FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Analysis of Emotions in Text-Based Negotiations

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics
066 926

by

Günther Pfeffer, MSc
Registration Number 1128594

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Assistant Prof. Mag.rer.soc.oec. Michael Filzmoser, PhD

Vienna, 21.04.2015

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Günther Pfeffer, MSc
Pichl 21, 2871 Zöbern

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Source Code und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Abstract

The present work deals with automated emotion recognition in text-based negotiations. As such, a number of possibilities are considered before experiments are conducted using exemplary implementations of applicable methods. The foundation for the corresponding experiments is a given dataset generated by negotiations between two fictitious companies in an experimental setup. Each negotiation message in the dataset comes with values for valence and activation according to Russell's circumplex of affect, which are generated by Multidimensional Scaling. Derived from these two values, class labels for individual document instances (negotiation messages) are generated with respect to radius and location on the bipolar, two-dimensional space. Text analysis is conducted in four major phases based on the framework by Aggarwal and Zhai. Thus, essential preprocessing and document representation aspects are taken into account before, finally, learning methods are chosen. In terms of preprocessing, approaches concerning stopword removal, tokenization, stemming and Part-Of-Speech tagging are explored, while for representation purposes, Bag-Of-Words using Term Frequency/Inverse Document Frequency weighting - also in interaction with Part-Of-Speech tagging - is found to be a promising constellation. In total, 16 experiment settings are put together and applied in combination with supervised learning methods. Particularly, representative algorithms of decision tree, probability-based, Support Vector Machine and proximity-based classifiers are determined for subsequent experiments. Empirical exploration is conducted using the WEKA toolkit, where J48, Naive Bayes Multinomial, Sequential Minimal Optimization, and Instance-Based k Learner are the respective implementations of the classifier families mentioned above. For activities relating to Part-Of-Speech tagging, the Stanford Part-Of-Speech tagger is utilized.
To summarize, experiments employing 10-fold cross-validation reveal that the probability-based and the Support Vector Machine approaches are capable of achieving performance measures above 50% in terms of accuracy, precision, recall and F-score, while decision tree and proximity-based variants settled at around 40% in the best case. However, this is still almost double the baseline value of 21.51%, the share of the most frequent class occurring in the training set. In particular, experiment settings considering unigrams and bigrams as features boosted performance of the two better performing learning methods, which delivered the best results when combined with stemming. This, though, turned out to be a general tendency unless a Part-Of-Speech adjusted dataset is used, which is prepared such that it only consists of nouns, verbs and adjectives. Furthermore, stopword removal is generally found to have a negative impact on classifier performance, as features with high discrimination power in terms of Information Gain are neglected. In contrast, the application of Porter's stemmer and bigrams leverage classification results regardless of the particular learning method employed.

Kurzfassung

Die vorliegende Arbeit behandelt das Thema rund um die automatisierte Erkennung von Emotionen in text-basierten Verhandlungen. Dementsprechende Möglichkeiten werden von theoretischem Standpunkt aus erörtert und im Anschluss exemplarisch implementiert. Dem zugrundeliegend wird ein Datensatz herangezogen, welcher textuelle Nachrichten von Verhandlungen zwischen zwei fiktiven Unternehmen beinhaltet. Jede Nachricht dieses Datensatzes ist mit mittels Multidimensional Scaling ermittelten Werten für Valence und Activation versehen, entsprechend dem sogenannten „Circumplex Model of Affect“ nach Russell. Basierend auf diesen Werten werden für sämtliche Nachrichten Klassen je nach Radius und Lage im bipolaren, zweidimensionalen Raum des Circumplex abgeleitet. Die notwendigen Schritte der Textanalyse werden in Anlehnung an das Textanalyse-Framework von Aggarwal und Zhai evaluiert. Bezüglich der Aktivitäten im Rahmen des Preprocessing werden Ansätze der Tokenisierung, der Entfernung von Stopwords, des Stemming und des Part-Of-Speech Tagging untersucht. Repräsentiert werden textuelle Nachrichten mittels der Bag-Of-Words Methode, wobei die Häufigkeiten der Features durch Term Frequency/Inverse Document Frequency gewichtet werden – was auch in Kombination mit Part-Of-Speech Tagging als fruchtende Konstellation erscheint. Insgesamt ergeben sich 16 experimentelle Setups, welche mit ausgewählten Supervised Learning Methoden kombiniert und durchgeführt werden. Dabei werden Algorithmen bzw. entsprechende Implementierungen angewandt, welche auf den Konzepten von Decision Trees (J48), Wahrscheinlichkeiten (Naive Bayes Multinomial), Support Vector Machines (Sequential Minimal Optimization) und Ähnlichkeiten (Instance-Based k Learner) basieren. Für deren Umsetzung werden das WEKA Toolkit und die Stanford Part-Of-Speech Tagging Library herangezogen.
Mittels 10-Fold Cross-Validation kann eine Performance von mehr als 50% Accuracy, Precision, Recall und F-Score in Experimenten mit Naive Bayes Multinomial und Sequential Minimal Optimization beobachtet werden. J48 und Instance-Based k Learner basierte Szenarien kommen hingegen bestenfalls auf etwa 40% Accuracy. Diese Werte liegen jedoch immer noch etwa 20% über der Baseline von 21,51%, dem Anteil der Klasse mit den meisten Nachrichten im Trainingsset. Die Ergebnisse zeigen, dass vor allem Naive Bayes Multinomial und Sequential Minimal Optimization durch Berücksichtigung von Uni- und Bigrams als Features bei gleichzeitiger Anwendung des Porter Stemmers überdurchschnittlich gute Ergebnisse liefern. Ausschließlich aus Nomen, Verben und Adjektiven bestehende Datensätze sind hingegen nicht förderlich für die Emotionserkennung, da viele Wörter mit hohem Unterscheidungspotential hinsichtlich der Emotionsklassen - gemessen anhand von Information Gain - eliminiert werden. Ähnliches gilt auch für den Einsatz von Stopword-Listen. Dagegen offenbaren Stemming und der Einsatz von Bigrams positive Effekte unabhängig vom angewandten Algorithmus.

Abbreviations

ARFF Attribute-Relation File Format

BOW Bag-Of-Words

CLI Command Line Interface

CMC Computer Mediated Communication

CRF Conditional Random Field

CSV Comma-Separated Values

CSS Custom Stylesheets

DSS Decision Support System

ECUE E-mail Classification Using Examples

ERG English Resource Grammar

FAR False Acceptance Rate

FN False Negative

FP False Positive

FRR False Rejection Rate

GUI Graphical User Interface

HCI Human-Computer Interaction

IBk Instance-Based k Learner

IE Information Extraction

IF Information Filtering

IG Information Gain

IR Information Retrieval

ICT Information and Communications Technology

kNN k-Nearest Neighbor

LSA Latent Semantic Analysis

LIWC Linguistic Inquiry and Word Count

MaxEnt Maximum Entropy

MDS Multidimensional Scaling

NAA Negotiation Agent Assistant

NAVA Noun Adjective Verb Adverb

NBM Naive Bayes Multinomial

NLP Natural Language Processing

NRC National Research Council Canada

NSA Negotiation Software Agent

NSS Negotiation Support System

PMI Pointwise Mutual Information

POS Part-Of-Speech

PTB Penn Treebank

SMO Sequential Minimal Optimization

SNoW Sparse Network of Winnows

SVM Support Vector Machine

TEP Text-based Emotion Prediction

TF-IDF Term Frequency/Inverse Document Frequency

TN True Negative

TP True Positive

VSM Vector Space Model

WSD Word Sense Disambiguation

Contents

1 Introduction **1**
- 1.1 Problem statement . . . 1
- 1.2 Aim of the work . . . 2
- 1.3 Methodological approach . . . 2
- 1.4 Structure of work . . . 3

2 Literature review **5**
- 2.1 Electronic negotiations . . . 5
- 2.2 The essence of emotions . . . 7
- 2.3 Emotions in electronic negotiations . . . 8
- 2.4 Approaches to measure emotions in written text . . . 9
  - 2.4.1 Multidimensional scaling . . . 10
  - 2.4.2 Knowledge-based emotion assessment . . . 11
  - 2.4.3 Machine learning based approaches . . . 14
  - 2.4.4 Machine learning in sentiment analysis . . . 17

3 Method **19**
- 3.1 Preliminary data analysis . . . 19
- 3.2 Algorithm evaluation & selection . . . 20
  - 3.2.1 Characteristics of text mining . . . 20
  - 3.2.2 Text analysis framework . . . 20
  - 3.2.3 Text preprocessing . . . 21
  - 3.2.4 Document representation . . . 24
  - 3.2.5 Knowledge discovery - algorithm selection . . . 27
  - 3.2.6 Building classes . . . 31
  - 3.2.7 Validation of machine learning results . . . 33
  - 3.2.8 Summary . . . 35
- 3.3 Tool selection . . . 36
- 3.4 Initial data preparation . . . 40

4 E-negotiation data analysis **45**
- 4.1 Negotiation experiment . . . 45
- 4.2 E-negotiation system . . . 46

4.3 Negotiation transcript description 48

5 Results & discussion **51**
5.1 Preliminary assessment towards emotions 51
5.2 Standard descriptive statistics 54
5.3 Learning results 56
5.3.1 Preliminary notes 57
5.3.2 Probability based classification 58
5.3.3 Decision tree based classification 60
5.3.4 Support Vector Machine based classification 61
5.3.5 Proximity based classification 63
5.3.6 Feature selection 64
5.3.7 Performance comparison & results reasoning 67
5.3.8 Class specific performance 69
5.4 Recommendations 71

6 Conclusion & outlook **73**
6.1 Main findings 73
6.2 Limitations 75
6.3 Future work 75

List of figures **79**

List of tables **80**

Bibliography **81**

CHAPTER 1

Introduction

The introduction chapter gives a brief overview of the fundamental topics and disciplines of the present work, and outlines the objectives, this thesis is trying to achieve. Furthermore, the chosen approach in order to fulfill the declared goals is pointed out, together with the structure of subsequent chapters.

