algorithms that allow computers to learn. This usually means that an algorithm is given a set of data and subsequently infers information about the properties of the data; that information allows it to make predictions about other data that it might come across in the future (Segaran 2007: 3). The ability to make predictions about unseen data is possible because almost all non-random data contains patterns that allow machines to generalize (Segaran 2007: 3). In order to generalize, the computer trains a *model* with what it determines are the important aspects of the data (Segaran 2007: 3).

Machine learning does have its weaknesses; the algorithms vary in their ability to generalize over large sets of patterns, and a pattern that is unlike any seen by the algorithm before is quite likely to be misinterpreted (Segaran 2007: 4). In language, frequently occurring patterns are rare, and rarely occurring patterns are predominant; this makes that machine learning methods can only (limitedly) generalize based on the information that they have already seen (Segaran 2007: 4), while humans have a large world knowledge base that supplies them with countless training data and possibilities for feature construction.

3.2.1. Features

The **bag-of-words** approach is the simplest representation of a text. A bag-of-words is an unordered set of words, with their exact position ignored (Jurafsky & Martin 2009: 675). Bag-of-word features are effective at capturing the general topic of the discourse in which the target word has occurred (Jurafsky & Martin 2009: 675). Stop words (e.g. "dus" and "of zo" in Dutch), which intuitively seem statistically irrelevant, are often filtered out, but are nevertheless sometimes very helpful in text categorization tasks (e.g. in computational stylometry). In this study, we use two bags-of-words.

***N*-grams** are another classic approach to collecting features. *N*-gram features provide the ability to identify *n*-word (idiomatic) expressions (e.g. "United States of America", "picture perfect"), and can provide more context to a sentence or document, which can be useful for word sense ambiguation (Boiy et al. 2007: 353). Features of *n*-grams with more than three words will be less useful without gigantic amounts of data[18], since these co-occurrences will be less likely to be found in the same document or others. *Character* n-grams have proven to be even more useful than *word* n-grams, for example in language classification.

Lemmatization is another helpful feature selection process, since various forms of one word are considered to be one lemma (e.g. the lemma of "running" and "ran" is "run"). This way, the features are generalized and classifying new documents will be easier (Boiy et al. 2007: 353).

Detecting **negation** in text can also prove helpful for classification tasks. For a sentence like "I didn't like that book by Aldous Huxley", a learner that is looking only at words or *n*-grams can think that the author liked this book (e.g. only paying attention to the word "like"). A solution for this problem is tagging each word after the negation until the first punctuation (e.g. with "/NOT") (Boiy et al. 2007: 353). A more advanced approach to negation detection is described in Morante & Daelemans (2009).

Several researchers have worked with lists of particular words: **opinion words**, subjective **adjectives and/or adverbs**, etc. They argue that these words are good markers for a category (e.g. the opinion words "good", "beautiful", "darling" are markers for 'positive' and "bad", "awful", "savage" are markers for 'negative').

Words are not the only useful features: paying attention to usage of (excessive) **punctuation marks** (e.g. exclamation marks for aggressive attitude and potentially marker for 'negative', question marks for surprise or inquisitive attitude and potentially informative for 'neutral'), (excessive) **capitalization** (e.g. full capitalization can be marker for 'chat' or 'negative', while no capitalization can also be marker for 'chat') and **emoticons** (e.g. ":-)" is a possible marker for 'positive' and ":-(" for 'negative', and both are good markers for 'chat') can also produce useful features for classification. Pak & Paroubek (2010), for instance, collect positive and negative messages posted on Twitter on the basis of happy (":)", "=)", etc.) and sad (":(", "=(", etc.) emoticons.

Part-of-Speech (POS) tags uncover grammatical structures and provide a deeper linguistic analysis, unlike the more superficial features we mentioned before. *N*-grams of POS tags are also a very useful

[18] For instance, Google has trained word five-grams on all of the available data on the Internet.

features: they reveal grammatical patterns in text. See Section 4.2.1 for more on linguistic analysis features and why we cannot use them in this study.

3.2.2. Feature Selection

The next step after gathering useful features is *feature selection* or *feature subset selection*. It determines which subset of a feature set is the most suitable and profitable for the task at hand; it eliminates features with little or no predictive information. Feature selection can significantly improve the comprehensibility of resulting classifier models (Kim, Street & Menczer 2003: 80). Feature selection in supervised learning aims to find a feature subset that produces higher classification accuracy (Kim, Street & Menczer 2003: 80).

There are three main directions in feature selection: wrappers, filters and embedded methods. **Wrappers** use the learning machine as a black box to score subsets of features according to their predictive power (Guyon & Elisseeff 2003: 1166). Especially greedy search strategies appear to be computationally advantageous and robust against *overfitting* (Guyon & Elisseeff 2003: 1166): too many features can cause an algorithm to have a higher chance of relying on idiosyncrasies of the training data (especially if it is small, which is the case) that do not generalize well to new examples (Bird, Klein & Lopez 2009). There are two methods of greedy search strategies: forward selection and backward elimination. Forward selection progressively incorporates features into larger subsets, whereas in backward elimination the starting point is the set of all features, where the least promising ones are progressively eliminated (Guyon & Elisseeff 2003: 1167). **Filters** select feature subsets as a pre-processing step independently of the chosen predictor (Guyon & Elisseeff 2003: 1166). They are assumed to be faster than wrappers, and provide a generic selection of features that are not tuned for or by a given learning machine. Since it is a pre-processing step, it reduces space dimensionality and overcomes overfitting (Guyon & Elisseeff 2003: 1170). **Embedded methods** select features in the process of training, and are usually specific to given learning machines (Guyon & Elisseeff 2003: 1166). They are considered to be more efficient than wrappers because they make better use of the available data, by not needing to split the training data into a training and validation set (Guyon & Elisseeff 2003: 1167). Decision trees have embedded methods to perform feature selection.

In this study we define two kinds of feature sets, by way of filter selection systems: word frequency and information gain. We will also use a classifier that supports an embedded selection method: the Decision Tree classifier.

3.2.3. Classification Techniques

Machine learning usually distinguishes between three learning methods: supervised, weakly supervised and unsupervised learning. Reinforcement learning is also a machine learning technique, but it is not used for text classification and will therefore not be discussed.

3.2.3.1. Supervised Learning

Supervised machine learning techniques implicate the use of a labelled training corpus to learn a certain classification function (Boiy et al. 2007: 354) and involve learning a function from examples of its inputs and outputs (Russell & Norvig 2003: 650). The output of this function is either a continuous value ('regression') or can predict a category or label of the input object ('classification'). In this study, we will focus only on classification. In this section, we will discuss five supervised learning classifiers: Naïve Bayes, Maximum Entropy, Decision Trees, TiMBL and Support Vector Machines.

A well-known supervised machine learning technique that is often used for classification tasks is a **Naïve Bayes (NB)** classifier. This classifier is called *naïve* because it assumes that the probabilities being combined are independent of each other: the probability of one word in the document being in a specific category is unrelated to the probability of the other words being in that category (Segaran 2007: 124). Calculating the entire document probability is a matter of multiplying all the probabilities of the individual words in that document (Segaran 2007: 124). Bayesian classifiers are often used for document classification because they require far less computing power than other methods (Segaran 2007: 140). According to Jurafsky & Martin (2009: 247), the NB classifier is often used as a good baseline method, with results that are sufficiently good for practical use.

A **Maximum Entropy (ME)** classifier, or conditional exponential classifier, is parameterized by a set of weights that are used to combine the joint-features that are generated from a set of features by an encoding[19]. The encoding maps each pair of feature set and label to a vector. ME classifiers belong to the set of classifiers known as the exponential or log-linear classifiers, because they work by extracting some set of features from the input, combining them linearly (each feature is multiplied by its weight and added up) and then using this sum as exponent (Jurafsky & Martin 2009: 227).

A **Decision Tree (DT)** classifier is a tree in which the internal nodes are labelled by the features, the edges leaving a node are labelled by tests on the feature's weight, and the leaves are labelled by categories (Feldman & Sanger 2007: 72). It categorizes a document by starting at the tree root and moving successfully downward via the branches (whose conditions are satisfied by the document) until a leaf node is reached (Feldman & Sanger 2007: 72-73). The document is then classified in the category that labels the leaf node (Feldman & Sanger 2007: 73). DTs have been used in many applications in speech and language processing (Jurafky & Martin 2009: 247).

TiMBL, or Tilburg Memory-Based Learner, is a classifier based on *k*-Nearest Neighbour (*k*-NN) algorithms and memory-based learning. The *k*-NN algorithm is a classifier based on closest training examples in the feature space, and is a type of instance-based or lazy learning: an object is classified by a majority vote of its neighbours, with the object being predicted as belonging to the class that is most common amongst its *k*-neighbours (*k* is usually a small integer). TiMBL determines the similarity between already seen and new data by looking for cases in memory with the highest overlap in feature values, whereby information gain is used to weigh the relevance of the features. Memory-based learning assumes that in learning a cognitive task from experience, people do not extract rules or abstract representation from their experience, but reuse their memory of that experience directly (Daelemans & van den Bosch 2005: 5). It incorporates two principles: learning is the simple storage of a representation

[19] http://docs.huihoo.com/nltk/0.9.5/api/nltk.classify.maxent.MaxentClassifier-class.html [2010-05-24]

of experiences in memory, and solving a new problem is accomplished by reusing solutions from similar previously solved problems (Daelemans & van den Bosch 2005: 5).

Support Vector Machines (SVMs) are often regarded as the classifier that yields the highest accuracy results in text classification problems. They operate by constructing a hyperplane with maximal Euclidean distance to the closest training examples (Boiy et al. 2007: 354). Simply put, SVMs represent examples as points in space which are mapped to a high-dimensional space where the mapped examples of separate classes are divided by an as wide as possible tangential distance to the hyperplane. New examples are mapped into that same space, and depending on which side of the hyperplane they are positioned, they are predicted to belong to a certain class. SVM hyperplanes are fully determined by a relatively small subset of the training instances, which are called the *support vectors* (Feldman & Sanger 2007: 76). The rest of the training data have no influence on the trained classifier. SVMs have been employed successfully in text classification and in a variety of sequence processing applications (Jurafky & Martin 2009: 247).

Other supervised learning methods include Decision Rule classifiers, (Artificial) Neural Networks, Logistic Regression, Rocchio Methods and Random Forests.

3.2.3.2. Weakly-Supervised and Unsupervised Learning

Supervised methods cannot always be used, because labelled corpora are not always available. Unsupervised and weakly-supervised methods are another option for machine learning that does not require pre-tagged data.

Unsupervised methods involve learning patterns in the input when no specific output values are supplied (Russell & Norvig 2003: 650), this means that the learner only receives an unlabelled set of examples. Unsupervised methods can also be used to label a corpus that can later be used for supervised learning (Boiy et al. 2007: 354). An agent purely based on unsupervised learning cannot lean what to do, because it has no information as to what constitutes a correct action or a desirable state (Russell & Norvig 2003: 650). Examples of unsupervised learning methods are (*k*-means) *clustering* or *cluster analysis*, the problem of discerning multiple categories in a collection of objects (Russell & Norvig 2003: 650) and the *expectation-maximization algorithm*, an algorithm for finding the maximum likelihood of examples.

Weakly-supervised learning, or **semi-supervised learning**, involves learning a classification task from a small set of labelled data and a large pool of unlabelled data (Ng & Cardie 2003: 94).

3.3. Methodology Used in This Study

Symbolic techniques are currently considered to be outdated, and machine learning techniques have proven to be more accurate and user-friendly. In this study, we will employ supervised learning methods. The feature sets and supervised classifier employed in the experiments are discussed in Section 4.

Section 4
Experimental Setup

In this section, we will discuss the experimental setup of this study. First, we will briefly discuss the data set, followed by a description of the feature vectors and sets, and the tools used for programming. Then we will discuss the used evaluation methods. Finally, we compute baselines to verify whether the classifiers actually learned something.

4.1. Data set

The data set used to conduct this report is the manually annotated corpus DNC, discussed in detail in Section 2. We opted to keep the data set quite small, consisting of 5,500 short messages, since previous research has shown that small corpora are able to achieve a high enough F-measure (ca. 66%) to be considered viable for training a subjectivity classifier (Banea et al. 2008).

4.2. Feature Vectors and Sets

4.2.1. Feature Vectors

Previous research has pointed out an improvement of sentiment classification by using **abstract linguistic analysis features**, such as part-of-speech trigrams and constituent specific length measures (Gamon 2004: 842). Implementing these linguistic features could be helpful in this research, but we decided not to use any features based on linguistic analysis, because the data is very noisy: it contains colloquial, dialect and chat Dutch that will not be processed correctly by POS-taggers, lemmatizers, parsers, etc. For example, the standard Dutch sentence “Heb jij gisteren die voetbalwedstrijd gevolgd op de televisie?” in its ‘dialectical’ form “edde gij gistre da matchke gevolgd op den teevee??” will not be parsed correctly since, for instance, the parser will not recognize certain words (e.g. “edde gij” instead of “heb jij” and “den” instead of “de”).

Previous work has reported that in some domains unigrams outperform bigrams (e.g. when performing sentiment classification on movie reviews, Pang, Lee & Vaithyanathan 2002). In other work, this has been countered: Dave et al. (2003) have reported that bigrams and trigrams work better for product review classification. The Dutch Netlog Corpus is a different domain from movie and product reviews, so we opted to use two simple, superficial unigram feature vectors[20]: the most frequent and the most informative words in the corpus.