1.1 Problem statement

Nowadays, there can be no doubt that negotiations are more than just a straightforward process with an economic outcome at the end. Communication, and therefore negotiation, is not suited to a simple, standardized evaluation since each individual interprets the semantics of words, behavior and emotions differently [9]. In addition to the individual's perception of communication, communication can happen in four different aspects, where each aspect potentially conveys a different meaning - content, self-revelation, appeal and relationship [84]. Things become even more complex when taking the underlying structure of interactions into account. As such, negotiations furthermore depend on factors including the means, the topics discussed, synchronicity, the number of parties and their roles [79]. Focusing on text-based negotiations, some special circumstances need to be considered with respect to the factors mentioned above. E-Mail, as a representative form of a text-based communication channel, reduces the extent of communication capabilities in a way that important communication aspects such as mimic, gestic, voice etc. are absent compared to a face-to-face situation. Consequently, negotiating parties are not able to sufficiently compensate for this lack in written form, which is very often the cause of conflicts and escalations [27]. Going one step beyond e-mail as a text-based communication and negotiation system, Negotiation Support Systems (NSSs) are utilized for guiding and optimizing negotiations. However, although negotiation courses and outcomes are significantly influenced by emotions [21], NSSs mainly still focus on the economic track of negotiations while tending to ignore the effects of emotions conveyed in text-based messages exchanged [44].
This work should formulate the foundation for closing the gap in the regards mentioned. As

such, next generation NSSs can be enhanced by the integration of outcome estimation based on emotion recognition. Ideally, a real-time analysis of emotions supports each individual negotiation situation by predicting negative (e.g. the escalation of a conflict) or positive (e.g. cooperative attitude, aiming for win-win outcomes) trends. With such capabilities in place in an NSS, negotiators are enabled to take appropriate action to remediate the current direction of a negotiation.

1.2 Aim of the work

Taking the problem statement above into account, the final step in a broad field of investigation is the enhancement of NSSs in terms of optimization of the economic and especially relationship outcome of negotiations. However, the scope of this thesis is not to enhance or develop an NSS with emotional patter recognition capabilities. The thesis focuses on the necessary pre-work for such next generation NSSs by addressing two objectives of research:

1. *What are possibilities for automated emotion recognition within the scope of text-based negotiations?*

2. *Selection and exemplary implementation of applicable methods to determine their accuracy on real negotiation data.*

As such, this work deals with the machine learning supported analysis of emotions and the patterns of those emotions in text-based negotiations with respect to the outcome of a certain negotiation. After extensive literature review, an evaluation of text mining mechanisms reveals the best choice of mechanisms from a theoretical perspective. The implementation of the selected methodologies proves the theoretical assessment and ideally deliver accurate results in the field of emotion recognition.

1.3 Methodological approach

The research model applied for this thesis is a rather constructive one. The open problems derived from a real world situation are mentioned in the section 1.2. To reach a solution that fulfills the research objectives, it will be necessary to utilize established concepts of business intelligence and affective science. The foundation for a solid scientific investigation of the problem statement is a phase of analysis of relevant materials already in existence. Literature regarding emotions in e-negotiations, machine learning based approaches of emotion recognition and related topics are the key artifacts to be considered in this phase. During the successive design phase, evaluations and preparations for the empirical part of the work are conducted. The final evaluation phase represents this empirical part by implementing carefully selected methods in order to fulfill the research objectives. A more detailed view of the approach is itemized as follows:

1. **Literature research and study** At the beginning it is necessary to examine and review the existing literature regarding emotional patterns and emotions in negotiations in order to address the first research objective. Furthermore, for both research objectives the examination of literature dealing with text mining, machine learning and related topics is necessary.

2. **Evaluation and definition of empirical approach** After literature research, the detailed description of the approach for automated recognition of emotional patterns is given. As for this thesis, the most promising setup of empirical research and experiments should be applied; an evaluation of potential approaches and algorithms make clear, how further exploration of the addressed problem is conducted. For instance, it makes sense to take algorithms from various classification families into account during that evaluation phase, e.g. Naive Bayes, proximity and decision tree based approaches. Also, unsupervised learning methods like association rule learning and clustering approaches (hierarchical, density-based, partitional etc.) might be relevant.

3. **Data preparation and analysis** The provided negotiation data is evaluated and prepared for the application of the selected text mining mechanism. Data definitely used for this purpose will be coming from an experiment executed in 2012. Thus, certain characteristics of qualitative and quantitative nature are obtained, in order to employ the provided dataset in declared text mining processes properly. The usage of text-based data only seems like a limitation of scope of the research objective because all dimensions of communication are packed into the text exchanged via an NSS. However, in the end this is a necessary precondition for the application of specific text mining methods on various sets of training data.

4. **Implementation of methodology** After the selection and compilation of applicable approaches and methods, the next step deals with the experimental investigation using prepared negotiation data. At a first glance this seems trivial, however, it is not since the optimal experiment setting and parameterization of data mining algorithms in general is an essential task. Consequently, results obtained in the experiments are compared and interpreted such that appropriate recommendations for automated emotion detection in negotiation messages can be derived. One step before the actual implementation, the selection of tools and software installations needs to be evaluated to a certain extent.

1.4 Structure of work

According to the methodological approach described above, the structure of this work is such, that after this introduction part, it starts with a comprehensive literature review in Chapter 2. In this chapter the topics of e-negotiations, emotions and emotional patterns, and data mining are explored. Corresponding subsections consist of relevant findings of the fields of investigation already mentioned. In Chapter 3, promising approaches for recognition of emotions and emotional patterns are evaluated and selected for application in subsequent steps. It additionally deals with the preparation of the provided negotiation data set in order to have clean data to which selected

data mining algorithms can be applied. A detailed look on the provided dataset and its origin is taken in Chapter 4. Chapter 5 deals with the outcome of the implemented approach defined in Chapter 3. As such, results of dataset investigation, as well as comparison and reasoning of learning method performances are outlined. The work concludes with a discussion of results and gives indications about possible limitations and potential future work in Chapter 6.

CHAPTER 2

Literature review

As this work deals with several topics including electronic negotiations and machine learning, the purpose of this chapter is to give an overview of relevant concepts in related disciplines. Consequently, existing research results regarding electronic negotiations, emotions and their measurement are covered in the following sections.

2.1 Electronic negotiations

This section is about Computer Mediated Communication (CMC) and especially negotiations supported by NSSs. It also handles the special aspects surrounding computer-mediated negotiations.
In global business, CMC has become more and more popular in recent decades. This is hardly surprising as this form of communication has enabled people to instantly communicate and negotiate independently from their geographical location at low cost [64]. As Information and Communications Technology (ICT) has evolved, several approaches have been implemented to deal with conflict management, search for consensus and similar topics [42]. In this regards, Kersten and Lai examined the field of computer-based support systems and concluded that an NSS is characterized by five major attributes with respect to its users: independence regarding decision-making power; interdependence regarding participants' objectives; possibility of sharing interests; power to stop the negotiation process at any time; and finally the ability to propose, reject and request offers. These aspects represent what separates an NSS from other support systems such as Negotiation Software Agents (NSAs), Negotiation Agent Assistants (NAAs) and Decision Support Systems (DSSs). The latter, however, is a sub-component of an NSS as those systems help to define objectives and search for solutions. The gap NSSs close is related to the communication and coordination abilities that guide participants through the negotiation process and let them interact accordingly. Additionally, classification of NSS tools can be done from different perspectives. In terms of activeness, negotiation software can range from passive systems, which require users to take full control over their actions, to proactive intervention-mediation

systems, which actively support users (e.g. in problem-solving and concession-making) and evaluate their planned and taken steps. Furthermore, NSSs can be categorized by the roles they occupy in the negotiation situation in which they are utilized and by their main focus during a certain phase in the negotiation process [42].
With the establishment of software-supported negotiation systems, questions have arisen concerning deviations from common, face-to-face negotiation processes. Accordingly, Pesendorfer et al. investigated the behavior of participants in electronic negotiations throughout the negotiation process. The authors concentrated their research especially on the perception of conflict with respect to a two-phase [85] and a four-phase [3] negotiation model, which were initially designed for face-to-face negotiation scenarios. Taking the former model into consideration, it could be ascertained that the first stage is dedicated to exchange of information such as priorities and needs, which is typical for the first phase, the so-called differentiation phase. In the second stage - known as the integration phase - negotiators were used to exchanging solutions and offers, which was found to be an indication that the participants are trying to reach an agreement. More detailed results were revealed when examining the four-phase model. The first phase showed the negotiators exchanging a high number of off-task comments and displaying affective behavior, which indicates their endeavor to manifest their interpersonal relationships. Only after this phase was completed, negotiations participants moved on to exchange their interests in phase two, which was therefore clearly more competitive than the first phase. Phase three, in turn, was characterized by more and more effort invested in order to achieve satisfying results. This was supported by the finding that pieces of information exchanged plummeted while the number of offers increased. Consequently, Pesendorfer et al. were able to confirm that both the two-phase and the four-phase model apply to electronic negotiations [63].
Pesendorfer and Koeszegi explored the effects of communication modes on electronic negotiations, especially with respect to behavioral styles. As such, the two modes - synchronous and asynchronous - were analyzed regarding their unique properties. While asynchronous electronic communication is known for revisability and reviewability, its synchronous counterpart is additionally characterized by co-presence, co-temporality and simultaneity. Taking these attributes into account, the asynchronous mode can be compared to exchanging mails or letters, whereas the synchronous mode can be represented as being similar to informal chat. Experiments were conducted under the following assumptions: in synchronous electrical communication scenarios, participants show competitive, offensive and unreflective emotional behavior due to time pressure. Asynchronous communication, on the other hand, supposedly lets communicators reflect their emotions and apply problem-solving attitudes as they have more time to spend on their actions. These key assumptions were confirmed by the results of a laboratory experiment. Therefore, the asynchronous communication mode can be utilized in order to encourage integrative, problem-solving behavior, whereas synchronous setups lead to affective and less friendly behavior. The authors explain the differences between the two modes by the reciprocal expectation of near-instant responses in synchronous communication, which is to say, as mentioned above, by time pressure [64].

2.2 The essence of emotions

In order to be able to discuss the influence of emotions in negotiations, a basic understanding of emotions is required. Thus, this section provides information about the nature of emotions.
Many researches have tried to outline the fundamental concepts that define and explain emotions. A detailed investigation in this regard was undertaken by Cowie et al. The phrase "Plato's middle ground" was identified as an approach to give the domain of emotions a name without making clear commitments regarding boundaries. The so-called middle ground comes from Plato's theory that the mind consists of three parts: appetite, reason and spirit, the part connecting appetite and reason, which for example is expressed by anger. However, confusion arises when the latter part should be clearly defined, especially when it comes to distinguishing between emotion, affect, feeling, passion and expression. The mentioned terms describe the domain from different perspectives and are used under various circumstances. Thus, in contrast to feeling, expression is rather objective and observable, and passion is defined as an emotional state where feeling exceeds reason. A special form of the expression of internal feelings is affect. Another major concept of Cowie et al. is the "emotional life", which mainly consists of (internal) feelings and (external, observable) expression and describes the emotional portion of human life. The dimensions involved in the description of the nature of those emotions are of three basic kinds, namely valence, activation and potency. That the feelings involved are either negative or positive is represented by valence. Activation indicates an individual's potential to act, while potency deals with one's capability to handle events faced. An emotion appraisal approach is defined as a sequence of checks by Sander et al. [69]. The initial step involves a check of relevance of emergent emotion, which is followed by an assessment of implications. As a third step, one evaluates the options for coping with a situation before possible outcomes are finally checked against personal and social values [16].
In order to structure the "emotional life" further, a taxonomy of emotion-related states was established. At the core of this idea are the generic states given by Scherer et al. [72]. The list of states consists of emergent emotions, interpersonal stances, moods, attitudes and affective dispositions, but was extended by further states including various stances and established emotion. The study revealed that people are rarely either in states without any emotion or in states consisting of pure emotion. Indeed, most often people are in states between those two positions, which can be defined as, for instance, mood or an altered state of emotion [16].
Ekman extracted six basic emotions - anger, disgust, fear, happiness, sadness and surprise [22]. These basic emotions seem to have specific characteristics, which on the one hand distinguish emotions from phenomena such as mood, and on the other hand allow a differentiation between certain emotions. However, some features are mentioned as being very common to all basic features, for example, rapid onset, brief duration and automatic appraisal. Taking the defined characteristics into account, Ekman suggested twelve further candidates for the list of basic emotions, including guilt, shame, relief and satisfaction. Ekman's perspective fundamentally differs from other theories, relying on the belief that emotions are essentially the same, but are separated by intensity or pleasantness [22]. In terms of basic emotions Izard identified a set of ten emotions representing the full spectrum of emotional classes: anger, contempt, disgust, distress, fear, guilt, interest, joy, shame and surprise [39]. Thus, the question regarding the ex-

istence of something like a comprehensive list of basic emotions has been intensively examined by different researchers, who have also been active in the discussion surrounding the term "basic" in this context. Ortony and Turner compared different theories including very minimalistic approaches to basic emotions (such as the one from Weiner and Graham consisting only of happiness and sadness [87]), but also extensive theories, such as the one supported by Izard. They furthermore conclude that an abstraction of emotions to a basic level may not lead to appropriate results as emotions should rather be viewed as a compilation of constituents, which are basic, but not necessarily emotions themselves [60].

2.3 Emotions in electronic negotiations

Combining the subjects of the sections above, this paragraph covers research regarding emotions in the context of electronic negotiations.
Emotions were found to be a driver for improving or threatening negotiations and the resulting relationships. Those emotions are determined by a set of variables defining the negotiation process and the context [21]. For instance, happiness is an emotion that is, in general, considered to be positive for a negotiation trail. However, if a negotiator fakes happiness the counterpart may be suspicious of the actual intentions. As such, emotions can be interpreted from different perspectives. Firstly, the social aspect of emotions explains how relationships and further interaction are established and maintained. Research has shown that positive emotion boosts negotiation indicators such as trustworthiness and increased joint outcomes. However, the contextual setting, or rather the power of a negotiator, cannot be neglected. Secondly, emotions can be distinguished by their certainty, meaning that a negotiators' - positive or negative - emotional mindset influences the perceptions of emotions faced in a negotiation in a positive or negative way. A third aspect deals with gender specific emotional expression. Research has revealed that women express a broader range of emotions, but some specific emotions are more often found in men, e.g. anger and pride [21].
Obeidi et al. deal with emotional effects in conflict situations [56]. They identified Game Theory as an approach to decision-making and conflict resolution, which involves all decision makers, their opportunities and their preferences. However, Game Theory does not consider emotional aspects since relational parameters are not considered. To overcome this absence, a conflict is defined as process rather than state, characterized by different perceptions, incompatible objectives of relevance and interference of interests. The appraisal theory for emotion activation by Lazarus furthermore explains how emotion is induced and evaluated by individuals. Primary appraisal components cope with the question of personal relevance, i.e. whether emotions are triggered or not. Secondary appraisal components are associated with the handling of a situation, i.e. which emotions are activated [46]. As such, Obeidi et al. connects the revealed characteristics of conflicts to Lazarus' appraisal model and determines anger, frustration and fear as key emotions in conflicts. As for their resolution, conflicts are further represented as a graph model, which consists of so-called states. Those states represent combinations of technically feasible options for all decision makers involved. Feasible states in the final graph are then identified as recognized, potential or acceptable. Recognized conflict states are those that consist of commonly known and acceptable choices to solve a conflict. A potential state can be reached by the

application of correct positive emotional actions, while hidden states are highly unlikely options due to a persistent conflict caused by mainly negative emotions [56].
Sokolova and Szpakowicz have studied to what extent language patterns help to implement negotiation strategies for the purposes of which they applied classification methods for early negotiation outcome prediction. A basic concept that was considered is the "weak get strong" effect, which describes the tendency of e-negotiators to emphasize risk and aggression in comparison to a face-to-face situation, although the emotional style is less present. In fact, the tactics employed in negotiation strategies are rich in emotion, exchanged between the participants through written messages [80].
Hine et al. highlights that there are obvious differences between face-to-face and distributed interactions, accomplished via electronic communication channels [37]. Specifically, a great portion of unspoken information may be lost, e.g. context and initial perceptions. The concepts of cues filtered in and cues filtered out are found to be applicable to CMC, which is to say that rich communication information is either neglected or approximated when transferred via electronic channels. As such, messages in CMC contain not only cognitive information, but also emotional information which together influence negotiation aspects such as decisions and strategies. The investigation further deals with emotion and tone of language in an e-negotiation situation by looking not only at what is being communicated, but also how something is being communicated can influence the meaning of messages [37].
As an extension to the study from Pesendorfer and Koeszegi [64], Koeszegi et al. examined the influence of synchronous and asynchronous communication modes on e-negotiations [43]. The latter is more likely to establish successful electronic negotiations, since this mode tends to produce friendlier and less competitive negotiations than those accomplished using a synchronous communication mode. This observation is mainly explained by the time pressure exerted in a synchronous setting. As such, synchronous communication often leads to heated discussions while calmer conversations can be expected when communicating asynchronously. the authors suggest, therefore, that the the asynchronous communication mode whould be considered when a topic to be discussed has the potential to trigger emotions in an extensive way [43].

2.4 Approaches to measure emotions in written text

Now, taking the objectives of this work into consideration, a key matter is the recognition of emotion in written text, specifically in text-based negotiations. As such, this section gives an overview of methods appropriate to the analysis of emotions in texts. The following subsections are structured regarding the underlying concepts of measurement. Figure 2.1 illustrates at a glance how the subsequent methodologies of emotion measurement can be differentiated. From the perspective of automation, there are principally three families of measurement, namely manual, knowledge based and machine learning based. While the manual method is represented by Multidimensional Scaling (MDS), the two latter measurement groups are further divided. Apart from the chosen categorization with respect to automation, other approaches describe measurement methods from different perspectives, for instance, from the perspective of the research approach. As such, content analysis via MDS is empirical, while dictionary and rule based methodologies can be interpreted as deductive as new knowledge is derived from existing facts.

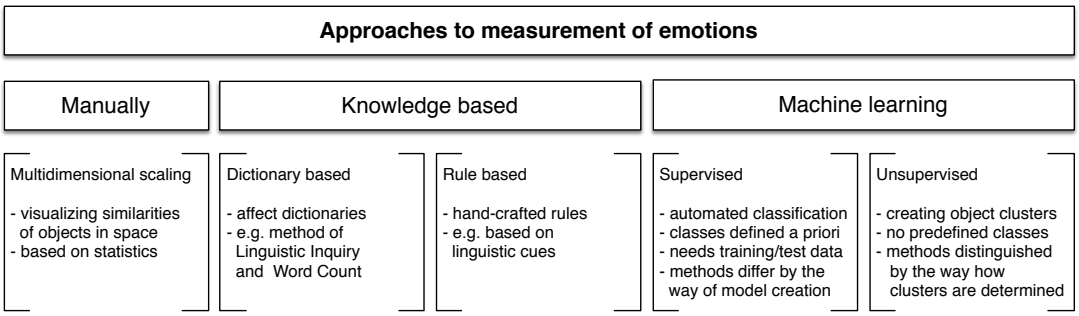

Figure 2.1: Categorical overview of approaches found to adequately determine emotions in texts from an automation perspective

Machine learning, on the other hand, can be seen as an inductive methodology as classification and clustering, applied to real-world data, are capable of deriving corresponding models and, consequently, theories.

2.4.1 Multidimensional scaling

MDS is a technique that visualizes similarities/dissimilarities (in other words, proximities) of data objects based on statistics. As such, objects are represented as points in a low n-dimensional space, where the distance between the points reflects (dis)similarity. Applications of MDS can be useful in various disciplines, such as sociology and economics, usually as a method for exploring and understanding data. However, MDS can also be applied when reducing the number of dimensions, for example, in a scenario where an object's attributes are put on a high number of dimensions, MDS helps to represent the distances in a low dimensional space. In such a case, the fit of the computed configuration decreases, i.e. the so called "stress" as a measure of fit increases. Thus, the level of stress can be an indicator of a proper amount of dimensions, although for interpretation purposes up to three dimensions are selected. MDS not only works with known, metric (dis)similarities, but also in the case that objectives have been ranked in a certain order, namely by assigning numerical numbers that represent the order in the best way [32] .