[20] The short time span also did not allow for more varied feature sets; see Section 7.2 for an overview of proposals for future work.

In the first stage, the **most frequent words** of the corpus are gathered by using the frequency distribution of the words; we will perform experiments with ten variations or subsets of this feature set (see Section 4.2.2). The same experiments will be performed on different feature subsets in the second stage, which comprise the **most informative features** of the corpus: the *entropy* of the distribution of each word over the different output classes is calculated, and the words with the lowest entropy (or highest information gain) are considered the most relevant features for the classifiers. To measure entropy, the frequency distribution of a feature (in this case a word) over the output classes is computed. The more skewed this distribution is, the lower the entropy. On the other hand, the entropy is high if the distribution is even.

4.2.2. Feature Sets

We have created **ten feature subsets** per feature set (most frequent and most informative), which means that there is a **total of 20 feature subsets** used for this study:

- 1,000 most frequent/informative *valence* features
- 500 most frequent/informative *valence* features
- 250 most frequent/informative *valence* features
- 100 most frequent/informative *valence* features
- 1,000 most frequent/informative *valence* features with '*only standard'* data
- 1,000 most frequent/informative *valence* features with '*only dialect'* data
- 1,000 most frequent/informative *valence* features with '*only chat'* data
- 1,000 most frequent/informative *valence* features with '*only non-chat'* data
- 1,000 most frequent/informative *performance* features
- 1,000 most frequent/informative *chat* features

We narrowed down the feature sets to a maximum of 1,000 features to prevent overfitting. Furthermore, all feature sets are binary: for classifying a document, the classifiers check whether the features are present in the corpus and when they are not.

The features for classifying valence are the most relevant for this study, which is why the feature sets for classifying performance and chat are narrowed down to only one set of 1,000 features each, instead of also defining additional feature sets with smaller samples. We also create valence feature sets that contain features extracted from 'limited' data sets (e.g. only messages tagged as 'standard') that we call 'only standard' (OS), 'only dialect' (OD), 'only chat' (OC) or 'only non-chat' (ON). These features have been constructed to validate our hypothesis that standard and non-chat data are easier to classify than dialect or chat data.

4.2.2.1. Most Frequent Words

The feature set consisting of the *n*-most frequent words in the corpus is a simple bag-of-words. These words are gathered by using the FreqDist module of NLTK (see section 4.3), which calculates how often words occur in the corpus.

Table 7 presents the ten most frequent features for valence, performance and chat classification (top features overall), and for OS, OD, ON and OC data for valence classification, along with their frequency in the corpus. It should also be noted that the data sets with ‘only standard’ and ‘only non-chat’ are small compared to the others, which means that there is less training data for these two sets (and that the frequency distributions for these features are visibly lower).

Top ten features overall		**Top ten features *only standard* data**		**Top ten features *only dialect* data**		**Top ten features *only non-chat* data**		**Top ten features *only chat* data**	
‘’	2408	**‘je’**	693	**‘’**	1506	**‘’**	167	**‘’**	1825
‘x’	1510	**‘’**	486	**‘x’**	1164	**‘je’**	167	**‘ik’**	1300
‘ik’	1506	**‘ik’**	467	**‘ik’**	949	**‘ik’**	116	**‘x’**	1267
‘je’	1498	**‘een’**	386	**‘d’**	930	**‘een’**	93	**‘je’**	1233
‘d’	1210	**‘en’**	305	**‘da’**	834	**‘en’**	71	**‘d’**	1018
‘en’	1114	**‘het’**	219	**‘p’**	804	**‘het’**	66	**‘en’**	951
‘p’	1052	**‘de’**	208	**‘is’**	757	**‘de’**	56	**‘is’**	897
‘is’	1035	**‘van’**	193	**‘en’**	717	**‘is’**	49	**‘p’**	889
‘een’	965	**‘is’**	189	**‘je’**	707	**‘dat’**	42	**‘da’**	838
‘da’	885	**‘met’**	166	**‘u’**	589	**‘van’**	42	**‘een’**	792

Table 7. Top ten most frequent features with their frequency in the DNC.

Table 7 shows that the most frequent words in the corpus are function words such as pronouns (i.e. ‘ik’ and ‘je’), articles (i.e. ‘een’, ‘de’ and ‘het’) and conjunctions (i.e. ‘en’), but also single letters (i.e. ‘x’, ‘p’ and ‘d’). In all sets, the features ‘ik’ and ‘x’ occur in the first three most frequent words. The single letters ‘p’ and ‘d’ can be explained: we have split the corpus on non-alphabetical letters, which means that these single letters were once part of an emoticon (e.g. “:D”, “:p”). Emoticons can prove to be valuable features, especially for a corpus with chat language, but we have chosen to use just word unigrams. However, the top (or second) feature for every feature set is an empty string. They are very frequent[21], but uninformative, so they probably do not effect classifier accuracy.

4.2.2.2. Most Informative Words

Since we know that a frequent word does not necessarily make a relevant word for the learners, we use the 1,000, 500, 250 and 100 features with the highest information gain to form the feature sets for the second stage of the experimentation.

[21] They are caused by the inherent Python re.split() module, which results in empty strings if the capturing groups in the separator match the start or the end of the string (see docs.python.org/library/re.html). This problem is dealt with in further experiments.

Top ten features *valence* data	**Top ten features *performance* data**	**Top ten features *chat* data**
‘schoonheid’	‘eigelijk’	‘ale’
‘tel’	‘moogt’	‘laura’
‘sgone’	‘zukke’	‘eigelijk’
‘___________’	‘aja’	‘moogt’
‘mevrouw’	‘wss’	‘vorig’
‘moooi’	‘tel’	‘aub’
‘___’	‘jhuaa’	‘cava’
‘______’	‘nene’	‘zukke’
‘____’	‘meu’	‘altyd’
‘_______’	‘sgone’	‘aja’

Table 8. Top ten most informative features for ‘valence’, ‘performance’ and ‘chat’ data.

Top ten features *only standard* data	**Top ten features *only dialect* data**	**Top ten features *only non-chat* data**	**Top ten features *only chat* data**
‘welkom’	‘mevrouw’	‘tel’	‘schoonheid’
‘tel’	‘sms’	‘vrienden’	‘sgone’
‘knap’	‘mooiie’	‘mooie’	‘mevrouw’
‘langs’	‘sgone’	‘wensen’	‘moooi’
‘prachtig’	‘knappe’	‘kan’	‘bericht’
‘dame’	‘lieve’	‘verjaardag’	‘sms’
‘gewenst’	‘prachtig’	‘liefs’	‘donderdag’
‘fotos’	‘alvast’	‘gelukkige’	‘mooiie’
‘knappe’	‘khou’	‘alle’	‘nummer’
‘prachtige	‘lieveke’	‘erg’	‘mooiste’

Table 9. Top ten most informative features extracted from ‘only standard’, ‘only dialect’, ‘only non-chat’ and ‘only chat’ data.

All features presented in Tables 8 and 9 have entropy values of -0.0, which means that there is no hierarchy in the columns (except for non-chat, which is a small dataset). The top features for valence show that despite of the built-in ‘trash’ category, there is still a lot of noise in the data: half of the top ten features are a number of underscores (because we slit on non-alphabetical characters, which does not include the number 0 to 9 and the underscore). Content extracted from social networking websites is bound to be noisy, and even careful annotation cannot always resolve this. However, it is clear that Table 9 has much ‘cleaner’ features than Table 8: the limited data feature sets are clearly an interesting subject for future work (see Section 7.2).

4.3. Programming Tools

Three supervised techniques will be used for conducting the experiments: the Naïve Bayes, Maximum Entropy, and Decision Tree classifiers. We will test each technique individually and evaluate its performance. The procedure is, as is standard in supervised machine learning tasks, first training a classifier on pre-classified training data and then evaluating the performance of the classifier on a held-out set of test data (Gamon 2004: 842).

For its ease of use, we opted to work with the Natural Language ToolKit (NLTK)[22]. This package is equipped with several classifiers (i.e. NB, ME, DT and the Weka classifier library). The associated book by Bird, Klein and Loper (2009) served as a very useful guideline for implementing the classifiers.

All programming has been done in the Python[23] programming language and executed in the programming environment Eclipse[24] or Mac's Unix environment Terminal.

4.4. Evaluation Methods

***N*-fold cross-validation** is a reliable accuracy measurement approach used in the majority of computational linguistics research. It performs *n*-separate experiments. In the case of ten fold cross-validation, the data is subdivided into ten parts: 90% of the data is used for training and 10% is used for testing. This is repeated ten times to ensure that all available data is used as test data. By subsequently taking the average or mean of the ten measurements, a more reliable estimate of accuracy can be computed. All the results presented in Section 5 are obtained by calculating the mean of the ten folds.

To evaluate the performance of the different classifiers, the **accuracy** of each separate classifier is computed. Accuracy measures the percentage of input in the test set that the classifier has labelled correctly. Furthermore, the precision and recall are calculated. **Precision** is the number of true positives divided by the total number of elements labelled as belonging to that class. A high precision means that the majority of items labelled as for instance 'positive' indeed belong to the class 'positive'. **Recall** is the number of true positives divided by the total number of items that actually belong to that class. A high recall means that the majority of the 'positive' items were labelled as belonging to the class 'positive'. The **f-measure** or f-score combines the precision and recall to give a single score, and is defined to be the harmonic mean of the precision and recall (Bird, Klein & Lopez 2009):

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

4.5. Baselines

Baselines are a way of checking whether a classifier has actually learned something, or whether he just 'guessed' a category for the input. Generally, researchers work with three kinds of baselines: the majority, random and upper baselines. In the DNC, there are three annotation levels: valence, performance and chat. For each of these levels, the baselines have been calculated.

4.5.1. Majority Baseline

The majority baseline accuracy is the probability of the most frequently occurring class. This means that the majority baseline classifier would classify every test item as being the most frequent class.

[22] www.nltk.org [2010-05-06]
[23] www.python.org [2010-05-06]
[24] www.eclipse.org [2010-05-22]

In the DNC, there are five valence tags: 'positive', 'negative', 'both', 'neutral' and 'n/a'. 'Positive' is the most frequently occurring class, in 40.1% of the cases. This means that the majority baseline for valence is 40.1%.

	Valence	Performance	Chat
Majority baseline	40.1%	65.4%	81.1%

Table 10. Majority baselines for valence, performance and chat.

The most frequently occurring class on performance level is 'dialect', with 65.4% (as opposed to 20.9% of the standard messages). This makes the majority baseline for performance 65.4%.
The 'chat' label is almost omnipresent on the chat level with 81.1% (as opposed to 5.2% of the non-chat messages). The majority baseline on this level is 81.1%.

If the classifiers do not produce results higher than the majority baseline, then it is possible that they did learn something, but that this is not enough to be valuable.

4.5.2. Random Baseline

The random baseline is calculated by adding up the squared probabilities of all the classes. The random baseline classifier thus picks a class at random, instead of choosing the most frequent one.

For example, on valence level this results in the following equation:

$$0.401^2 + 0.058^2 + 0.046^2 + 0.36^2 + 0.137^2 = 0.31465$$

The valence-level random baseline is 31.5%. The random baseline for performance is 47.1% and 66% for chat level.

	Valence	Performance	Chat
Random baseline	31.5%	47.1%	66%

Table 11. Random baselines for valence, performance and chat.

If the classifier does not perform better than the random baseline, this means that it has learned nothing at all, and randomly predicts categories.

4.5.3. Upper Baseline

As upper baseline for each annotation level, we took the kappa coefficient calculated for that level in Section 2.5 (Table 6). This means that the upper baseline is just an indication of what would be a reasonable expectation of maximum accuracy, and of how much noise would be expected in the data if an equal amount of data were annotated by the different annotators. In our case, the annotation is highly homogeneous: the author of this report annotated 90.9% of the data, while only 9.1% of the data is annotated by the second annotator.

	Valence	Performance	Chat
Upper baseline	79%	71%	66%

Table 12. Upper baselines for valence, performance and chat.

At the valence level, the *k* value or the upper baseline is 79%. At performance level, the upper baseline is 71% and at chat level, it is 66%. Compared to the other two baseline percentages, the 66% for chat level is low, which indicates that the human annotators disagree in a fair amount of these cases. This can be caused by the subjective nature of the task, or by the guidelines not being optimal or detailed enough. If a classifier outperforms this baseline, it means that it performs better than the human annotators, which is very impressive.

Section 5
Results

This section presents the results obtained by experimenting in two phases, with two different kinds of feature sets (the most frequent words and the most informative words) and three different supervised classifiers (Naïve Bayes, Maximum Entropy and Decision Trees).

As explained in Section 4.2, the two different kinds of feature sets both consist of ten feature sets: four valence feature sets (the 1,000, 500, 250 and 100 most frequent or informative words in the corpus), four valence feature sets extracted from limited data sets (the 1,000 most frequent or informative words from 'only standard', 'only dialect', 'only chat' or 'only non-chat' data), one chat feature set (the 1,000 most frequent or informative words) and one performance feature set (the 1,000 most frequent or informative words).

In each stage, the results will be presented in three levels: valence, performance and chat classification. We discuss the accuracy results and perform error analysis by computing recall, precision and f-measures, and conclude at the end of each level. As described in Section 4, all experiments are calculated by taking the mean of the ten folds of the cross-validation.