This concept was evaluated by Okada and Takeuchi in an experiment dealing with differences of fonts. Eight fonts served as stimuli, whereas all 28 possible pairwise combinations were printed on cards. 16 participants then individually ranked the 28 cards according to their perception about dissimilarity of the fonts on each card. The objective of MDS was to obtain the configuration of test subjects regarding their judgments. Dissimilarities between each pair of participants could then be calculated using Kendall's rank correlation [1]. As a stress value of 0.233 was considered to be acceptable, the objects were put in an unidimensional space. According to the resulting configuration and the determined characteristics of font, the authors concluded that fonts were distinguished based on blackishness and style rather than on presence of serifs or italics. The MDS space, which is an axis in the unidimensional case, reveals which subjects consider differences of blackishness, style or both to be important when judging dissimilari-

ties [57].
In order to find emotions in messages of negotiations, Griessmair and Koeszegi applied a data-driven approach without utilizing a predefined taxonomy of emotions [31]. Specifically, negotiation messages were analyzed by MDS on three levels: the level of the overall negotiation, the level of a message, and the level of single statements. After test subjects divided negotiation statements up into different piles by emotional nature, two main dimensions were determined to properly classify full negotiation messages, namely valence and arousal. Valence is associated with the level of friendliness and related labels used for the mentioned piles (e.g. "impolite", "aggressive"). Arousal is interpreted as fact-orientation and, furthermore, labels such as "co-operative" and "compromising". The two dimensions in combination result in four additional emotional message styles. Firstly, the style of "personal relation" combines friendliness with low fact-orientation and concentrates on a good relationship. Secondly, at the opposite end to the dimension "personal relation", "impersonal transaction" stands rather for unpleasant, highly fact-oriented messages. When the same messages are not considered fact-oriented, they are perceived to be of the kind "resignation/termination". Finally, a "cooperation" message style describes highly fact-oriented messages, expressed in a friendly way. Similar to the message level, the fine-grained level of single utterances is properly reflected on a two-dimensional map. The dimensions in this case are "assertiveness" and the level of integration behavior. Depending on the direction the phrase takes within the two dimensions, a statement is formulated and perceived either positively or negatively in terms of emotions. Finally, the evolution of emotions on a message level was investigated by extracting emotional patterns of successful and unsuccessful negotiations. It was found that unsuccessful negotiations end with messages of the style "resignation/termination", while successful negotiations end with messages of the kind "personal relation", which can be interpreted as willingness of the negotiation partners to build up long term relationships. Regardless of the fact-orientation, successful negotiations tend to contain mainly friendly messages; unsuccessful negotiations, on the other hand, include message styles associated with unfriendly and negating messages. All the negotiations explored utilized a broad range of emotions, though successful negotiations tend to return to a friendly style quickly after drifting to rather more conflicting phases in negotiations [31].

2.4.2 Knowledge-based emotion assessment

This kind of emotion analysis relies on predefined information that helps to exploit the occurrences of words in a text [81]. The given knowledge can be of various different natures. However, the methods in this section utilize on the one hand special kinds of dictionaries containing words and their affective meaning, and on the other hand certain sets of rules, e.g. rules regarding linguistic cues, which are applied in order to extract information about emotions in texts. A particular method of the former procedure is Linguistic Inquiry and Word Count (LIWC). In reality, LIWC is a program consisting of two components, namely a processing unit and a manually crafted dictionary. The processing component goes through input documents and examines each word by looking it up in the dictionary. Besides lexical information, the dictionary also stores indicators for positive or negative emotion for the corresponding words. Once processing of a document is finished, a report is generated showing the usage rates of each category (e.g.

positive emotion, negative emotions, swear words, past tense) [82]. Consequently, the results of knowledge-based content analysis can be used to indicate valence as shown by the experiments below reveal.

Hine et al. employed the technique of LIWC in order to determine valence in text. By applying this method, hypotheses regarding emotion, language and pronouns were created and assessed with respect to the success of the examined e-negotiations. In order to do so specific words were associated with predefined LIWC variables (e.g. "happy" and "joy" were associated with "positive emotion"). As a basic data foundation, a set of multi-issue negotiation data, generated mainly by students negotiating the buying and selling of bicycle parts. Inspire[1] was the tool utilized for the distributed negotiations, resulting in more than 2,500 negotiations for assessing the hypothesis. In terms of emotions, it was found that successful e-negotiations contain significantly more emotionally positive expressions than unsuccessful e-negotiations. On the other hand, this is not true for negative emotions used in e-negotiations, i.e. words associated with negative emotion are not used more often in unsuccessful negotiations than in successful negotiations. However, the results may be biased due to the laboratory setup of the negotiation data; it is possible that a more natural setting could reult in different emotional intensities [37].
Not directly in the context of e-negotiation but rather in non-negotiation communication, Hancock et al. explored the field of emotional expressions in text-based interaction [36]. Three aspects were investigated in detail, the first of which aimed to describe the strategies used to express positive and negative emotion. Secondly, it was investigated whether a communication partner is able to detect another's emotional state by their written messages. Finally, based on the finding that women may be more sensitive to emotional states in face-to-face conversations, the third aspect deals with gender specifics in the context of text-based communication. The analysis was based on a study, that collected communication data of governed dialogs between students. After the dialog, an additional surveys was completed in order to assess speakers and listeners' emotional expressions and states. Besides the manual assessment via surveys, the conversations were examined using LIWC. Besides the finding that gender does not influence emotional expression in text-based interaction, expressers stated that they tried to convey positive emotions through increased punctuation, quick responses and emphasized agreement with the communication partner. From a linguistic perspective, analysis indicates that negative expressers use more negations and terms showing negative feeling, while positive expressers use exclamation marks a lot more and generally use more words. However, when assessing the emotional state of the communication partner, test subjects rely significantly on negations and exclamation marks. Limitations in this study were pointed out to be on the one hand a restricted set of emotional dimensions (i.e. positive versus negative emotions) and, on the other hand, the unnatural communication data generation, as seen in the work of Hine et al. mentioned above [36].
As an extension to the study of Hancock et al., Gill et al. explored emotion in CMC in a more fine-grained way, i.e. not only classifying emotions to be positive or negative, but applying eight main categories (fear, surprise, joy, anticipation, acceptance, sadness, disgust, an anger). Furthermore, instead of an artificial interaction scenario, the study from Gill et al. relied on blog posts, which were assessed by raters in terms of kind and strength of emotion. The goal was

[1] http://invite.concordia.ca/inspire/about.html, 17.09.2014

to determine the ability of test subjects inexperienced in emotional rating in comparison with expert raters. Results showed there was strong agreement between expert and naive raters regarding emotions with high valence, namely anger, disgust, joy and anticipation. Raters did not agree particularly strongly on the remaining emotions, especially when the rating was based on a shortened version of the investigated text [29].
In the research field of emotion detection, the identification of the cause of emotion constitutes an important topic. Lee et al. propose a text-driven, rule-based approach in this regard. The focus of the authors was clearly to analyze causes of emotion as only explicitly expressed emotions in the text corpus were taken into account for their research. As key components in their investigation, linguistic cues were examined (positions of cause events, experiencer and emotion keyword, action verbs, conjunctions, etc.). Generalized rules derived from those linguistic cues were the basis for a rule-base system. Preliminary to the actual experiment, dedicated annotators marked emotion keywords and corresponding cause events, which were required for determining linguistic cues, such as positions of parts of a sentence. The subsequent experiments - which were executed manually - showed that the extracted rules were quite effective in determining emotional causes as the F-score of 51.66% compared to the baseline of 30.83% shows, while F-score is defined in 2.1 [47].

$$F\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{2.1}$$

The motivation of Strapparava and Mihalcea to analyze emotions in text is based on possible use cases in the fields of sentiment analysis, computer assisted creativity and verbal expressivity in Human-Computer Interaction (HCI) [81]. This analysis used news headlines in order to classify them with respect to emotions by applying various methods, concentrating on the emotions of anger, disgust, fear, joy, sadness and surprise. In particular, five different methodologies were applied, where one set relied on knowledge-based emotion annotation and the second set was represented by a Naive Bayes classifier. The knowledge-based mechanisms utilize on the one hand the WORDNET AFFECT database[2], which is a hierarchy breaking down affective concepts (e.g. emotion, mood, emotional response) into corresponding synsets [25]. On the other hand, the vector space model Latent Semantic Analysis (LSA) was applied, which gives a homogeneous representation of single words up to complete texts in the LSA space. The Naive Bayes classifier mentioned above - clearly a machine learning mechanism, but mentioned in this section to maintain the experiment context - is trained on blog posts and therefore relies on a corpus-based approach. The different methodologies expose distinct strengths and weaknesses. The WORDNET AFFECT method performs well in terms of precision, while LSA based methodologies dominate in terms of recall. Regarding individual emotions, Naive Bayes delivered good results for the emotions that were annotated the most in the blog post training data. In all other respects, the LSA models performed best [81].

[2] http://wndomains.fbk.eu/wnaffect.html, 15.10.2014

2.4.3 Machine learning based approaches

As manual and less complex approaches to emotion analysis have been reviewed in the paragraphs above, the focus of this section is on machine learning methodologies in the context of emotion recognition.

2.4.3.1 Supervised machine learning

Supervised learning is a concept of automated classification of objects. The classes to which the objects are assigned are defined apriori, meaning that they are given without taking the actual data into account. Such classification methods rely on models created based on a certain amount of representative training data. Ideally, training data is well diversified in terms of object attributes and is properly distributed over the given classes for accuracy reasons. The output attributes, i.e. the classes, are provided in the training data, while test data is used to evaluate the built model [33]. Implementations of supervised learning are divided into a few basic approaches; as research has shown, three of these - each being capable of processing categorical attributes - are commonly used in emotion classification. Firstly, decision tree algorithms build up a hierarchy of internal and leaf nodes, where the former represent decision points and the latter stand for the classes. The trained model is the result of creating a tree with the intention to maximize the discriminatory power of each decision node. Secondly, Naive Bayes is a probabilistic classifier based on Bayes' theorem [20]. As such, each object to be classified is part of a hypothesis stating that the object belongs to a certain class. The probability of a hypothesis to be true then determines which class an object is actually assigned to [33]. Support Vector Machine (SVM) approaches generate models based on mathematical functions. In detail, a function in the training phase reflecting a hyperplane in an n-dimensional feature space is created. While doing that, objects - represented as vectors - of the different classes are separated in such a way that the margin of the hyperplane to the objects is maximized [89].

Chaffar and Inkpen's research efforts have been into the exploration of possible ways in which to recognize Ekman's basic emotions in heterogeneous texts using supervised machine learning algorithms. In contrast to knowledge-based approaches, which use linguistic models and existing knowledge (such as taxonomies) for text classification, they applied learning mechanisms in order to build models from annotated texts. The utilized texts - five different sets in total - were chosen to cover a wide range of characteristics. As such, the five distinct datasets employed consist of headlines, fairy tales, sentences from diary-like blog posts, emotion rich sentences from blog posts and an groups of sentences extracted from various stories. The annotations included are oriented in line with both Ekman's and Izard's emotional classifications. The datasets were cleaned up in preparation for the experiments with learning algorithms, which is to say that neutral words such as "I" and "the" were removed, and stemming was applied. Additionally, contracted negations (e.g. "don't") were replaced by their full, uncontracteds forms ("do not"). The following step was to train the chosen classifiers J48, Naive Bayes and Sequential Minimal Optimization (SMO) were trained as they represent three different approaches: the classifier families of decision trees, Bayesian and SVM. The feature sets used for classification relied on three major techniques, namely Bag-Of-Words (BOW) (where each sentence is represented

as a vector indicating the words occurring in a certain sentence), n-grams, and lexical emotion features based on WORDNET AFFECT. In putting everything together, Chaffar and Inkpen were able to show that after the training phase the SVM implementation SMO performed the best in terms of accuracy and regardless of which the training dataset was used. The training of the classifiers was done based on BOW, which turned out to be the most accurate feature set after applying a trained classifier on testing datasets [14].
A series of experiments combining affect lexicons and supervised machine learning was conducted by Mohammad. Not only did he investigate the potential of classifying emotional text by applying three different lexicons, but he also compared results with the results of an n-gram approach. The affect lexicons utilized for this purpose was the WORDNET AFFECT database, a lexicon called National Research Council Canada (NRC) 6, with annotations of Ekman's six basic emotions and the extended NRC-10 lexicon including trust, anticipation, and positive and negative sentiment annotations. As training and test data, Mohammad employed the data set from Strapparava and Mihalcea mentioned above, which consisted of newspaper headlines. By applying logistic regression and SVM methods, experiments revealed that automatic methods show good performance for recognizing emotions, which are also well recognized by human raters (e.g. sadness and fear). Furthermore, a combination of n-grams and NRC-10 delivers better classification results than the n-gram approach alone, which was not true for the WORDNET AFFECT lexicon. However, the n-gram performance decreased drastically when testing the approaches on another domain, i.e. emotion labeled blog posts, due to overfitting of n-gram features. Mohammad concludes that emotion classification of sentences can properly be achieved by using word-level affect lexicons, where the performance increases with the size of the lexicon [54].
The focus of Sokolova and Szpakowicz was on language pattern recognition and predicting negotiation success using machine learning [80]. To this end, three major sentence building blocks were investigated, namely modal verbs, personal pronouns and main verbs. Modal verbs were designated the function of indicating a tactical move such as "request" or "suggestion". Main verbs, on the other hand, were categorized regarding the action they describe, e.g. "communication" or "attitude". With the utilization of personal pronouns, the level of immediacy was measured. Taking all this into account, analysis of trigram, 4-gram and 5-gram models revealed that the predominant tactical approach was suggestion by using event verbs in combination with personal pronouns. Further investigation regarding identified patterns and negotiation outcomes has shown that especially the successful negotiations (defined as a deal agreed within a certain time frame) can be predicted with an accuracy up to 85% using the classification approaches Naive Bayes, decision tree and SVM [80].
Alm et al. approached the sentence-level emotion recognition with the subsequent aim of natural sounding text-to-speech synthesis [6]. Therefore, the authors applied the Sparse Network of Winnows (SNoW) learning architecture - a supervised machine learning method - in the domain of fairy tales [13]. The experiments implemented were of two kind: determining sentences to be emotionally enriched or neutral and, secondly, identifying emotional sentences' valence (positive emotion versus negative emotion). Consequently, SNoW was fed with continuous Boolean values representing 14 different features, including the first sentence in a story, special punctuation, words solely in upper-case, sentence length in words, positive and negative word counts,

content BOW, etc. The accuracy of classification was found to be more then 63% when all features were considered. However, the experiments revealed that the data set was too small and too complex as it was put together from 185 children's stories. This weakness was especially emphasized in the second experiment, where classification performance decreased dramatically. In addition, the experiments revealed that features were not independent of each other, and varied parameters settings influenced their contributions to classification [6].

2.4.3.2 Unsupervised machine learning

Distinct from supervised learning cluster analysis is an unsupervised methodology for data analysis. Unsupervised in this context means that classes are not predefined, but are rather a result of the procedure. Therefore, an unsupervised approach generates clusters of objects with similar attribute values in common, i.e. they are within a certain distance of each other in a multidimensional environment [33]. According to Gupta clustering approaches are divided into four major groups: hierarchical methods, density-based methods, grid-based methods and model-based methods. Hierarchical methods produce a tree of clusters, which can either be generated bottom-up (aggregate small clusters), or top-down (break down one big cluster). The clusters in a density-based approach are built around dense regions in the feature space, meaning that objects need a particular number of other objects within their neighborhood in order to build a cluster. Grid-based methods separate the feature space into a grid founded on the characteristics of given data. Based on probability, distribution model-based methods rely on models, that aim for maximum similarity within clusters and low similarity between the clusters [33]. In practice, the latter method was applied as will be described in the following paragraph.

In the field of sentence-level emotion detection, Agrawal and An examined further machine learning methodologies. Taking a different route than Chaffar and Inkpen, they followed the unsupervised learning approach without the limitations of predefined lexicons and categories of emotions. Thus, Agrawal and An's methodology goes beyond classification of texts into fixed sets of emotions based on manually defined affect dictionaries and considers the context of words in a text. This also overcomes the problem of methods based on linguistic rules, which must be created and maintained accordingly. Regarding supervised learning algorithms, their objective was to examine whether unsupervised approaches perform better when applied to different domains. The unsupervised procedure consists of four basic steps. It starts with simply extracting so-called Noun Adjective Verb Adverb (NAVA) words, which are those bearing affect. The NAVA words were then used in order to calculate emotion vectors. Those vectors, however, rely on the calculation of semantic relatedness of word tuples, determined by Pointwise Mutual Information (PMI), a measure of similarity based on probability of co-occurrence [15]. Further precision is gained by the introduction of syntactic dependencies between words, for which combinations of words were analyzed in order to reveal inter-word relationships and their influence on one another. In the final step the emotion vector of a sentence is computed by simply aggregating and averaging the vectors of the NAVA words it contains. Compared to supervised and other unsupervised methods, the unsupervised approach of Agrawal and An produces the

best results across the tested, stemmed datasets in terms of F-score [5].

2.4.4 Machine learning in sentiment analysis

A widely discussed and oft-examined topic is that of sentiment analysis. Automated sentiment analysis has great potential in various fields, such as film reviews, newspaper editorials and stock analysis [19], and indeed any case that boils down to analysis of massive amounts of user input [62]. Rather than the extraction of detail on the emotional spectrum, sentiment analysis deals with the overall opinion towards an explored topic, i.e. if it is either positive or negative. Since sentiment can be expressed in a much more subtle way, it is expected to be more difficult to classify than a typical topic categorization [62]. Although automated sentiment analysis does not aim to extract detailed emotional information, it is still covered in this section as it utilizes very similar mechanisms.

In contrast to the analysis of typical datasets such as fairy tales and news headlines, Sidorov et al. investigated opinion mining in tweets, where opinion mining is the discipline of computational sentiment orientation of a short text [78]. As such, a Twitter message can be broken down to the object of the opinion, the feature of the object discussed, the opinion's sentiment polarity, the author of the tweet and, finally, the posting time. In order to apply supervised machine learning algorithms - in this particular case Naive Bayes, J48 and SVM - an extracted corpus of Twitter messages dealing with cell phone brands was created and manually annotated with the four classes positive, negative, neutral and informative. In terms of preprocessing four methods were used to normalize the investigated tweets. First, orthographic errors were eliminated with the help of dictionaries and statistical models. Secondly, special tags were introduced as a replacement for usernames, hashtags, etc. Further, the methods of lemmatizing and Part-Of-Speech (POS) tagging were utilized to decrease the number of word forms and categorize the words in the tweets (into nouns, verbs, articles and so on). Finally, negations were transformed into special prefixes of corresponding words. Practical experiments then revealed a proper setup for opinion mining: In terms of precision, the SVM classifier performed best given a unigram feature size. Regarding classifier training, precision increased with the size of the training set, although it stagnated at approximately 3000 tweets. Additionally, high performance could be maintained when classifying only into positive and negative sentiment within one and the same domain. Major causes of incorrect classifications were identified as shortened messages, misspelling, various kinds of humor, and human tagging errors [78].
Another way to classify Twitter sentiment classification was examined by Go et al. Instead of manual class annotation of Tweets for training purposes, they were categorized via emoticons, i.e. by a distant supervision approach. The emoticons, however, were stripped out from training data due to the discovery that accuracy was negatively impacted. Instead, unigrams, bigrams and POS related representations were applied as features. Generally, the authors identified the high frequency of misspellings in Tweets as problematic, since they are entered via various devices with different input methods. In contrast to Sidorov et al., the domain of investigated Tweets was not restricted to a specific topic. Training data was comprised of 1.6 million unique (i.e. no retweeted or repeated) Tweets, in which positive and negative Tweets were equally distributed.