To differentiate between the two stages, the levels discussed will be numbered, i.e. valence$_1$, performance$_1$, chat$_1$, and valence$_2$, performance$_2$ and chat$_2$.

5.1. Stage One: Most Frequent Words

In the first stage of experimentation, we use a simple bag-of-words approach with the most frequent words of the corpus (or 'bag-of-frequent-words').

5.1.1. Valence$_1$

5.1.1.1. Accuracy

The accuracies of the 'bag-of-frequent-words' approach on the valence$_1$ level are noticeably higher than the majority and random baseline (respectively 40.1% and 31.5%), which suggests that the classifiers have effectively learned something. However, the results do not come close to the upper baseline yet.

If we perform a paired t-test on the ten results (over the ten folds) with the 1,000 most frequent features and achieve a *p* value under 0.05, the difference between the two tested results is regarded to be statistically significant. All differences with majority and random baselines are significant ($p < 0.01$), and all the classifiers perform significantly lower than the upper baseline ($p < 0.01$).

	NB	ME	DT
1,000	**63.5%**	43.5%	56.2%
500	60.8%	61.9%	56.1%
250	61.1%	59.3%	59.2%
100	54%	55.8%	51.3%

Table 13. Valence$_1$ accuracy results.

In Table 13, we present the accuracy results for the four different valence$_1$ feature sets. The NB classifier presents the best overall result, 63.5%, with the feature set of the 1,000 most frequent words. For this feature set, this percentage is significantly higher than the result of the ME classifier ($p < 0.01$), and higher, but not significantly ($p = 0.09$), for the DT classifier. The DT classifier also performs significantly better than the ME classifier ($p < 0.01$). The ME classifier runs second (61.9%) with the feature set of 500 words. The DT classifier presents poor results, with accuracies between 51% and 59%. The worst result, 43.5%, is produced by the ME classifier with the 1,000 words feature set.

The feature set with the highest accuracy mean comprises the 250 most frequent words (mean of 59.9%), followed by the set with the 500 most frequent words (mean of 59.6%).

The three employed classifiers each have different training methods, which means that the feature set delivering the best overall result varies from classifier to classifier. The feature set with the 1,000 most frequent words of the corpus yields the best results for the NB classifier. For the ME classifier, the feature set of the 500 most frequent words delivers the best results, while the feature set of the 250 most frequent words produces the best results for the DT classifier.

One of the hypotheses mentioned in the Section 1, is that messages written in standard and non-chat Dutch are 'easier' to classify than those containing chat and dialect Dutch, i.e. they produce better accuracy results.

	NB	ME	DT
'Only standard'	75.9%	**77.5%**	76.9%
'Only dialect'	66.2%	68.7%	55.3%
'Only non-chat'	75.2%	63.7%	68.3%
'Only chat'	68%	50.2%	58.4%

Table 14. Valence$_1$ accuracy results for 'only standard', 'only dialect', 'only non-chat' and 'only chat' data.

Table 14 shows the results of valence$_1$ classification with four feature sets of the 1,000 most frequent words extracted from the 'only standard' (OS), 'only dialect' (OD), 'only chat' (OC) or 'only non-chat' (ON) data sets. For each classifier, both OS and ON data sets result in higher accuracies than the OD and ON sets. The improvements from OD to OS and OC to ON range from 7.2% to an impressive 21.6% (DT classifier). The data set consisting of OS messages performs even better than the ON data. For OS and OD sets, the ME classifier performs best (with a mean of 72.9%), while the NB classifier delivers the best results for ON and OC messages (mean of 71.6%).

Furthermore, it is notable that the accuracy results for the limited data feature sets are much higher than with the entire data set (we will discuss this in Section 6).

5.1.1.2. Error Analysis

In the error analysis sections, we will discuss precision, recall and f-scores of the performed experiments, based on the results of feature sets with the 1,000 most frequent words.

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
'Positive'	66%	77.2%	**71.2%**	41.9%	97.4%	**58.6%**	64.6%	66.1%	**65.3%**
'Negative'	20.8%	6.9%	10.4%	14.3%	0.3%	0.6%	29.8%	11.4%	16.5%
'Both'	20.8%	68.2%	31.9%	20%	0.4%	0.8%	18.5%	7.5%	10.7%
'Neutral'	67%	68.2%	67.6%	53.9%	7.7%	13.5%	56.6%	72.8%	63.7%
'N/A'	73.7%	61.6%	67.1%	86.1%	9%	16.3%	62.7%	40.2%	49%

Table 15. Precision, recall and f-score of the three classifiers for valence$_1$ classification.

Table 15 presents the precision, recall and f-scores of the three classifiers for classifying valence$_1$ with the corpus (see Section 4.4 for computing an f-score). The *'positive'* class yields mediocre (58.6%) to relatively high (71.2%) f-scores, which are all considerably better than the results of the other classes. The messages in the *'neutral'* class are also handled fairly good, with average precisions and recalls (except for the ME classifier recall). The *'negative'* category is handled very poorly (f-scores from 0.6% to 16.5%), as is the *'both'* category (f-scores from 0.8% to 31.9%), which means that none of the classifiers really learned something useful about these classes. Remarkably, the messages labelled as *'n/a'* obtained the highest precision percentages, which means that the majority of the messages labelled as 'n/a' actually are noisy or 'n/a' messages. In Section 2.4.2.5, we stated that messages tagged as 'n/a' could be very useful for classification and for noise-reduction if the classifiers often correctly label this class. With these high precision results, we could say that this is the case (especially for the NB classifier, which also delivers an acceptable recall percentage).

5.1.1.3. Conclusion

Valence$_1$ categorization yields accuracy results from 43.5% to 77.5%, depending on the feature subset and classifier. The NB classifier produces the best results, followed by the DT classifier. The ME classifier performs poorly. The feature set with the 250 most frequent words of the corpus yields the best mean accuracy results.

In Table 14, we looked at the accuracy results for four feature subsets: each containing features extracted from OS, OD, OC and ON data. We hypothesized that the OS and ON sets would present better accuracy results than the OD and OC data sets. This appears to be correct: the OD set produces a mean accuracy of 63.4% over the three classifiers, while the OS set renders a mean accuracy result of 76.8%. The OC yields a mean accuracy of 58.9% (worse than the OD data), as opposed to the 69.1% mean accuracy produced by the ON set. Not only do the OS and ON messages perform better than the OD and OC sets, the OS features also outperform the ON features.

In comparing the classifiers, we learn that the NB classifier delivers the best overall f-scores. We can therefore say that for valence$_1$ classification, this is the best machine learner. From the poor f-scores of the ME classifier, we can determine that it has not learned anything useful for classifying valence$_1$ at all.

5.1.2. Performance$_1$

Although classification of performance is beyond the actual scope of this study, we do perform some explanatory experiments with the gathered data. We hypothesize that accuracy results will be above majority and random baseline, and close to the upper baseline. It is important to keep in mind that 65.4% of the corpus is tagged as 'dialect', 20.9% as 'standard' and 13.7% as 'n/a'. The feature set consists of the 1,000 most frequent words in the corpus.

5.1.2.1. Accuracy

Performance$_1$ accuracies for every classifier seem to be significantly higher than the majority and random baselines (respectively 65.4% and 47.1%) and the NB classifier seems to be higher than the upper baseline of 71% (except for the ME classifier).

Performing the paired t-test shows that the NB and DT classifiers are significantly more accurate than the majority baseline ($p < 0.01$) and the random baseline ($p < 0.01$). The NB classifier performs significantly better than the upper baseline ($p < 0.01$), which means that this classifier performs very well for classifying performance. The DT classifier does not perform significantly better than the upper baseline ($p = 0.18$). The result of the ME classifier is not significantly higher than the majority baseline ($p = 0.29$), but is significantly higher than the random baseline ($p < 0.01$). The ME classifiers' accuracy is significantly lower than the upper baseline ($p < 0.01$).

	NB	ME	DT
1,000	**77.6%**	66.2%	71.9%

Table 16. Performance$_1$ accuracy results.

The NB classifier yields the best accuracy result (77.6%) with this feature set of 1,000 words. The other classifiers perform less well, but still deliver accuracies above random and majority baselines (66.2% for the ME and 71.9% for the DT classifier).

5.1.2.2. Error Analysis

Table 17 shows that all classifiers produce high f-scores (from 79.6% to 84.1%) for the 'dialect' class. This class is represented by more than half of the data (65.4%), which means that the classifiers have more training data for this category and are enabled to learn more about this class.

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
'Standard'	65.4%	66.6%	66%	72.7%	0.7%	1.4%	58.5%	26.7%	36.7%
'Dialect'	84.3%	83.9%	**84.1%**	66%	99.8%	79.6%	72.5%	92%	81.1%
'N/A'	65%	64.5%	64.8%	82.8%	6.4%	11.9%	71.6%	39%	50.5%

Table 17. Precision, recall and f-score of the three classifiers for performance$_1$ classification.

Compared to the valence$_1$ classification, the precision and recall results are better and more evenly distributed; in particular the f-scores of the NB classifier for the less-frequent classes are good in comparison with the other classifiers. The number of classes might be the cause of this improvement: valence$_1$ classifiers have to predict over five classes, while performance$_1$ classifiers only have to deal with three. The ME classifier again performs very poor for the class with the least amount of data in the corpus, with an f-measure of merely 1.4% for the 'standard' class.

5.1.2.3. Conclusion

The accuracy results for performance$_1$ classification vary from 66.2% to 77.5%. The NB classifier presents the best overall accuracy, the DT classifier comes second with 71.9%, and the ME classifier performs poorly (66.2%). All classifiers perform significantly better than the majority (except for the ME classifier) and the random baseline, and the NB classifier even performs significantly better than the upper baseline.

Again, the NB classifier returns the best overall results for this classification task. The ME and DT classifiers perform better at this task compared to the previous one, but the results are still too low to be reliable.

5.1.3. Chat$_1$

'Chat' versus 'non-chat' classification is also a classification task that is also beyond the scope of this study, but some explanatory tests have been performed. Keep in mind that 81.1% of the corpus is tagged as 'chat', and merely 5.2% of the data is labelled as 'non-chat', while 13.7% is considered 'n/a'. The majority, random and upper baselines are respectively 81.1%, 66% and 66%. The feature set comprises the 1,000 most frequent words in the corpus.

5.1.3.1. Accuracy

From the performed paired t-tests we can see that the NB and DT classifiers perform significantly better than the majority baseline ($p = 0.025$ for NB, $p < 0.01$ for DT), the random baseline ($p < 0.01$) and the upper baseline ($p < 0.01$). The ME classifier performs better than the random and upper baselines, but does not significantly outperform the majority baseline ($p = 0.3$).

	NB	**ME**	**DT**
1,000	82.7%	81.9%	**83.2%**

Table 18. Chat$_1$ accuracy results

The DT classifier produces the best accuracy result for the chat$_1$ classification task, with 83.2%. This is not significantly higher than the result of the NB classifier ($p = 0.19$), but does significantly outperform the ME classifier ($p < 0.05$). The NB classifier in its turn does not perform significantly better than the ME classifier ($p = 0.32$). The NB classifier delivers the second best result, and is 82.7% accurate. The ME classifier presents the poorest result (81.9%).

5.1.3.2. Error Analysis

Table 19 presents the precision, recall and f-scores of the three classifiers for chat$_1$ classification.

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
'Chat'	89.5%	90.1%	89.8%	87%	93.5%	90.1%	85%	97.4%	**90.8%**
'Non-chat'	20.7%	19.9%	20.3%	21%	9.1%	12.7%	17%	0.4%	0.8%
'N/A'	64.8%	63.3%	64%	65.1%	50.7%	57%	71%	35.8%	47.6%

Table 19. Precision, recall and f-score of the three classifiers for chat$_1$ classification.

'Chat' messages are predicted very well by all classifiers, with f-scores around 90%. The classifiers label the 'n/a' data averagely, with f-scores from 47.6% to 64%. However, the data tagged as 'non-chat' does not even get an f-score above 20.3%. This data set is very small (only 5.2% of the corpus), which means that the classifiers do not have much training data for this class.

5.1.3.3. Conclusion

The accuracy results for chat$_1$ classification vary from 81.9% to 83.2%. The DT classifier presents the best overall result, the NB classifier comes second with 82.7%, and the ME classifier yields the lowest result. All of the classifiers perform better than the majority, random and upper baselines, except for the ME classifier, which does not outperform the majority baseline significantly.

It is obvious that the NB classifier, again, trumps the other classifiers performance-wise. Although its f-score for 'non-chat' is poor, it is significantly higher than the other classifiers. It is however obvious that neither of the classifiers have learned anything about classifying 'non-chat' data.

5.1.4. Conclusion

The highest accuracy result produced for valence$_1$ classification is 63.5% (NB classifier). This percentage is not very impressive, which means that the features are not yet optimal for this task. The highest f-score is obtained with the 'positive' class and the NB classifier (71.2%). Performance$_1$ classification yields a maximum accuracy result of 77.6% (NB classifier) and a highest f-score of 84.1% with the NB classifier and the class 'dialect'. Chat$_1$ classification produces a maximum 83.9% accuracy (DT classifier) and 90.8% f-score ('chat' class and DT classifier).

At the valence$_1$ level, we performed additional experiments with features extracted from limited data sets, the 'only standard' (OS), 'only dialect' (OD), 'only chat' (OC) or 'only non-chat' (ON) data sets. We hypothesized that the OS set would perform better than the OD set and that the ON would produce better results than the OC set. Table 14 proves that this is the case: OD data produces a mean accuracy of 63.4% over the three classifiers (almost as high as the best accuracy result), while the set comprising OS data yields a mean accuracy of 76.8%. We reckon that this improvement of approximately 13% is caused by the nature of the data: as opposed to dialect Dutch, standard Dutch is characterized by a formal and uniform grammar, lexicon and orthography. OC data performs even worse than the OD; it produces a mean accuracy of only 58.9%. The data consisting of ON yields a mean accuracy of 69.1%. The accuracy results improve with approximately 10%, which is mostly based on orthographical improvement ('non-chat' is more uniform and does not contain words with deleted or added characters). It is clear that the data tagged as 'standard' delivers the best result of this entire first stage. These results point out that user-generated data, which is gathered from the Internet, is mostly written in dialect and chat language and has a need for adapted features.

Two patterns become clear over the three different classification tasks with the different feature sets and classifiers. Firstly, frequently occurring classes (i.e. 'positive', 'dialect' and 'chat') are very frequently classified correctly, which is also reflected in the high f-scores for these classes; data represented by a minority of the corpus (e.g. 'both', 'standard' and 'non-chat') rarely receives the correct label, which is reflected in low f-scores. Although most accuracy results are significantly higher than the majority and random baselines, which implies that the learners effectively learn to classify on the basis of the feature sets, this first pattern suggests that the classifiers still opt the most frequent classes (represented by a considerable amount of data) above the less frequent classes (not well represented). Secondly, the first pattern is most apparent in the results of the ME and DT classifier. They yield high f-scores for the frequent classes, but very poor results for the other classes, which means that they act as the majority baseline. The NB classifier on the other hand produces good results for the frequent classes, and also presents acceptable results for the other classes. In other words, it is better at predicting all classes, which leads to the best overall mean accuracy and f-scores.

From this preliminary conclusion, we can already conclude that with the feature set with the 1,000 most frequent words of the corpus combined with the NB classifier produces the best results for classifying valence$_1$, performance$_1$ and chat$_1$.

5.2. Stage Two: Most Informative Words

In this second stage, we create our feature sets based on information gain or entropy. As explained in Section 4.2.1, the entropy of the distribution of each word over the different output classes is computed, and the words with the lowest entropy (or highest information gain) are considered the most relevant features for the classifiers.

Only words that appear at least eight times in the corpus are included to define the feature sets (or 'bags-of-informative-words'). It was first intended to take words that occur at least 20 times in the corpus, but for the smaller classes, this resulted in lists of less than 1,000 or 500 features. Therefore we choose to deal with words that occur at least 8 times in the corpus, to represent all classes equally in the feature sets.

In this stage, we also work with ten different feature sets, and all accuracy results are calculated by taking the mean of the ten folds of the cross-validation.

5.2.1. Valence$_2$

Recall that the random baseline for valence is 31.5%, the majority baseline is 40.1% and the upper baseline is 79%.

5.2.1.1. Accuracy

Table 20 presents the valence$_2$ accuracy results for the four feature sets containing the *n*-most informative features of the corpus. A paired t-test on the data set with the 1,000 most informative features reveals that all classifiers perform significantly better ($p < 0.01$) than the majority and random baselines, but also that all percentages are still significantly lower than the upper baseline ($p < 0.01$).

	NB	ME	DT
1,000	64.1%	43.6%	60.3%
500	**65.1%**	43.7%	62.7%
250	61.2%	59.4%	58.6%
100	52.1%	47.7%	47.1%

Table 20. Valence$_2$ accuracy results.

The NB classifier returns the highest accuracies for all feature sets. The overall best result is 65.1% accuracy, with the feature set of the 500 most informative words of the corpus. If we compare this result with the best overall result for valence$_1$ in the first stage, we can determine that using the most informative features instead of the most frequent ones is an improvement of only 1.6%. The worst overall result (43.6%) is again produced by the ME classifier with a feature set comprising 1,000 words.

The feature set with the highest accuracy mean is (as was in the first stage) the set with the 250 most informative words (mean of 59.7%), followed by the set with the 500 most informative words (mean of 57.2%).

The feature set with 500 words delivers the best accuracy result for the NB classifier. For the ME classifier this is the set with the 250 most informative words (59.4%), while the set with 500 words also works best for the DT classifier (62.7%). Remarkable is that while the NB and DT classifier perform better in this stage than in the first one, the ME classifier performs even worse here.

	NB	ME	DT
‘Only standard’	77.6%	66.2%	76.7%
‘Only dialect’	60.3%	61.3%	49.9%
‘Only non-chat’	**84%**	64.1%	74.9%
‘Only chat’	58.8%	60.1%	49.6%

Table 21. Valence$_2$ accuracy results for ‘only standard’, ‘only dialect’, ‘only non-chat’ and ‘only chat’ data.

Table 21 presents the accuracy results for the four limited valence$_2$ sets, each consisting of the 1,000 most informative words extracted from the ‘only standard’ (OS), ‘only dialect’ (OD), ‘only chat’ (OC) or ‘only non-chat’ (ON) data. The feature sets extracted from OS and ON data sets perform better than those from OD and OC sets, with improvements that range from 4.1% to an incredible 26.8% (the discrepancy between OS and OD for the DT classifier). The ON data achieves a slightly higher mean accuracy (74.3%) than the OS data (73.5%), and also delivers the highest accuracy result for valence yet (84%). The classifier with the best mean result of the OS and OD task is the NB classifier with 69% accuracy. It is also the NB classifier that delivers the best mean result for the ON and OC task, with an accuracy of 71.4%.

5.2.1.2. Error Analysis

The precision, recall and f-scores for valence$_2$ classification of the three classifiers is presented in Table 22. Some comparisons with the precision, recall and f-scores from the first stage will be made (Table 15).

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
'Positive'	68.3%	73.6%	**70.8%**	42.3%	97.4%	58.9%	68.3%	65.3%	66.8%
'Negative'	24.7%	13.6%	17.5%	60%	0.9%	1.9%	36.8%	10.1%	15.8%
'Both'	20.2%	19.8%	20%	28.6%	0.8%	1.5%	13%	5.9%	8.2%
'Neutral'	67.8%	65.8%	66.8%	53%	8.8%	15.1%	56.6%	77.7%	65.5%
'N/A'	64.7%	67.8%	66.2%	89.4%	10.1%	18.2%	62.1%	39.4%	48.2%

Table 22. Precision, recall and f-score of the three classifiers for valence$_2$ classification.

The *'positive'* class yields mediocre f-scores (58.9% to 70.8%), which are considerably better than the other classes, but not better than the f-scores that resulted from the first stage experimentation. However, the *'negative'* class is processed better in this stage: f-scores are slightly higher and range from 1.9% to 17.5% (the first stage delivered f-scores from 0.6% to 16.5%). It is however noticeable that the *'n/a'* class, relatively well-classified in stage one, performs somewhat worse here, although this difference appears not to be significant.

5.2.1.3. Conclusion

Valence$_2$ categorization does not produce the hypothesized improvements in comparison to valence$_1$ classification. The best overall accuracy result is augmented by merely 1.6%. The results in this second stage range from 43.6% to 65.1% accuracy. Again, the NB classifier produces the best results, in this case for all of the different feature sets. The feature set with the 250 most informative words of the corpus yields the best mean accuracy results.

In the first stage, we already validated the hypothesis stating that feature sets with the 1,000 most frequent features extracted from 'only standard' (OS) and 'only non-chat' (ON) data yields better results that those extracted from 'only dialect' (OD) or 'only chat' (OC) data. Table 21 showed that this is also the case for feature sets with the 1,000 most informative features extracted from the limited data. Over the three classifiers, the mean accuracy of the OS set is 73.5%, while it is only 57.2% for the OD set. This mean accuracy is 74.3% for the ON and 56.2% for the OC set. From these percentages, we can conclude that the best accuracy results can be achieved by using the feature set extracted from ON data. The 84% accuracy produced by the NB classifier with the ON data is the highest accuracy result for valence classification in this study yet.

It is discernable from the f-scores that the NB classifier yields the best results, closely followed by the DT classifier. Again, the ME classifier produces the worst results. Although results for frequent classes are good, the poor results for the non-frequent classes prove that the classifiers still have difficulties with learning these categories.

5.2.2. Performance$_2$

Recall that the majority baseline for the performance data is 65.4%, the random baseline is 47.1% and the upper baseline is 71%. The feature set used for these experiments consists of the 1,000 most informative features of the corpus.

5.2.2.1. Accuracy

The results shown in Table 23 are only slightly higher (up to 0.2%) than the results we reported in the first stage (see Table 16), and the DT classifier performs slightly worse (decrease of 0.4%).

	NB	ME	DT
1,000	**77.8%**	66.3%	71.5%

Table 23. Performance$_2$ accuracy results.

The NB and DT classifiers are significantly more accurate than the majority and random baselines ($p < 0.01$). The NB classifier performs better than the upper baseline ($p < 0.01$). The result of the ME classifier is only significantly higher than the random baseline ($p < 0.01$).

Again, the NB classifier produces the best overall accuracy, 77.8%. The ME classifier's result is poor in comparison to the other classifiers.

5.2.2.2. Error Analysis

Table 24 shows precision, recall and f-scores of the three classifiers for the task of classifying performance$_2$.

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
'Standard'	66.4%	68.2%	67.3%	75%	0.5%	1%	60%	26.6%	36.8%
'Dialect'	84.9%	83.4%	**84.1%**	66%	99.9%	79.5%	72.7%	92.4%	81.4%
'N/A'	62.6%	65.4%	64%	92.3%	6.4%	11.9%	71.6%	40.3%	51.6%

Table 24. Precision, recall and f-score of the three classifiers for performance$_2$ classification.

All classifiers render high f-scores for the 'dialect' class (from 79.5% to 84.1%), but only the NB classifier produces good f-scores for the other two categories. This classifier has definitely learned something about predicting between 'chat', 'non-chat' and 'n/a', but the other two classifiers have not (with an f-measure down to 1% for the 'standard' class and ME classifier).

5.2.2.3. Conclusion

Performance$_2$ classification results range from 66.3% to 77.8%. The NB classifier yields the best result, and the ME classifier performs the worst. The NB and DT classifier perform significantly better than the random and majority baselines, while the ME classifier only performs significantly better than the random baseline. The NB classifier also performs better than the upper baseline.

All classifiers render very high f-scores for the ‘dialect’ class, but only the NB classifier also delivers good f-scores for the other two categories.

5.2.3. Chat$_2$

Recall that the random baseline for chat is 66%, the majority baseline is 81.1% and the upper baseline is 66%.

5.2.3.1. Accuracy

The accuracy results of the NB and DT classifiers are slightly higher with this feature set than with the set of the most frequent words (improvements of respectively 0.3% and 1%, see Table 24). The accuracy of the ME classifier remains the same.

	NB	ME	DT
1,000	83%	81.9%	**84.2%**

Table 25. Chat$_2$ accuracy results.

All classifiers result in significantly higher percentages than the random, majority and upper baseline ($p < 0.01$), except for the ME classifier, whose accuracy result is not significantly higher than the majority baseline ($p = 0.3$).

The DT classifier delivers the highest score, with 84.2% accuracy. This is significantly higher than the result of the ME classifier ($p < 0.05$), but the difference between the ME and the NB classifier is not significant ($p = 0.1$). The result of the NB classifier is higher than that of the ME classifier, but not significantly ($p = 0.3$). The NB classifier presents the second best result, 83%, while the ME classifier performs the worst, with 81.9% accuracy.

5.2.3.2. Error Analysis

Table 26 shows the precision, recall and f-scores of the three classifiers.

	NB			ME			DT		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
‘Chat’	89.6%	89.9%	89.7%	81.8%	99.9%	89.9%	85.0%	97.8%	**91%**
‘Non-chat’	21%	17.8%	19.2%	100%	1%	2.1%	60%	1%	2.1%
‘N/A’	64.3%	66.9%	65.6%	89.6%	5.7%	10.8%	73.6%	35.2%	47.7%

Table 26. Precision, recall and f-scores of the three classifiers for chat$_2$ clasification.

As in Table 19, the messages labelled as ‘chat’ have very high f-scores, but this is downsized by the mediocre results for the ‘n/a’ class (from 10.8% to 65.6%) and the poor results for the ‘non-chat’ class (down to an f-score of 2.1%). Notable is the 100% recall and 1% precision for the ME classifier of the ‘non-chat’ class, which means that almost none of the ‘non-chat’ items have been identified, but that those that have been identified, were all predicted correctly. This confirms that the ME classifier learns nothing at all.

5.2.3.3. Conclusion

The accuracy results for chat$_2$ classification in this second stage range from 81.9% to 84.2% and were lower than expected. Here, the DT classifier produces the best result, while the ME classifier still predicts poorly. The DT and ME classifiers perform significantly better than the majority, random and upper baselines, except for the ME classifier, whose percentage is not significantly higher than the majority baseline.

The NB classifier achieves the best overall results, while the DT classifier performs the highest f-score for the ‘chat’ class.