The learning methods selected for sentiment determination were Naive Bayes, Maximum Entropy (MaxEnt) and SVM. The resulting models were subsequently applied to 177 negative and 187 positive Tweets in the test set. Best results in terms of accuracy could be achieved with the MaxEnt classifier when using bigrams in combination with unigrams (83%). While using only unigrams as features also resulted in good performance, bigrams delivered lower accuracy, as the feature space is very sparse, which turned out to be problematic especially for MaxEnt and SVM classifiers [30].
Sentiment analysis in the scope of particular subject was explored by Pang et al. by using movie review data as input datasets. Similar to the work of Go et al., no data hand labeling was required as the star rating in addition to each review entry indicated the sentiment needed for training and evaluation. Before employing learning algorithms, baseline values were obtained by simple word counts of positive and negative words in the reviews, which resulted in 69% accuracy. Preliminary to the learning method experiments, 700 positive and negative reviews were extracted and divided into three folds as preparation for cross validation. The experiments were conducted utilizing Naive Bayes, MaxEnt and SVM, resulting in similar findings to those of Go et al. Unigrams perform better than bigrams, especially when the presence of unigrams were taken into account rather than their frequency. The authors thus concluded that bigrams are barely sufficient in order to capture context of words in a corpus. The highest accuracy for Naive Bayes (81.50%) was achieved with a combination of unigrams and POS, while the performance of MaxEnt came close to Naive Bayes' accuracy by utilizing top 2633 unigrams. With slightly more than 82% accuracy, SVM provided the best results by using unigram features, or unigram plus bigram features [62].

CHAPTER 3

Method

The purpose of this chapter is to illustrate the chosen methodology with respect to the objectives of this work. As such, the selection of algorithms and tools required and the procedures applied are included in this part of the thesis.

3.1 Preliminary data analysis

Before the evaluation and selection of machine learning algorithms for emotion analysis, the analysis of the given dataset provides information that may be essential for that selection and the further configuration of a particular learning approach. As such, the preliminary data investigation consists of three steps:
Firstly, the way in which data was generated and obtained is examined. As the dataset used for this work does not contain real-world data, a detailed description of the experiment setup including test subjects, case and utilized tools is obligatory in order to identify possible biases and limitations regarding computed results and is necessary for the subsequent exploration. The second step of preliminary data analysis deals with the manually guided detection of emotions in messages which were generated in the corresponding experiment. Further, the application of MDS in the particular scenario is explained with respect to the concepts of Russell's circumplex of affect [67]. Finally, the values for valence and activation as results of MDS conducted by Hippmann [38] are subject to standard descriptive statistics. In particular, mean, deviation, minimum, maximum and quantiles are obtained for the provided numerical values of valence as well as of activation. Furthermore, the statistical measures are furthermore obtained separately for messages of negotiations concluding in an agreement and for those without agreement. The execution of a series of t-tests is done for the sake of determining central tendencies of valence and activation values.
The findings of this three-step approach are then considered in section 3.2. In detail, the results of MDS are used in order to form proper class labels, as discussed in subsection 3.2.6. The selection of promising learning methods fundamentally relies on proven concepts of text pro-

cessing and text mining and is also part of the subsequent section. Corresponding tools and their utilization for (pre)processing data and machine learning are discussed in sections 3.3 and 3.4.

3.2 Algorithm evaluation & selection

A key component needed to properly answer the research question of this work is the accurate selection of machine learning algorithms with respect to the characteristics of negotiation data. As such, this section contains the evaluation of methods that show promise in this regard. The approach is twofold: Starting with a deductive part, in which reasonable methodologies are derived from existing literature, the subsequent tasks follow an inductive approach. These procedures combined enable an optimal selection and optimization of applied data mining mechanisms. Note that interdependencies exist between this section and section 3.3.

3.2.1 Characteristics of text mining

When trying to extract emotions from texts by machine learning methods, it comes down to the discipline of text mining. The origins of text mining can be found in data mining [88], which has gained in popularity due to advances in software and hardware technology and the surge in available data [4]. Both mining categories have the ultimate goal of finding patterns in large volumes of data. Although text and data mining algorithms are based on the same concepts, the major difference according to Weiss et al. is that data mining relies on highly structured, relational data, while text mining typically relates to unstructured text documents. At first glance, those characteristics require different approaches. However, by transforming text blocks (e.g., single words, phrases, punctuation, etc.) into document attributes, the resulting spreadsheet representation can feed typical data mining algorithms [88]. Table 3.1 illustrates a very simple form of one such spreadsheet table, which holds documents as rows and the words as attributes of those documents in the columns. The actual data cells contain either the frequency of occurrences of attributes in a certain document, or the cells contain the binary values of “1” if the attribute occurs in a document, or “0” if not. Two aspects can be derived from this method of representation: firstly, that there are no missing values and secondly, data is sparse and high dimensional [88] [4]. Aggarwal et al. therefore suggest techniques for reducing dimensionality of data. Furthermore, semantically enriching the text data would help to improve the ability to find particular patterns. However, such methodologies are not yet sophisticated enough to represent texts of unrestricted domains in an accurate semantic manner [4]. Word-based approaches (such as BOW) are still the most promising form of text representation according to Aggarwal et al.
The purpose of text mining in the current case is focused on prediction on the basis of classification and optionally clustering, but regardless of the specific implementation of a mechanism, the evaluation of outcomes is crucial in order to measure the margin of error. As such, recall and precision are key concepts for determining accuracy in the context of document analysis [88].

3.2.2 Text analysis framework

As the previous subsection has shown, several considerations need to be taken into account when text mining in order to produce accurate results. The framework developed by Aggarwal et al.

Document content	Company	Income	Job	Overseas
“Income Overseas”	0	1	0	1
“Company Job Overseas”	1	0	1	1
“Compnay Income Job”	1	1	1	0
“Overseas”	0	0	0	1

Table 3.1: Example spreadsheet of words in documents, where each row represents a document instance [88]

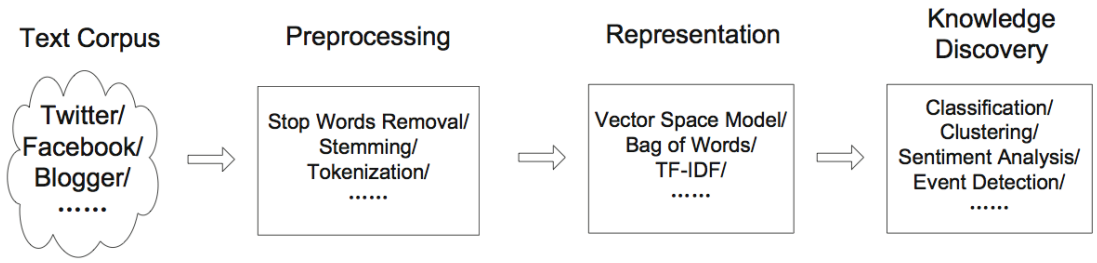

Figure 3.1: Four steps of a traditional text analysis framework [4]

structures text analysis into four steps (Figure 3.1), each of which must be addressed and certain decisions taken to be made in order to come to a satisfactory result [4]. The first step in the outlined framework deals with the text corpus that is to be analyzed. In the current case, this step covers the procedure generating negotiation data, e.g., case setup and conducting the experiment. In the next phase, input data is prepared such that subsequent tasks can handle text data smoothly, which leads to better results. Typical preprocessing methods are stopword removal, stemming and tokenization. Given a black list of words, stopword removal mechanisms eliminate those words from input text. Stemming, on the other hand, helps to decrease the number of distinct features by reducing variations of words to their root (e.g. “shout”, “shouted” and “shouting” are recognized as variations of “shout”). Another preprocessing task is tokenization, of which the objective is an accurate split of text streams into words, or rather tokens.

After the phase of text preparation, input data is modeled, in such a way that allows learning algorithms to compute them accordingly. A common approach in terms of text representation is BOW, which transforms documents to numeric vectors containing weights of tokens in the vector. In the final step of knowledge discovery, learning algorithms of different natures are applied to the model generated to represent the original text input.

The following subsections discuss potential methods in the context of the steps of preprocessing, representation and knowledge discovery with respect to the given negotiation dataset.

3.2.3 Text preprocessing

Datasets used in text mining are usually compiled from a highly diversified set of words since the input data is founded on natural language. As such, words like “a”, “the”, “and”, etc. occur many

times and are assumed to be meaningless for the reasoning of emotions. Leveling investigated the effect of different stopword lists applied to datasets prior to information retrieval and classification [48] as part of research conducted in the scope of "SMS-Based FAQ Retrieval". One part of the experiment was to classify out-of-domain queries with a k-Nearest Neighbor (kNN) classifier, which makes stopword elimination a relevant topic for this research as well, especially due to the fact that the accuracy of classification varied with the size of the stopword list applied. Leveling conducted experiments with 13 different stopword lists and in addition used lists with the 10, 20, 30, 40 and 50 most frequent words extracted from the corpus. As representative stopword lists for the experiments of this work, three sets are applied: an empty list in order to gauge the performance without stopword removal; secondly, Swish-E[1], which achieved the best accuracy in Leveling's classification experiments; and finally a list of the 50 most frequent words in the messages of the negotiation dataset, because the accuracy based on the extraction of the top 50 list was superior to that of the other four lists of top words by frequency. Table 3.2 summarizes the selected stopword lists.