5.2.4. Conclusion

This second stage of experiments was performed with ten feature sets based on the *n*-most informative features of the corpus, as measured using an information-theoretic approach. In Section 4.2.2.2, we stated that a frequent word does not necessarily make a relevant word for the classifiers; in other words, we expected that the bag-of-informative-words would produce better results than the bag-of-frequent-words. However, as reported results have established, this is not the case. There is some slight improvement, but this is not enough to be significant. This means that in the context of these experiments the bag-of-informative-words and bag-of-frequent-words are as effective, which could mean that they may overlap and contain a certain amount of the same features (which is not surprising given the relatively small corpus).

The highest accuracy result produced for valence$_2$ classification with the entire data set is 65.1% (NB classifier). This is a 1.6% improvement over the first phase. The NB classifier also yields the highest f-measure (70.8% for the ‘positive’ class), which is lower than the f-score in the first stage. The highest accuracy result for performance$_2$ classification is produced by the NB classifier (77.8%, an improvement of 0.2%). F-measure is the highest for the ‘dialect’ category (84.1% by NB classifier, the same percentage as in stage one). Chat$_2$ classification presents the best accuracy result 84.2% (improvement of 0.3%), which is produced by the DT classifier. As reported in 5.1.4, the latter two accuracy results are higher than valence$_2$ accuracy because the classifiers have to deal with fewer predictable categories. It is also remarkable that the f-scores in this stage are slightly lower to those of the first stage.

At valence$_2$ level, we performed experiments with feature sets extracted from OS, OD, OC and ON data. In the first stage, our hypothesis that experiments with OS and ON data would achieve better results than with OD and OC data was confirmed. In Section 5.2.1.3, we also verified this for the feature sets with the most informative words: mean accuracy results for the OS (73.5%) and ON (74.3%) sets considerably outperform those of the OD (57.2%) and OC (56.2%) sets.

The two patterns discussed at the end of the first stage are also found in this second phase. First, there is the fact that frequently occurring data show high f-scores, while the less frequent data yield poor results. This is probably due to the skewed class distribution of the corpus: a few classes dominate the corpus, while others have little training data. Secondly, we noticed that the DT classifier, and especially the ME classifier, demonstrate this first pattern very clearly, which indicates that they learn less than the well-performing NB classifier. Examining the precision and recall results showed that the ME classifier acts as the majority baseline, and almost always predicts the most frequent class.

In this conclusion, we can confirm that the NB classifier produces the best results for valence and performance classification in general. In the two stages it has become clear that the DT classifier performs better at classifying the chat data.

Section 6
Discussion

This next to last section is divided into three subsections: the first will discuss the main findings of this study, while the second compares these findings to some previous work. We conclude with reporting on a few additional experiments.

6.1. Main Findings

In this subsection we will discuss the main accomplishments and findings of this study: starting with an overview the corpus, and ending with a discussion of the results of the experiments.

6.1.1. The corpus

The first objective of this study was to construct our own corpus with posts and comments extracted from the social networking website Netlog. We annotated these messages on three levels: valence, language performance and chat versus non-chat. Valence was divided into five categories: 'positive', 'negative', 'both', 'neutral' and 'n/a'. Performance and chat were each annotated into three categories: 'standard', 'dialect' and 'n/a' for performance and 'chat', 'non-chat' and 'n/a' for chat. This resulted in the Dutch Netlog Corpus (DNC), which contains 5,500 annotated short messages.

Of these 5,500 messages, Netlog provided us with metadata of 4,563 users (82.96% of the data). This metadata includes the gender, age, country and region of the users. We determined that there are slightly more female users (59.5%) than male users (40.5%). Age-wise, the majority of the users in the corpus are teenagers (72.2% of the users are younger than 20 years old), while only a small part of the users is older than 20 (24.8%). Almost all of the users (91.5%) listed Belgium as their country of origin, while a minority is from the Netherlands (7.1%). A small percentage of the users (1.5%) listed another country as their home. Most of the Belgian authors originate from Antwerp (28%), East-Flanders (22.4%) or West-Flanders (19.7%). Most of the Dutch authors come from North-Holland (1.1%), South-Holland (1.1%) or North-Brabant (1.1%).

We computed annotator agreement (see Section 2.5) for the three levels of annotation, using Cohen's kappa coefficient. For valence, this agreement percentage is quite high: 79%. Performance annotation on the other hand delivered a kappa of 71%. Annotation agreement for annotating chat is mediocre, 66%.

In Section 2.6, we looked at the class distribution of the DNC. Over half of the corpus is annotated as subjective (50.4%), while 35.9% is objective ('neutral') and 13.7% was considered trash ('n/a'). Figure 5 showed the distribution of the valence class: 40.1% of the corpus is annotated as 'positive', 5.8% is 'negative', 4.6% is labelled as 'both' and 35.9% is 'neutral'. Performance and chat distribution were presented in Figure 6, which showed that 65.4% of the corpus is annotated as 'dialect' and 20.9% is labelled as 'standard', while 81.1% of the corpus is annotated as 'chat' and merely 5.2% as 'non-chat'. Furthermore, we looked at the label combinations, from which we concluded that for all valence tags the combinations of dialect with chat are the most frequent. Given the medium and the profiles of the users, this is not surprising. The second most frequent combination is standard with chat, followed by standard with non-chat. The combination of dialect and non-chat is close to non-existent.

6.1.2. Experimental Study

The experimental study was divided into two stages. In each stage, we performed the same experiments, but with different binary feature sets. In the first stage we used feature sets with the *n*-most frequent words of the corpus, while feature sets with the *n*-most informative words of the corpus were employed in the second stage. Each of these two feature sets was divided into ten feature subsets (see Section 4.2.2), which means that we worked with a total of 20 feature sets for all of the experiments.

We hypothesized that the experiments with the most informative features would produce better results than the experiments with the most frequent words, because for text classification tasks frequency is not always an indicator of relevance.

In the remainder of this subsection, we will discuss the results of the valence, performance and chat classification tasks and evaluate employed feature sets and classifiers. In each table, the best overall percentages are bolded.

6.1.2.1. Valence Classification

Table 27 compares accuracy results between the two stages and the three classifiers that are based on the feature set with the 1,000 most frequent/informative words and the feature set with the 250 most frequent/informative words. We chose to present the results of just these two sets because the latter presented the best mean accuracy results for both stages, and the former was used for computing the precision, recall and f-measures in both stages.

		NB	**ME**	**DT**
Stage One	**1,000**	63.5%	43.1%	56.2%
	250	61.1%	59.3%	59.2%
Stage Two	**1,000**	**64.1%**	43.6%	60.3%
	250	61.2%	59.4%	58.6%

Table 27. Comparison of valence accuracies between the stages and the three classifiers.

From Table 27, we conclude that the NB classifier is the best classifier for valence classification in both stages, with a mean accuracy of 62.5% over these four results, and that the ME classifier produces the poorest results, with a mean accuracy of only 51.4%. The DT classifier performs with a mean 58.6% accuracy over these results. Furthermore, there are some slight improvements noticeable in the results of the second stage that range from 0.1% to 4.1%. However, we expected to see more improvement. The overall best result for valence classification over the two stages is 65.1%, produced by the NB classifier in combination with the feature set of the 500 most informative words.

	'Positive'	**'Negative'**	**'Both'**	**'Neutral'**	**'N/A'**
Stage One	**71.2%**	10.4%	31.9%	67.6%	67.1%
Stage Two	70.8%	17.5%	20%	66.8%	66.2%

Table 28. Comparison of valence f-scores for the NB classifier between the stages.

Table 28 presents valence f-scores for the best classifier, based on experiments with the feature sets of the 1,000 most frequent/informative words of the corpus. This table shows how well the NB classifier predicts the different valence classes and compares these results between the two stages. The 'positive', 'neutral' and 'n/a' categories clearly have higher f-measures than the 'negative' and 'both' classes in both stages. The latter two classes are represented by the least amount of training data (skewedness of the class distribution), which explains this discrepancy. If we compare the two stages, we see that the first stage yields the best results, except for the 'negative' class. Using the feature set of the 1,000 most informative words improves the f-score for the negative measures with a considerable 7.1%.

In Section 4.1, we reported Banea et al.'s (2008) statement that even small corpora can prove helpful for research in text classification tasks if the employed classifiers perform with a high enough f-measure (which they set at approximately 66%). The results presented in Table 46 surpass this f-measure of 66% in three of the five cases: the classes with the largest amount of data outperform this 66%. This means that the 'negative' and 'both' categories of corpus should be expanded.

In Tables 14 and 21, we presented the classifier accuracy results for the feature sets with the 1,000 most frequent/informative features extracted from the 'only standard' (OS), 'only dialect' (OD), 'only non-chat' (ON) and 'only chat' (OC) data. The mean accuracy result of the OS set in the second stage (73.5%) is lower than that in the first stage (76.8%), but the mean accuracy of the ON set in the second stage (74.3%) is higher than that of the first stage (69.1%). It is clear that the best accuracy results for valence classification are achieved by using the OS set.

6.1.2.2. Performance and Chat Classification

In this section, we will briefly compare the accuracy results between the two stages for performance and chat classification.

	NB	**ME**	**DT**
Stage One	77.6%	66.2%	71.9%
Stage Two	**77.8%**	66.3%	71.5%

Table 29. Comparison of performance accuracies between the stages and the three classifiers.

Table 29 compares performance accuracy results between stages one and two for the three classifiers. As with valence classification, the NB classifier produces the best results (mean of 77.7%), followed by the DT classifier (mean of 71.7%). The ME classifier performs the poorest at this task (mean of 66.25%). Improvements range from 0.1% to 0.2%, which is even less remarkable than for valence classification. The DT classifier even loses 0.4% accuracy in the second stage.

	NB	ME	DT
Stage One	82.7%	81.9%	83.2%
Stage Two	83%	81.9%	**84.2%**

Table 30. Comparison of chat accuracies between the stages and the three classifiers.

Table 30 presents the comparison of accuracy results between the two stages and the three classifiers. The DT classifier renders the best results for this classification task, with a mean accuracy of 83.7%. The NB classifier follows with a mean accuracy of 82.85%. Again, the ME classifier performs worse than the other classifiers (mean of 81.9%). As we saw in the other classification tasks, the observed improvement is not remarkable. Here it ranges from an extra 0.3% to 1% increase, which is however slightly higher than in the other classification tasks.

The accuracy results for the performance and chat classification task are considerably higher than those for the valence classification task. We assume that this is caused by the number of classes or by the skewedness of training data; to validate this, we carried out additional experiments in Section 6.3.

6.1.2.3. Evaluation of Feature Sets and Classifiers

Section 4.2.2 presented an overview of the 20 employed feature sets. Since experiments with performance and chat classification were only performed with one feature set each, we cannot evaluate or compare these sets. However, there are four feature sets for valence classification, and four subsets of one of these feature sets. First, we have the 1,000, 500, 250 and 100 most frequent/informative features of the corpus. Second, there are four variants of the 1,000 most frequent/informative feature sets, which are extracted from the OS, OD, OC or ON limited data sets.

In the first and second stage, the set with 250 features delivers the best accuracy results (respectively means of 59.9% and 59.7%), followed by the feature set of 500 words (respectively means of 59.6% and 57.2%). In the first stage, the best overall result (63.5%) is achieved by the NB classifier in combination with the set of the 1,000 most frequent words. The combination of the NB classifier with the set of the 500 most information words deliver the best overall result (65.1%) in the second stage. Looking at the feature sets extracted from the limited data in the first stage, we concluded that the OS set delivers the best accuracy results, with a mean 76.8%. The best overall result, also with the OS data, is produced by the ME classifier: 77.5%. In the second stage, the set with ON delivers the best mean accuracy result (74.3%). The NB classifier produces the best overall result with the ON set: 84%.

These results prove that smaller feature sets sometimes work better than larger ones: in both stages of the experimentation, the small feature sets consisting of 250 and 500 words produce better results than the larger feature set with 1,000 words.

As reported in Section 4.3, we have used three supervised classifiers in the experimental phase of this study: the Naïve Bayes, Maximum Entropy, and Decision Tree classifiers. In the valence and performance classification experiments executed in stages one and two, the NB classifier came out as the best classifier for the task. In both stages, the DT classifier performed best for the chat classification task. If we look at the individual usability and performance of the classifiers, we ascertain that the ME classifier can deliver good results with certain feature sets[25], but performs very poorly with others[26]. This classifier has two disadvantages: first, it is very slow and takes much computing time to perform even a simple experiment, and second, we reported that it does not learn anything about predicting valence, performance, or chat and acts as the majority baseline. This poor performance of the ME classifier was surprising, since previous work reported good results (e.g. Boiy et al. (2007)). This contrast can perhaps be explained by the ME classifier's implementation in the NLTK. The NB classifier on the other hand performs well for all classification tasks. Aside from this advantage, it is also very fast (as Boiy et al. (2007) already mentioned). The DT classifier produces rather mediocre results: it does not perform better than the NB classifier and not worse than the ME classifier; this was quite surprising since the DT classifier has an embedded method for feature selection. Even though it is not as slow as the ME classifier, it is still considerably slower than the NB classifier.

In short, the best accuracy results are achieved by using the NB classifier and the feature set with the 1,000 most informative words extracted from OS or ON data (with the DNC it delivered the highest accuracy result in this entire study up until now: 84%).

6.2. Comparison with Previous Work

In this section, we will compare the results obtained in this study with some reported results in the literature. We will only look at accuracy reports obtained with feature sets of word unigrams. Although the conditions of these reported experiments are different than the ones executed in this study, they provide verification as to how good, mediocre or bad our obtained results are. Our highest overall accuracy result achieved with the entire data (ED) set is 65.1%, while the highest overall accuracy result achieved by using limited data (LD) sets is 84% (both by NB classifier).