Stopword list	**No. of stopwords**	**Accuracy[1]**
empty	0	75.5%
Swish-E	337	82.4%[2]
Top 50	50	81.4%

[1] accuracy according to out-of-domain queries [48]
[2] best accuracy across all stopword lists considered

Table 3.2: Selection of stopword lists applied in the series of experiments of the present work

Another task during the preprocessing phase is lexical analysis, which splits text into strings of characters, or tokens [52]. Tokenizing, however, is a procedure that has received little attention in terms of research regarding quality and adaptability [23], despite tokenization being a necessary step prior to any further operations on a given text [55]. The difficulty of tokenization boils down to the separation into chunks of characters in a way that allows the meaning of the resulting tokens to be perceived. For example, "newsstand", "news stand" and "news-stand" are frequently used forms, but when tokenizing is applied with certain delimiters, the words "news" and "stand" are separated and thus have a completely different meaning [55]. There are many more ambiguities in this regard [88]. Punctuation is one of the major challenges for tokenization as numbers can contain, for example, colons, periods and comma. Abbreviations are potentially problematic since they often contain periods (e.g., U.S.). Another sign causing trouble in the process of tokenization is the apostrophe, due to the fact that they can occur withing words (e.g., don't, can't), but also as an indicator of possession (e.g., Chris' car) and as boundaries of quotes. Similar problems occur with the dash as it can be used as part of a word (for instance newsstand), a sentence, a phone number, a calculation and so on.
Clearly, when tokenizing texts by delimiters, the influence of selected delimiters on the quality of tokenization cannot be ignored. For separation of tokens by simple boundary characters, Nugues suggests the usage of whitespaces, the dot (.), other boundary signs (,;:?!+$%-/\), brackets (()[]<>) and quotes (" ' " ') [55]. Mendez et al. conducted a series of experiments combining

[1] http://swish-e.org/, 04.02.2015

techniques of tokenizing, stemming and stopword removal in the context of spam filtering [52]. In terms of tokenizing, the authors applied spam filtering techniques including the machine learning algorithms Naive Bayes, Adaboost [71] and SVM to tokenized texts, which in one case separated tokens by whitespaces and punctuation marks, and in another case blanks only. Those two tokenization approaches were combined with stopword removal and stemming in order to find out the best configuration for automatized techniques in terms of spam filtering. Through the conducted experiments, it was possible to observe that tokenization using both blanks and punctuation together delivered the best results for machine learning algorithms with respect to correctly classified instances. Specifically, using those delimiters worked best when no stemming and no stopword removal was applied. However, when interpreting the results, one should consider that the corpus of spam emails is somewhat specialized as they are often enriched with noise (e.g., "M-O-N-E-Y!", "€lick HERE!!!"), which could have a negative impact on the effect of stemming and stopword removal.
A straightforward separation of tokens by defined signs seems to be simple, but more sophisticated tokenization approaches exist. For example, one approach is to define a set of tokenization rules, which treat a delimiter character only as such if certain characters accur to the left and right of it. This would support contiguous character sequences containing dashes, quotes, etc. being joined together into one token [55]. Even more sophisticated tokenization can be achieved by using machine learning. In such a scenario, each token would be classified to separate a token or not. In order to train a corresponding model, a text corpus with annotated token boundaries is necessary. A classifier then scans the annotated text and stores relevant information for each character, such as the previous character(s), the subsequent character(s), and whether the current character is a token delimiter or not. The classifier and the model created can then be used to tokenize any other unannotated text corpora [55]. Fares et al., for example, have shown that tokenization models adapted to particular domains can outperform state-of-the-art rule-based approaches in terms of accuracy [23]. In particular, they experimented with Conditional Random Field (CRF), a probabilistic learning approach [45], with the intention of learning two diverse tokenization schemes: Penn Treebank (PTB) [51] and English Resource Grammar (ERG) [26].
The delimiter set utilized for tokenization of the negotiation messages that are the subject of this work is based on the characters proposed by Nugues. Although Mendez et al. obtained adequate results by using only whitespace signs as delimiters, this approach seems to be especially successful in the domain of spam mail detection, which is known for abusive usage of corresponding characters. Further tokenization approaches are neglected for this thesis as these are likely to require corresponding pre-work (e.g., manual annotation, model training) and a separate series of experiments going beyond the scope of this work. However, the way extracted tokens are transformed into features - the quantification of text - is addressed in subsection 3.2.4.

When a text has been segmented into tokens, it can make sense to convert the resulting chunks (words) into a standardized form [88]. This process is called stemming and can be beneficial for later phases of the text analysis framework, such as information retrieval systems [18]. By using a stemmer, the number of attributes decreases as words are aggregated to common stems while the frequency of terms in documents increases, which is essential for algorithms based on frequency. However, there is no general rule that stemming improves performance of machine

learning, but rather its employment is application-dependent [88]. Stemming is furthermore distinguished by the aggressiveness of normalization of words. As such, inflectional stemming is a softer approach that deals with plurality and tenses. More aggressive forms of stemming return words to their root form, neglecting all prefixes and suffixes. The aim of such strict stemming is the drastic reduction of attributes and therefore more reliable distributional statistics, though the level of aggressiveness is application-dependent [88].
Taking that into account, experiments conducted in the context of this work need to reveal the effect of stemming. Different stemming algorithms were evaluated by Madariaga et al. [18], who investigated stemmers based on the frequency of errors they produce. Such errors are of two kinds: overstemming errors happen when words with different meaning are stemmed to the same root. Understemming errors occur in cases where words with the same semantic sense become two different stems. Depending on the weight - on the aggressiveness - of stemmers, one kind of error occurs more often than the other. Therefore, light stemmers produce few overstemming errors, while heavy stemmers are not prone to understemming errors [18]. By combining the two errors, Madariaga et al. derived an error rate to indicate the accuracy of stemmers. Based on experiments with three text corpora, they concluded that the Paice/Husk stemmer [61] performed better than the light weight stemmers of Porter [65] and Lovins [50]. Four variations of Hafner's stemmers [34] fluctuate in weight, but generally perform worse than the Paice/Husk, Porter and Lovins stemmers.
For the empirical part of this work, the influence of stemming on the performance of emotional pattern recognition is examined by application of the Porter stemmer as it is a very popular and successfully applied representative of lightweight stemmers used in a wide range of research studies [18] [10] [52] [24]. In addition to this a comparison between results with and without any stemming is made due to the fact that the performance of stemmers is rather application-dependent and may ultimately decrease machine learning performance. An investigation into the effects of further stemmers - such as the Paice/Husk stemmer reported to perform well by Madariga et al. - is not part of this work in order to to avoid over-expanding the scope.

3.2.4 Document representation

Perhaps the most crucial step prior to the application of a specific learning mechanism is the transition of the investigated text corpus into a processible representation. Existing methods in this regards cover a broad spectrum of complexity and, as such, affect the performance of text mining disciplines like Information Filtering (IF), Information Retrieval (IR), Information Extraction (IE), and document classification and clustering [77]. Furthermore, the selection of features, i.e. which characteristics describing the documents should be considered for extraction, is interdependently related to tasks in the preprocessing stage [88] [77].
The simplest form of quantitative text representation is the Bag-Of-Words (BOW) methodology, which was briefly mentioned in the introductory paragraph of this section. Table 3.1 illustrates the concept of BOW with rows representing documents, columns standing for features and each cell containing a number showing the relationship between the corresponding document and the feature. In reality, those numbers can represent: (a) the binary value, (b) the feature count, or (c) a weighted feature count [88]. The latter approach surpasses the binary and simple feature count variants in terms of performance [12]. A weighted feature count is commonly calculated

by Term Frequency/Inverse Document Frequency (TF-IDF), which combines a simple feature count with the occurrences of a certain feature across all documents. Formula 3.1 gives the corresponding weighting schema [4]. *tf(w)* is the simple feature frequency in a given document, while *df(w)* is the number of documents containing the feature, i.e. the document frequency, and *N* is the total number of documents considered.

$$tfidf(w) = tf \times log\frac{N}{df(w)} \tag{3.1}$$

The positive effect of weighting features in this way is that frequently used terms are less important (indicated by a low number, converging to zero if a term is used in almost every document) than those rarely used across documents. The higher the *tfidf(w)* value obtained for a term, the more important it is to the particular learning approach [88].
Representing a text in a Vector Space Model (VSM) as BOW, a feature does not necessarily have to be a simple term or word. Instead, the combination of words to n-grams can be highly predictive according to Weiss [88] and especially improves the performance of BOW [8]. This is the case if a combined group of single words reveal information that would be lost if the words were considered on their own. For example, "European Union" is a bigram conveying a meaning that is lost once the words are viewed separately. Additionally, n-grams are capable of capturing negations (e.g. "not happy"), representing an essential concept in emotion detection [14]. Introducing n-grams to the feature space drastically increases the number of features, which requires special treatment in terms of the feature selection described below. Although the task of building meaningful word combinations can be part of the preprocessing and representation phase, various learning methods are capable of combining terms themselves and suffer when given too much text preparation prior to model training [88].
Another technique for improving text mining results is based on the more detailed linguistic analysis of Part-Of-Speech (POS) tagging. The aim of this method is to categorize words and tokens into grammatical classes. Although the number of such classes is not exactly definable, linguists agree that the English language consists of at least nouns, verbs, adjectives, adverbs, prepositions and conjunctions [88]. The usage of POS can help to reduce ambiguity of tokens and therefore improve the quality of learning method results. In the context of opinion mining, nouns (e.g. crap, scandal), verbs (e.g. hate, hurt) and especially adjectives (e.g. amazing, anxious) are promising candidates for features as they are likely to indicate opinions [4]. Karimpour et al. examined the impact of POS tagging on Information Retrieval (IR) with Persian text corpora [40]. As such, the application of POS tagging was found to be beneficial in terms of precision of information retrieval, especially in combination with stemming. As part of the experiments conducted by Karimpour et al., tag-specific weighting schemata were applied, since in some IR applications, nouns were identified to be the most important tokens. However, results illustrated that the upscaling of nouns decreased the performance of the IR system under consideration [40].
A major drawback of the BOW representation is the loss of potentially essential information concerning the order of words. To overcome this issue, Bloehdorn and Hotho propose the integration of techniques on a conceptual level on top of lexical analysis [10]. As such, semantic enrichment of text addresses issues of synonymous words, polysemous words and generaliza-

tion of terms (e.g. car and lorry are vehicles), which are known difficulties of text representation. The key to this is the utilization of ontologies, such as the WORDNET database, which aims for a conceptual feature representation. In particular, before employing a certain learner, concepts were extracted from explored texts through a five-step process. This procedure involved multi-word expression detection, POS analysis, stemming, and two steps involving ontologies: Word Sense Disambiguation (WSD) and generalization of terms and phrases. Experiments with three different text corpora in the domains of news articles, agriculture and medical reports revealed a statistically significant improvement in classification performance. As such, the combination of terms and semantic concepts in the feature space was proved to deliver better results than classification based on simple term representation [10]. Similar findings were observed by Shaban et al., who examined document mining with a focus on the semantic understanding of the contents. By exploiting semantic information for similarity calculation between documents, Shaban et al. were able to improve clustering results in comparison with a traditional BOW/VSM approach [77]. Aggarwal, however, states that semantic approaches towards cross-domain text representation are not yet in a state to deliver robust and accurate results [4]. Thus, BOW is still widely used in the context of text mining, but representations involving semantics are becoming more important in particular domains, for example, biomedicine and semantic web.