In their (2002) paper, Pang, Lee & Vaithyanathan experiment with polarity classification ('positive' versus 'negative') on a movie review corpus. The unigram results are among the best of that study, with 81% for the NB classifier and 80.4% for the ME classifier (not based on unigram frequency but unigram presence, i.e. binary features). Boiy et al. (2007) execute the same task on the same corpus, and in their case, word unigram features yield 81.45% for the NB classifier and 84.8% for the ME classifier. Our 65.1% yielded by experiments with the ED set are clearly very poor compared to the results of Pang, Lee & Vaithyanathan and Boiy et al. However, the 84% rendered by the NB classifier with the LD set outperforms, or equals, these results. We assume that this enormous improvement is caused by the nature of the data: the ED set comprises colloquial 'chat' and 'dialect' classes, which differ from 'non-chat' and 'standard' classes in that the language use is anything but uniform. The LD set that produces this percentage consists of the 1,000 most informative features extracted from the 'only non-chat' data: this data is considered to be close to formal Dutch, with uniform orthography, lexicon and grammar. As argued in Section 2.7 and confirmed in both stages of experimentation, the 'only non-chat' and 'only standard' data sets yield better results than the ED set. User-generated text gathered from the Internet is highly noisy and has its own characteristics, which makes it different from normal opinion mining and classification, and needy for accustomed features.

[25] For instance, it delivers the highest accuracy percentage for valence classification: 77.5%. This result is achieved with the feature set consisting of the 1,000 most frequent words extracted from 'only standard' data.
[26] In the first stage, the f-measure of the ME classifier for the class 'negative' is only 0.6%.

Wilson, Wiebe & Hoffmann (2009) on the other hand, add annotations for contextual polarity to an existing corpus, the MPQA corpus. They experiment with four classifiers (BoosTexter, TiMBL, Ripper and SVM[27]) and report three kinds of classifications: first, they perform so-called neutral-polar classification ('neutral' versus 'polar'). Next, they present results for threefold classification between 'positive', 'negative' and 'both'. Thirdly, results for four-way classification are rendered. The neutral-polar classification results in accuracies of 74.6% for TiMBL and 74.6% for SVM. The threefold classification delivers accuracy results of 78.5% for TiMBL and 69.9% for SVM. Accuracy results of 60.1% for TiMBL and 64.5 for SVM are rendered in the four-way classification. From these accuracy results, we can conclude that classifiers perform well until a certain amount of predictable categories. Threefold classification yields the best results for Wilson, Wiebe & Hoffmann's experiments, followed by twofold classification. The results for four-way classification are considerably lower than the others. On these grounds, we decided to perform some additional experiments that are described in the next section.

6.3. Additional Experiments

The mediocre accuracy results for valence classification may be explained by the number of classes it has to deal with. Performance and chat classification, both threefold classification tasks, yield higher accuracies. As seen in Section 6.2, we have not come across reports of fivefold classification in previous work. On the other hand, the mediocre results can also be explained by too noisy 'n/a' and/or confusing 'both' classes, the nature of the data (most of the corpus is written in dialect and chat Dutch, which we have seen is more difficult to classify), and most likely because of the relatively small and skewed training sets for the categories 'negative' and 'both'.

In order to find out if these factors are actually responsible for impeding the results, we have performed four additional experiments. First, we remove the noisy messages ('n/a') from the data. Then we experiment with twofold classification (or polarity classification) between 'positive' and 'negative'. Furthermore, we perform experiments with threefold classification between 'positive', 'negative' and 'both'. A fourth additional experiment involves a threefold classification between 'positive', 'negative' and 'neutral'. All of these experiments are executed with the NB classifier and the feature set with the 1,000 most informative words.

6.3.1. Noise Free Classification

The 'n/a' class of the DNC contains messages that were considered as trash, for instance messages with more than half of the words in another language. These messages can cause noise in the feature sets, which is one of the factors possibly causing the mediocre accuracy results for valence classification.

[27] To minimize the amount of percentages in this paragraph, we will only present results produced by the TiMBL and SVM classifiers.

	Precision	**Recall**	**F-score**	**Accuracy**
‘Positive’	71.7%	78.4%	**74.9%**	**67%**
‘Negative’	23.5%	13.9%	17.5%	
‘Both’	20.8%	21.7%	21.2%	
‘Neutral’	71.7%	68.6%	70.1%	

Table 31. Accuracy result, precision, recall and f-measures for noise free classification.

Table 31 presents the precision, recall, f-measures and accuracy result for this first additional experiment (a fourfold classification task). The accuracy result is 67%, and compared to the result achieved with the ED, this is an improvement of 1.9%. In comparing Table 31 with Table 15, which presents the f-scores for valence$_1$ classification in the same conditions, we see that in most cases f-scores are higher in this experiment. Especially the 7.1% improvement for the ‘negative’ class is remarkable and proves that the ‘n/a’ class interferes with predicting this ‘negative’ class. However, the 10.7% impairment for the ‘both’ class is also remarkable. Recall and precision results are high for the two classes represented by the largest amount of data, ‘positive’ and ‘neutral’. Not taking the noisy ‘n/a’ category into account does improve results, but not considerably.

6.3.2. Twofold Classification

Many papers have been written about sentiment classification into just two classes, which we have called polarity classification in Section 1. In Section 6.2, we reported high accuracy results (from 80.4% to 84.8%) for the twofold classification studies of Pang, Lee & Vaithyanathan (2002) and Boiy et al. (2007). With this additional experiment, we want to verify whether polarity classification will increase accuracy for the DNC data.

	Precision	**Recall**	**F-score**	**Accuracy**
‘Positive’	91.2%	94.2%	**92.7%**	**87%**
‘Negative’	8.8%	63.1%	15.4%	

Table 32. Accuracy result, precision, recall and f-measures for twofold classification.

The produced accuracy result is an amazing 87%, which is a 29.1% improvement to the highest achieved accuracy result for fivefold valence$_2$ classification with the ED set. This percentage is also considerably higher than the reported results of Pang, Lee & Vaithyanathan (2002) and Boiy et al. (2007). Apparently, the DNC data is better suited for twofold classification into ‘positive’ and ‘negative’ classes than movie reviews: the latter often combine implicitly expressed and multiple contrasting opinions, while the short DNC data is usually either positive or negative. Looking at the precision, recall and f-scores reveals that the NB classifier performs very well for the ‘positive’ category, but that predicting ‘negative’ is still troublesome. Recall for ‘negative’ messages is mediocre (63.1%), which means that little over half of the messages that are ‘negative’ are actually predicted as ‘negative’. The poor precision (8.8%) means that less than 10% of the messages predicted as ‘negative’ by the classifier were effectively ‘negative’. The high accuracy result was promising, but the poor f-measure for ‘negative’ suggests, again, that the classifier has not learned enough (which is caused by the skewed training data set for the ‘negative’ class).

This experiment proves that narrowing down the predictable classes from five to two greatly improves accuracy and that skewedness is also an important factor. The other factor possibly causing the mediocre accuracy results in this study is the nature of the data: we already proved that the feature sets extracted from 'only standard' (OS) and 'only non-chat' (ON) data produce considerably higher results than those extracted from the 'only dialect' (OD) and 'only chat' (OC) data. Table 33 presents the accuracy results for polarity classification on these limited data feature sets.

	Accuracy
'Only standard'	**94.6%**
'Only dialect'	84.7%
'Only non-chat'	93.5%
'Only chat'	86.9%

Table 33. Accuracy results for twofold classification with features from limited data sets.

As expected, the OS and ON accuracy results are considerably higher than the OD and OC results, with improvements of 9.9% (OS to OD) and 6.6% (ON to OC). More importantly, these accuracy results are substantially higher than those achieved by the fivefold classification experiments (see Table 14 and 21). The best accuracy result for valence$_1$ classification is 77.5% for OS; for the valence$_2$ classification this is considerably higher, with 84% for ON. In the case of two-way classification, the best accuracies are an astonishing 94.6% for OS and 93.5% for ON.

The results show that by lowering the number of classes and by using features extracted from 'clean' data, the results can improve up to ca. 95% accuracy.

6.3.3. Threefold Classification

Twofold classification produces substantially higher accuracy results compared to five-way classification. Additionally, it is also interesting to discuss results of threefold classification with the DNC data. First, we will discuss results achieved by threefold classification into 'positive', 'negative' and 'both' (PNB) as can be seen in Table 34. Second, the results for threefold classification into 'positive', 'negative' and 'neutral' (PNN) will be presented in Table 35. In PNN, we will see the results of not taking the two noisy classes ('n/a' and 'both') in account.

	Precision	**Recall**	**F-score**	**Accuracy**
'Positive'	85.6%	90.9%	**88.2%**	**77.5%**
'Negative'	45.1%	24.6%	31.8%	
'Both'	26.4%	27.3%	26.8%	

Table 34. Accuracy result, precision, recall and f-measures for threefold classification into 'positive', 'negative' and 'both'.

	Precision	**Recall**	**F-score**	**Accuracy**
'Positive'	75.2%	80.3%	**77.7%**	**71.5%**
'Negative'	27.8%	19.9%	23.2%	
'Neutral'	72.2%	70.0%	71.1%	

Table 35. Accuracy result, precision, recall and f-measures for threefold classification into 'positive', 'negative' and 'neutral.

The results for PNB classification are considerably higher than those achieved by PNN classification: PNB classification performs with 77.5% accuracy, while PNN classification only sports 71.5% accuracy. Only the f-score of ‘neutral’ (PNN) is substantially higher than that for ‘both’ (PNB), but this is not surprising since the classifier has more training data for predicting ‘neutral’. These threefold classifiers perform considerably better than the fivefold classifier, but their results are not as impressive as the twofold classifier. The PNN result is lower than the PNB result, but this is mainly caused by a lower f-score for ‘positive’ and higher f-scores for the other two classes.

6.3.4. Conclusion

This section with additional experiments is created to verify whether the presumably impeding factors, the number of classes, nature of the data, or the skewedness of the class distributions, actually influence the results of the NB classifier and employed feature set combination.

	NB
Twofold	**87%**
Threefold	71.5% to 77.5%
Fourfold	67%
Fivefold	65.1%

Table 36. Accuracy results for two-, three-, four- and fivefold classification.

From Table 36 we can immediately see that increasing the number of classes decreases the results. Lowering the amount of classes is merely facilitating the classification task, while applications require fine-grained classification systems that can handle multiple classes.

Table 33 showed that using feature sets extracted from limited data sets, more specifically the limited sets with ‘clean’ data, essentially improves accuracy of the NB classifier. This means that the nature of the data indeed influences the performance of classifiers. The highest noted accuracy is 94.6% for the OS set.

As we saw earlier, the results improve considerably for the classes represented by the largest amount of data in de corpus, i.e. ‘positive’, ‘neutral’, ‘dialect’ and ‘chat’. This indicated that the class distributions of the DNC is skewed, and still needs some work.

Section 7
Conclusion

7.1. Main Conclusions

In the outset of this report, we stated two hypotheses: first, the ‘only standard’ and ‘only non-chat’ data should deliver better results than the ‘only dialect’ and ‘only chat’ data because we assume that the nature of the data can facilitate or hinder the classification process. Second, we assumed that the feature sets with the *n*-most informative words would produce better results than the feature sets with the *n*-most frequent words, since frequent words are not necessarily relevant words for classification. In Section 5, we confirmed that the first hypothesis is correct. We argued that user-generated content extracted from the Internet, valuable data for computational linguistics research, has a need for different feature vectors. The second hypothesis however remains unconfirmed in this study. In Section 6.2.1, we reported only slight improvements when comparing the results of the first and second stage. This means that our bag-of-frequent-words and bag-of-informative-words are as effective in the context of our experiments, which could mean that they are partly overlapping. The latter is not surprising given the relatively small corpus.

The mediocre results for valence classification in general are likely to be caused by a combination of three factors: the number of classes, the nature of the data and the skewed distribution of the classes. The additional experiments discussed in Section 6.3 have proved that this is the case: by decreasing the number of classes the accuracy increases (for twofold classification with the entire data set up until 87%), and by using the feature sets extracted from the clean ‘only standard’ and ‘only non-chat’ sets these results improve even more (for twofold classification up to 94.6% for ‘only standard’ set). The last factor, the skewedness of the corpus, is most probably the most probable cause for the mediocre results in this study: classes represented by large amounts of training data are handled very well by the classifiers, but as the f-scores demonstrate, the fewer occurring categories are processed very poorly, which causes the accuracy drops.

7.2. Future work

While considerable work has been done for this study, there was not enough time to accomplish all objectives. This section will list some future work that builds on this report.

In the Preface, we described a third objective that aimed to perform explanatory statistical analyses on the corpus, especially the correlations between the language use, age, gender and location of the Netlog authors used in the data. This data was acquired close to the deadline of this study, which left little time to implement the data into the corpus and to execute the statistical analyses. Since these analyses could provide insightful information about the corpus as to whether, for instance, boys use more dialect than girls or whether older users use more standard language than younger users, they will be performed at a later time.

Enlarging the DNC is another future objective, and particularly the ‘negative’, ‘both’, ‘standard’ and ‘non-chat’ classes need to be expanded in order to provide sufficient training data for the classifiers, which should lead to substantial improvements for the f-scores of these categories. It is probably the skewedness of the data that explains the mediocre results in this study.

Other future work involves implementing new features, in particular word and character *n*-grams and emoticons. We especially think that adding word bigrams and character bi- and trigrams to the (most frequent and most informative) feature sets could lead to better accuracy results, since they have proven to be useful feature vectors in previous work. Although reported results in Section 5 have pointed out that small feature sets with 250 to 500 features deliver better accuracy results, if *n*-grams are appended to these sets, it might be advantageous to use larger feature sets. As explained in Section 4.2.1, linguistic analysis features cannot be used with the DNC, since the majority of this data is either ‘dialect’ or ‘chat’, and therefore not suitable for linguistic analyses (because of the unusual and divergent orthography, lexicon and grammar).

This relates to another direction future work could take: further experimentation with the features extracted from the ‘only standard’ and ‘only non-chat’ sets. Section 4.2.2.2 showed that although there is a built-in ‘trash’ category in the corpus, there is still much noise in the entire data set. A closer look at the features from the limited data sets indicated that this data are much ‘cleaner’, and might therefore be more useful for opinion classification tasks. In the same vein, research also has to focus on finding effective and accustomed feature sets for classifying noisy and user-generated content.

In this study, we have employed three supervised classifiers, but it would be interesting to investigate the results rendered by other supervised classifiers, such as SVMs, TiMBL, Rocchio or neural networks.

Although it is far beyond the scope of this study, it could be interesting to use regression instead of classification on the DNC. Regression predicts a value between, for instance, -1 and +1, and provides more room for fine-grained prediction.

Research in sentiment analysis has already produced some groundbreaking applications, but there is still a very long way to go before computers will truly understand those two things that all humans have in common: emotions and language.

References

ALM, C. O., ROTH, D. & SPROAT, R. (2005), “Emotions from Text: Machine Learning for Text-Based Emotion Prediction”. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, p. 347–354.

BAHALUR, A., STEINBERGER, R., KABADJOV, M., ZAVARELLA, V., VAN DER GOOT, E., HALKIA, M., POULIQUEN, B. & BELYAEVA, J. (2010), “Sentiment Analysis in the News”. In *Proceedings of the LREC Conference 2010*, p. 2216-2220.

BALOG, K., MISHNE, G. & DE RIJKE, M. (2006), “Why Are They Excited? Identifying and Explaining Spikes in Blog Mood Levels”. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, p. 207-210.

BANEA, C., MIHALCEA, R., WIEBE, J. & HASSAN, S. (2008), “Multilingual Subjectivity Analysis Using Machine Translation”. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, p. 127-135.

BENAMARA, F., CESARANO, C., PICARIELLO, A., REFORGIATO, D. & SUBRAHMANIAN, VS. (2007), “Sentiment Analysis: Adjectives and Adverbs are Better than Adjectives Alone”. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM-2007)*, p. 203-206.

BIRD, S., KLEIN, E. & LOPER, E. (2009), *Natural Language Processing with Python. Analyzing Text with the Natural Language Toolkit*. Sebastopol: O’Reilly Media, Inc.

BLITZER, J., DREDZE, M. & PEREIRA, F. (2007), “Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for sentiment classification”. In *Proceedings of the Association for Computational Linguistics (ACL-2007)*, p. 440-447.

BOIY, E., HENS, P., DESCHACHT, K. & MOENS, M.-F. (2007), “Automatic Sentiment Analysis in On-Line Text”. In *Proceedings of the Conference on Electronic Publishing (ELPUB-2007)*, p. 349-360.

CHOI, Y., CARDIE, C., RILOFF, E. & PATWARDHAN, S. (2005), “Identifying Sources of Opinions With Conditional Random Fields and Extraction Patterns”. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2005)*, p. 355-362.

DAELEMANS, W. & VAN DEN BOSCH, A. (2005), *Memory-Based Language Processing*. Cambridge: Cambridge University Press.

DAVE, K., LAWRENCE, S. & PENNOCK, D.M. (2003), “Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews". In *Proceedings of the 12th International Conference on World Wide Web*, p. 519-528.

DAYAN, P. (1999), "Unsupervised learning". In *The MIT Encyclopedia of the Cognitive Sciences*, WILSON, R. and KEIL, F. [ed.].

ESULI, A. & SEBASTIANI, F. (2006a), "Determining Term Subjectivity and Term Orientation for Opinion Mining". In *Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, p. 193–200.

ESULI, A. & SEBASTIANI, F. (2006b), "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining". In *Proceedings of Language Resources and Evaluation Conference (LREC-2006)*, p. 417-422.

FELDMAN, R. & SANGER, J. (2007), *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.

FRADKIN, D. & MUCHNIK, I. (2006), "Support Vector Machines for Classification". In ABELLO, J. & CARMODE, G. [Eds.], *Discrete Methods in Epidemiology*, DIMACS Series in *Discrete Mathematics and Theoretical Computer Science*, Vol. 70, p. 13-20.

GAMON, M. (2004), "Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis". In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, p. 841-847.

GILLIS, S., DAELEMANS, W. & DURIEUX, G. (2000), "'Lazy learning': A comparison of natural and machine learning of stress". In P. Broeder and J.M.J. Murre [ed.], *Models of Language Acquisition: inductive and deductive approaches*, p. 76-99.

GUYON, I. & ELISSEEFF, A. (2003), "An Introduction to Variable and Feature Selection". In *Journal of Machine Learning Research* 3, p. 1157-1182.

HATZIVASSILOGLOU, V. & McKEOWN, K.R. (1997), "Predicting the semantic orientation of adjectives". In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics (ACL-1997)*, p. 174-181.

HIROSHI, K., TETSUYA, N. & HIDEO, W. (2004), "Deeper Sentiment Analysis Using Machine Translation Technology". In *Proceedings of the 20th International Conference on Computational Linguistics*, p. 494-500.

HU, M. & LUI, B. (2004), "Mining and Summarizing Customer Reviews". In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004 (KDD-2004)*, p. 168–177.

JURAFSKY, D. & MARTIN, J.H. (2009), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* New Jersey: Pearson

Education, Inc.

KIM, S.-M. & HOVY, E. (2004), "Determining the Sentiment of Opinions". In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, p. 1367-1373.

KIM, Y.S., STREET, W.N. & MENCZER, F. (2003), "Feature Selection in Data Mining". In WANG., J. (2002), *Data Mining: Opportunities and Challenges*, p.80-105.

KOBAYASHI, N., INUI, K. & MATSUMOTO, Y. (2007), "Extracting Aspect-Evaluation and Aspect-of Relations in Opinions Mining". In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, p. 1065-1074.

KÖNIG, A.C. & BRILL, E. (2006), "Reducing the Human Overhead in Text Categorization". In *Proceedings of the 12th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, August 2006, p. 598-603.

KU, L.-W. & CHEN, H.-H. (2007), "Mining Opinions from the Web: Beyond Relevance Retrieval". In *Journal of the American Society for Information Science and Technology* 58(12), p. 1838-1850.

LIU, B. (2010), "Sentiment Analysis and Subjectivity". To appear in *Handbook of Natural Language Processing*, Indurkhya, N. & Damerau, F.J. [Eds.].

LLOYD, L., KECHAGIAS, D. & SKIENA, S. (2005), "Lydia: A System for Large-Scale News Analysis". In *String Processing and Information Retrieval (SPIRE 2005)*, p. 161-166.

MEJOVA, Y. (2009), "Sentiment Analysis: An Overview". Comprehensive exam paper, available on http://www.cs.uiowa.edu/~ymejova/publications/CompsYelenaMejova.pdf [2010-02-03].

MISCHE, G. & GLANCE, N. (2006), "Predicting Movie Sales from Blogger Sentiment". In *AAAI Symposium on Computational Approaches to Analyzing Weblogs*, p. 155-158.

MITKOV, R. [Ed.] (2003), *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press.

MORANTE, R. & DAELEMANS, W. (2009), "A Metalearning Approach to Processing the Scope of Negation". In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009)*, p.21-29.

MULDER, M., NIJHOLT, A., DEN UYL, M. & TERPSTRA, P. (2004), "A lexical grammatical implementation of affect". In *Proceedings of the 7th International Conference Text, Speech and Dialogue, Lecture Notes in Computer Science* (3206), p. 171-178.

NG, V. & CARDIE, C. (2003), "Weakly-Supervised Natural Language Learning Without Redundant Views". In *Proceedings of the Conference on Human Language Technologies – North American Chapter of the Association for Computational Linguistics (HLT-NAACL-2003)*, p. 94-101.

PAK, A. & PAROUBEK, P. (2010), "Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In *Proceedings of the LREC Conference 2010*, p. 1320-1326.

PANG, B. & LEE, L. (2004), "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, p. 271-278.

PANG, B. & LEE, L. (2008), "Opinion Mining and Sentiment Analysis". In *Foundations and Trends in Information Retrieval* 2 (1-2), p. 1–135.

PANG, B., LEE, L. & VAITHYANATHAN, S. (2002), "Thumbs Up? Sentiment Classification Using Machine Learning Techniques". In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, p. 79-86.

PARROTT, W. (2001), *Emotions in Social Psychology*. Philadelphia: Psychology Press.

PRABOWO, R. & THELWALL, M. (2009), "Sentiment Analysis: A Combined Approach". In *Journal of Informetrics* 3 (2), p. 143-157.

QUIRK, R., GREENBAUM, S., LEECH, G. & SVARTVIK, J. (1985), *A Comprehensive Grammar of the English Language*. New York: Oxford University Press.

RENTOUMI, V., PETRAKIS, S., KLENNER, M., VOUROS, G.A. & KARKALETSIS, V. (2010), "United We Stand: Improving Sentiment Analysis by Joining Machine Learning and Rule-Base Methods". In *Proceedings of the LREC Conference 2010*, p. 1089-1094.

RUSSELL, S. & NORVIG, P. (2003), *Artificial Intelligence: A Modern Approach*. New Jersey: Pearson Education, Inc.

SEGARAN, T. (2007), *Programming Collective Intelligence*. Sebastopol: O'Reilly Media, Inc.

STONE, P.J., DUNPHY, D.C. & SMITH, M.S. (1966), *The General Inquirer: A Computer Approach to Content Analysis*. Oxford: MIT Press.

TURNEY, P.D. (2002), "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, p. 417-424.

WIEBE, J., BRUCE, R., MARTIN, M., WILSON, T. & BELL, M. (2004), "Learning Subjective Language". In *Computational Linguistics* 30, p. 277-308.

WIEBE, J. & MIHALCEA, R. (2006), "Word sense and subjectivity". In *Proceedings of COLING-ACL 2006*, p. 1065-1072.

WIEBE, J., WILSON, T. & CARDIE, C. (2005), “Annotating Expressions of Opinions and Emotions in Language”. In *Language Resources and Evaluation* 39 (2/3), p. 164-210.

WILSON, T., WIEBE, J. & HOFFMANN, P. (2005), “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis”. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)*, p. 347-354.

WILSON, T., WIEBE, J. & HOFFMANN, P. (2009), “Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis”. In *Computational Linguistics* 35 (3), p. 1-34.

YI, J., NASUKAWA, T., NIBLACK, W. & BUNESCU, R. (2003); “Sentiment Analyzer: Extracting Sentiments About a Given Topic Using Natural Language Processing Techniques”. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, p. 427-434.

YU, H. & HATZIVASSILOGLOU, V. (2003), “Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences”. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, p. 129–136.

Acknowledgments

First of all, I would like to express great gratitude to my supervisor Prof. Dr. Walter Daelemans for his time, his many insights, and his support. Thank you for pushing me to be a better researcher.

In addition I want to thank my assessor Dr. Roser Morante and everyone at the CLiPS research centre.

I thank Netlog for providing me the raw data from which I could construct a corpus.

I would also like to thank my big brother Prof. Dr. Benjamin Schrauwen (and his company Mollom) for his ideas and programming advice, but most of all for his patience.

At last, I wish to thank my (other) friends and family, for supporting me unconditionally.

Appendix

ANNOTATION MANUAL FOR THE DUTCH NETLOG CORPUS

For the annotation of the Dutch Netlog Corpus (DNC) we use three different kinds of tags: one denoting the valence of the text (whether an expression is 'positive', 'negative', 'both' or 'neutral') and two describing the language performance of the user ('standard' versus 'dialect' and 'chat' versus 'non-chat').

1. General rules

This corpus deals with expressions of opinion and sentiment; therefore only directly stated speech is allowed. In other words, messages with any kind of reported speech (i.e. not expressed by the author himself) are excluded (tag *not applicable* or *n/a*), even if the reported speech is only part of the text message.

Message (1) is discarded since it contains reported speech: aside from expressing his own feelings, the writer also reports what Rob said.

(1) "Ik vind het wel mooi, al zegt Rob dat hij het lelijk vindt"

A next step in deciding whether a text is suitable for the corpus, is asking the question *"How does the writer feel about someone/something?"* If this question is not applicable to or not answerable by the text, it does not have emotional content and it is regarded as being neutral (tag *neutral*).

For example, the question "How does the writer feel about someone/something?" cannot be answered by (2) and therefore this message is regarded as neutral. A sentence like (3) can however, and is tagged as positive.

(2) "Ik ga morgen winkelen, ga je mee?"
(3) "Prachtigge fotoow!"

Messages from Netlog can be mistakenly tagged as Dutch, while they comprise only English, French or words in other languages. Such texts are discarded as well (tag *n/a*). However, messages that comprise at least 50% words in Dutch are allowed. Furthermore, if a text contains more than 50% emoticons or punctuation marks over words, it is excluded as well (tag *n/a*). This 50% is allowed, because using foreign words or expressions and punctuation and emoticons is part of the evolution of natural language, especially in the context of the Web.

One last category of text that is excluded from the corpus is incomprehensible text (whether the cause be extreme dialect or extreme chat language, tag *n/a*). Incomprehensible text messages can comprise incomprehensible dialect words or incomprehensible 'chat' writing styles. This does not include words with added letters, but can include words with deleted letters if the meaning of the words (i.e. the *overall* meaning of the message) is completely unclear.

2. Valence tags

For valence or sentiment tags, we use five different categories: 'positive', 'negative', 'both', 'neutral' and 'not applicable'.

Messages that provide a positive answer to the question "How does the writer feel about someone/something?" are tagged **'positive'**. 'Positive' includes expressions of friendship, love, joy, excitement, etc. Usually, positive messages are not hard to detect because of typical semantic expressions ("`Ik hou van jou`", "`Wat is het mooi`", etc.). Sometimes however, words with a positive valence can form a message that is actually neutral or negative towards the topic discussed. This is why the validating question is relevant. A statement like (4) contains a typically positive word ("`mooi`"), but does not express a strong emotion of the writer towards someone or something (although it can be a negative one).

Note: messages that are (sexually) inappropriate but do express a positive feeling toward a person (e.g. "`Ik vind je zo mooi en lekker geil eruit zien`") are considered positive! Messages containing words of missing someone are also considered to be positive, since they encompass the positive feeling of liking someone needed to be able to miss this person.

(5) and (6) are examples of messages with a clearly positive undertone.

(4) "`Mooi zijn is ook niet alles`"
(5) "`Die haar haartjes zen kei mooi op dieje foto :) x`"
(6) "`mooie wagen love it grtjs`"

A negative answer to the question "How does the writer feel about someone/something?" leads to the tag **'negative'**. This includes hate, racism, profanity, sadness, bullying, etc. Negative messages are usually not hard to detect either, since they often lean towards extreme language use and also consist of some typical semantic expressions (e.g. (7) and (8)). As illustrated before, words with a negative valence can form a message that is actually neutral towards the discussed topic.

(9) and (10) are examples of messages with a clearly negative undertone.

(7) "`khaat uuu`"
(8) "`Wa'n slechte foto van u`"
(9) "`Egt waar kben kwaad op jou. vraag me niet wrm want kzeg tg niets`"
(10) "`oude doze!!`"

In some cases, both positive and negative emotions are uttered in one expression, for example (11). Saying a good thing about one person followed by a negative comment about that same person or another one is frequent, and the tag **'both'** applies here. In the corpus, message and reply sometimes co-occur within the same text message, which can lead to contrasting sentiments from two (or perhaps more) authors. In this case, the *both* tag also applies.

(12) and (13) are examples of messages that contain both negative and positive emotions toward something or someone.

(11) "Bent is fantastisch, maar Linda is een heks"

(12) "ja gisele is wel tof maar ze kan ook zage"

(13) "Sammeke das toch ne schrale patat h?:d maar wel een schatteke haha^^ xx"

The fourth possible tag is **'neutral'**. Neutral messages display no emotions towards someone or something, neither positive nor negative. These messages consist of general information like announcements or statements, e.g. a meeting place and/or time, a to do list, information about the user, random conversation, etc. The question "How does the writer feel about someone/something?" can therefore not be answered by these neutral messages. Neutral messages include: birthday, Christmas, new year, welcome wishes etc.; saying that they're sorry or they're thanking somebody, to take care, etc.; asking how somebody is doing.

(14) and (15) are examples of messages that do not contain emotions and are therefore considered neutral.

(14) "zit je weer al op de zomer te wachten??????"

(15) "helaba, hoe is het daar ? en al druk aan het werken alles in orde met de gezondheid ? groetjes"

The not applicable or **'n/a'** tag is relevant for the previously mentioned cases: reported speech, messages with more than half of the words in foreign language(s) and/or emoticons and punctuation marks, and incomprehensible text messages. Incomprehensible text messages can comprise incomprehensible dialect words or incomprehensible 'chat' writing styles. This does not include words with added letters, but can include words with deleted letters if the meaning of the words (i.e. the *overall* meaning of the message) is completely unclear.

(16) is an example of a message that contains more emoticons than words, while (17) contains too many punctuation marks. (18) is an example of an incomprehensible message (due to heavy dialect use) that is excluded from the corpus.

(16) "(: (: (: gelukkige verjaardag ^^ ;) ;) ;)"

(17) "hallo..................... oe ist???"

(18) "Fiisj kwarn dar wok (peisk &lt;_&lt; xd kwnmi)"

3. Language performance tags

Since language on the Internet is a good example of the rapid evolution of our language, and its use is largely performance-based, some elementary tags concerning the language performance of Netlog users can be very useful. This data is especially relevant in combination with the age and location of the users. There are two dimensions in this category of tags and therefore two phases in the annotation process: ‘standard’ versus ‘dialect’ Dutch and ‘chat’ versus ‘non-chat’ language.

Language on the Internet also makes use of a specific vocabulary (e.g. “overmooi”) and writing style (e.g. “fotoow”) that varies greatly between users (subcultures, age and peer groups, etc.) and that is often referred to as youth language: it is a hybrid of dialect (vocabulary) and chat (writing style) characteristics. Messages exhibiting these words or word forms are tagged dialect in the first phase and chat in the second phase.

3.1. Standard versus non-standard Dutch

In the first phase of the annotation process of language performance tags, we aim to determine whether a text message is written in standard or non-standard Dutch.

Standard Dutch refers to messages that are written in proper Dutch; this means that the vocabulary (and in a milder way the spelling) is consistent with formal standard Dutch conventions (i.e. Het Groene Boekje, Van Dale woordenboek, etc.). However, a minor spelling error that occurs frequently and often unconsciously in Dutch (e.g. dt-errors) or in any language (e.g. typo’s) is ignored, but when more than one of the words in the text are incorrectly spelled, or when the same error occurs twice, or when obvious deliberate spelling errors are made (e.g. frequent n-deletion at the end of verbs or deletion of characters in words), the text is categorized as non-standard or dialect Dutch (e.g. “wa” and “lope” instead of “wat” and “lopen”) or sometimes even as chat (e.g. “das ni waar” is dialect because of the contraction “dat is” and the –et deletion of “niet”, and it is chat because the writing of this contraction and deletion are not consistent with standard writing Dutch conventions). One occurrence of the contraction “ik” into “k*verb*”, or one occurrence of “da” or “me” instead of “dat” or “met” is enough to label a document as non-standard.

<u>Note</u>: messages that are (sexually) inappropriate but that are written conform the formal rules of standard Dutch (e.g. “Ik vind je zo mooi en lekker geil eruit zien”) are considered standard!

Non-standard or dialect Dutch is inconsistent with standard Dutch (e.g. using “gij” instead of “jij”, “tis” instead of “het is”, “ne gelukkige verjaardag”) and is distinguished by its vocabulary, grammar and pronunciation. Dialect can be geographically (regiolect) or socially (sociolect) different from standard Dutch. A text message is considered to be dialect if even only one word in the message is dialect (e.g. “gij” instead of “jij”, “drij” instead of “drie”, “kmoet” instead of “ik moet”, “ni” instead of “niet”, “ma” instead of “maar”, “nen auto” instead of “een auto”). This also includes –ke diminutives (e.g. “autoke”), inflection of articles (e.g. “den auto”) and pronouns (e.g. “mijnen auto”), ‘prefixes’ such as “kei-”, “mega -”, “over -”, etc.

If we look at examples (19a) and (19b), we can see that they both belong to the category standard Dutch. (19b) contains one minor spelling error, which is not enough to label it non-standard or chat (i.e. it could be a typo). Looking at example (20), we see that this is still a message written in standard Dutch, but capitalization and punctuation marks are missing. This message will be labeled chat in the second phase (see below). It is distinct from *dialect* in that dialect uses a distinct vocabulary (e.g. "gij" instead of "jij") and/or grammar (e.g. "ik em da ni gezien" instead of "ik heb dat niet gezien").

(19a) "Ken je hem? Want ik denk eigenlijk dat ik hem nog nooit gezien heb."

(19b) "Ken je hem? Want ik denk eigelijk dat ik hem nog nooit gezien heb."

(20) "ken je hem ik denk dat ik hem nog nooit gezien heb"

3.2. Chat versus non-chat language

The second phase of the language performance tag process evolves around the distinction between chat and non-chat language.

Chat language is very extensive and takes on many forms, but it is very distinct from non-chat language. It transforms dialect and standard Dutch by writing in a more phonetic way, by deleting characters or adding repetitive ones, by not using any punctuation marks or exceedingly much of them, by using emoticons or typical chat language abbreviations (e.g. "omg" for "oh my god", "btw" for "by the way", "hvj" for "hou van je" or "zjg" for "zie je graag"), etc. One occurrence of emoticons is enough to label a text message as chat (but: when the entire text is according to formal Dutch writing conventions (i.e. punctuation marks and capitalization), one emoticon is tolerated). One occurrence of the other abovementioned characteristics (e.g. deleting or adding a character) is not enough to label a text message as chat, because it could be a typo. At least two occurrences are necessary (e.g. "jee" could be a typo, but it is unlikely that "jeee" is still a typo; "thx").

Non-chat language conforms to the writing style of formal standard Dutch, which means using correct punctuation marks and capitalization, etc. This also means that texts containing rightly capitalized letters but no punctuation marks, and vice versa, are considered to be chat language. But, since these texts occur only in the context of the Internet, an occurrence of maximum three exclamation or question marks is tolerated per sentence, if the rest of the text meets the requirements of formal Dutch writing conventions. This rule also applies to the use of 'kisses' in the form of x's at the end of messages (maximum three occurrences, else *chat*). And if a text only misses one occurrence (e.g. in "ik val echt op jou!") it is also tolerated as *non-chat*.

Examples (21a) and (21b) are both written in dialect Dutch, but there clearly is a difference in typography: (20a) is non-chat dialect Dutch and (20b) is chat dialect Dutch.

We can see the same with (22a) and (22b): (22a) is not consistent with standard Dutch (e.g. "eilijk" instead of "eigenlijk" and "hem" instead of "heb"), but does not use correct capitalization and punctuation. It is therefore labeled as dialect in the first phase and as chat in the second phase. (22b), however, clearly shows chat characteristics (e.g. contractions of "wnt" for "want" and "gzien" for "gezien") and is categorized as dialect in the first phase and chat in the second phase.

(21a) “Ge zijt nen engel.”

(21b) “Gzyt n’engel”

(22a) “Ken je ‘em? Want ‘k denk eilijk dak ‘em nog nooit gezien hem.”

(22b) “ken jem, wnt kdenk eilijk dak m nog nooit gzien hem”

4. Tag codes and more examples

Writing tag codes instead of entire tag names, such as ‘positive’ or ‘negative’, saves a lot of time.

The tag codes for valence tags are:

+ for ‘positive’

“ghebt mooi haar :)”

“met jou wil ik wel gezien worden . vind je echt en super leuke mooie vrouw”

“Babbyy , kmisjeee =(x”

- for ‘negative’

“hja echt welt schoonste dat ge kunt krijgen in u familie”

“gy lacht oek ni he xd x”

“ja heel zeker spijtig wel maar ja niets aan te doen nu he in februari toch gedaan dus xxx”

& for ‘both’

“je kiekt gy lik alsan vies xd Hebjgraag!”

“gy staat der super sexya op. echt nie gij zijt gewoon sexy ;) ma ik vindt mij zelf lelijk !”

“Kben perfeecct (niwaar ze) jaweeeel :) Neenee kben ni compleet er zijn stukken die ontbreken :D”

_ for ‘neutral’

“eeey wrm doe je die tietspleeet weg”

“hoi zin in een babbelke”

“mensen die roeren te bewijzen da ze slim zijn op tv. meent ge da nu?”

/ for ‘n/a’

“Silkeeee - ily ily ily !”

“Fiisj kwarn dar wok (peisk &lt;_&lt; xd kwnmi)”

“tu eres loco y guapo y mono (: Haha , zie je wel da 'k spaans kan xd”

The symbols used for language performance tags are:

s for ‘standard’

“emma kan da_ mooi zingen”

“poes toch , jij bent zoveell beter ?”

“o zo mooi zeg mercitjz”

d for ‘dialect’

“ken joen lik van ergens”

“mercii xx jij bnt ook de best xdxx”

“hoi zin in een babbelke”

Maar niet hier:

“Shoeke, kom je heel graag en zomaar met heel veel liefde in je oortje fluisteren, dat jij de allerliefste schat bent op deze wereld!”

c for ‘chat’

“thx, nice ! xdank u :) Alles goed?Precies een grote fan van de kokoriko h? jij :) x”

“nen gelukkig nieuwjaar zus 4 x kusjes”

“Tis bere mooie foto :) x Merciii Poezeee x”

n for ‘non-chat’

“Shoeke, kom je heel graag en zomaar met heel veel liefde in je oortje fluisteren, dat jij de allerliefste schat bent op deze wereld!”

“Danku aan iedereen die mij ne gelukkige verjaardag hebben gewenst! xx”

“Als ge dat met u paard kunt dan kun ge pas spreken van een team!”