One characteristic of a BOW approach is the huge number of words extracted from the analyzed documents [12]. Although feature reduction mechanisms such as stopword removal and stemming decrease the number of tokens in the feature space, further feature selection is required. Regarding feature frequency, most frequent words are not likely to reveal information concerning class membership and therefore tend to be stopwords. Rarely used terms, on the other hand, are typically typos and as such can be safely ignored [88]. However, even if this approach reduces the number of features to some extent, the vast majority of features will remain in the feature space, regardless of their influence on the subsequent learning process. In order to select the most promising features concerning class prediction, certain feature selection methods were established. Omar et al. compared seven feature selection approaches in combination with three classifiers in the context of sentiment analysis [59]. Of these, the commonly used method of Information Gain (IG) [4] [12] [88] turned out to be stable and performed well across various classifiers and feature space sizes. Information gain is a measure for the contribution of a feature towards classification, calculated as illustrated in Equation 3.2. Here, P_i represents the general probability of class *i*, while $p_i(w)$ is the probability, that a certain document containing the word/feature *w*, belongs to class *i*. *F(w)* expresses the fraction of documents that include the word *w*, and *k* is the total number of classes [4].

$$\begin{aligned} IG(w) = -\sum_{i=1}^{k} P_i \times log(P_i) + F(w) \times \sum_{i=1}^{k} p_i(w) \times log(p_i(w)) \\ +(1 - F(w)) \times \sum_{i=1}^{k} (1 - p_i(w)) \times log(1 - p_i(w)) \end{aligned} \tag{3.2}$$

The higher the value of *IG(w)* in Formula 3.2, the more discriminatory power of the feature *w*. The application of IG indicates features well suited for subsequent learning mechanisms,

although, a suitable number of features varies and, therefore, needs to be discovered through a the series of experiments.
Taking the theoretical aspects mentioned above into account, the approach to emotional pattern recognition in this work is to apply BOW combined with TF-IDF feature weighting. The features considered are unigrams and, in order to obtain the potential of negations, bigrams. As the number of features is expected to be tremendously high due to the additional usage of bigrams, selection of features based on the Information Gain (IG) methodology is employed, and the number of features is empirically optimized. Additionally, a text representation composed only of features tagged as nouns, verbs and adjectives is chosen to reveal potential benefits of the implementation of POS tagging in the context of text based negotiation. Semantic involvement regarding text representation is outside the of scope of this work, because semantic databases focusing specifically on the domain of negotiation or communication do not exist, and the cross-domain application of semantic models does not seem too promising.

3.2.5 Knowledge discovery - algorithm selection

At the time the fundamental data is prepared and represented as described in the preceding sections, actual machine learning methods are applied in order to reveal information previously hidden to the human eye. In order to do so, several mining disciplines have been developed, each of them applicable depending on which objective is to be achieved, as well as on the characteristics of the dataset under investigation. Gupta points out three superior data mining families fundamental to the majority of mining approaches in the field [33]. While association rule mining aims for extraction of easily understandable rules on data of variable length, and cluster analysis is meant to find meaningful groups of instances in unknown (i.e. unlabeled) data, supervised classification is the technique appropriate for the experiments of this work. Supervised classification relies on the existence of training data, for which each instance is labeled with the correct class. Although the dataset used for the experiments in the scope of this work does not contain a class column, the corresponding label of each message in the dataset will be derived as described in subsection 3.2.6.
Classification of documents in the context of text mining can be achieved by classifiers of various nature. Those can be distinguished by the basic concepts of model building and consequently by class determination. As such, this paragraph briefly describes and selects algorithms that are to some extent proven in text mining and are, therefore, promising with respect to the classification of documents (in this case, negotiation messages) by emotional states.

3.2.5.1 Probability based classifiers

Classifiers of this classification family rely on the simple comparison of probabilities of word occurrences in documents [4]. Thus, the basic concept consists of the goal to find the highest $P(C_i|x)$ for a certain document instance, namely the most probable class C_i for a given vector x, which indicates presence and absence of words. In order to estimate that probability efficiently, Bayes' theorem [20] is employed accordingly, illustrated in Equation 3.3. $P(x|C)$ represents the probability of x when the class C_i is certain, $P(C_i)$ is the overall probability of a document being in class C_i, while $P(x)$ represents the probability of occurrence of the feature values x,

regardless of the class.

$$P(C_i|x) = \frac{P(x|C_i) \times P(C_i)}{P(x)} \tag{3.3}$$

Since *P(C_i)* is easy to estimate (the actual distribution of classes in the training set) and the estimator for *P(x)* can be neglected (independent from *P(C_i)* and irrelevant for the order of probabilities of any *P(C_i|x)*), the crucial term is *P(x|C_i)*. In order to get an estimation for this value, a naive approach is utilized, which assumes that all features describing a document are independent [33]. That assumption permits an estimation of *P(x|C_i)* by simply calculating the rates of each feature value in class C_i.
Regarding text mining, the Naive Bayes method works in combination with BOW representation and can be of two kinds. In the Multivariate Bernoulli Model, it is the presence and absence of features that represent a document, i.e. the features are binary. The Multinominal Model, on the other hand, is capable of handling word frequencies of tokens in the BOW representation.
As in the work undertaken here, BOW will be implemented using TF-IDF as feature representation, which means that the traditional Naive Bayes algorithm using the Multivariate Bernoulli Model won't perform. Generally, research has shown that Naive Bayes in text mining delivers moderate results at best [78] [81] [80]. Similar to decision tree classifiers, Naive Bayes is sensitive to n-gram size and number of classes, but seems to be rather stable with respect to the corpus size [78]. However, the experiments in the aforementioned literature were conducted on BOW representation using binary features, while this work examines WEKA's NaiveBayesMultinomial[2] algorithm in combination with the steps declared in this chapter.

3.2.5.2 Decision trees

Other than the probability based approach, decision tree classifiers utilize text features in order to break down text data into partitions hierarchically and, typically, at each node a condition based on a certain feature value is placed [4] [7]. The selection of attributes considered for the split at a particular node depends on aspects of information theory, an example of which is the concept of Information Gain (IG) [33]. In the context of text classification, the conditions deal with presence, absence or frequency of tokens/words in documents to be classified. The break down of the initial data is repeated recursively, naturally until a leaf holds a minimum number of objects. The induced tree model is then used to classify further document (test)instances. Thus, each test instance is applied top-down the model until a leaf node, which represents the actual class, is reached. For instance, a test sample with two numerical attributes *X* and *Y* is classified as circle or filled square when applied to the decision tree in Figure 3.2. Regarding reliability of a decision tree, the size of training data and the size of features need to be in balance [33]. Thus, as the feature space in text mining scenarios is potentially massive, decision tree approaches require huge training datasets. As a result, the complexity of decision trees correlates with the number of features considered, although the set of features accurately determining a class is relatively small. However, large feature spaces potentially result in overfitted trees, which are tailored to the training set, but perform poorly on unseen data. Consequently, models are reduced by the technique of pruning: reducing error rates of sub-trees by neglecting leaf nodes

[2] http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayesMultinomial.html, 07.03.2015

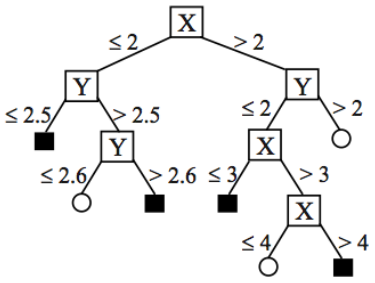

Figure 3.2: Simple decision tree based on two numerical features (X and Y) and two classes [49]

and substituting whole branches with leaf nodes [33].
Decision tree implementations such as J48[3] (which is the Java implementation of the C4.5 algorithm [66]) were already used in text mining for experiments into emotion/sentiment detection [14] [78]. The range of performance of the decision tree approach in those studies varies highly, especially depending on the number of classes, n-gram size and corpus size. Compared to the work undertaken here, these studies differ particularly in terms of document length, which are either (size-limited) Tweets, headlines or extracted sentences. However, other than that, neither general rules for the preprocessing phase, nor optimized settings for the classifier could be obtained. Consequently, the best combination of processing steps and optimized settings of the J48 classifier - especially regarding pruning - are explored during the subsequent experiments of this work.

3.2.5.3 SVM classifiers

The typical algorithm of choice when it comes to classification of high-dimensional data is of the family of Support Vector Machines (SVMs) as they outperform other learning methods in terms of accuracy, which was especially true for text classification [49]. This holds mainly due to the fact that SVM is capable of handling highly sparse data - as is the case in BOW representation - and because SVM is rather insensitive of accurate feature selection [88]. In the basic form, SVM can handle two-class classification by building a linear hyperplane between instances of the two classes, illustrated in Figure 3.3. The graphic furthermore answers the obvious question of which of the infinite possibilities to draw a boundary should be chosen, namely the one maximizing the distance between data samples of the distinct classes. In the likely case of nonlinear decision boundaries, kernel functions (e.g. the polynomial kernel [90]) compute dot products in order to map original data to a high dimensional space [49].
However, since SVM is basically designed for two-class problems, according enhancements are required, which are also necessary in the case handled here as the experimental data includes more than one class (subsection 3.2.6). A prominent approach in this regard is the one-versus-all or winner-takes-all method [88]. The method relies on the concept of computing and comparing the scoring of test documents for each possible class. For example, when given classes A, B and C, the classifier calculates the scoring of test document d in a pair-wise comparison: A versus B,

[3] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html, 07.03.2015

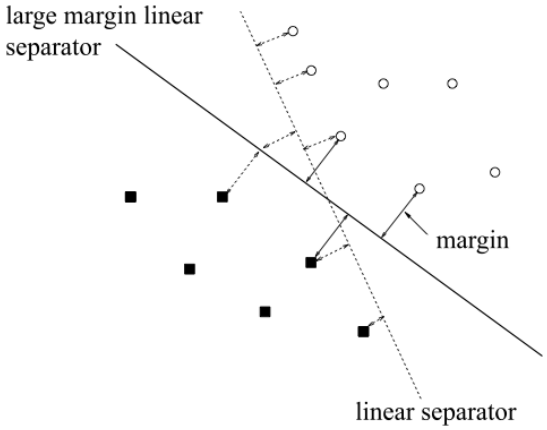

Figure 3.3: Linear hyperplane separating class instances [88]

A versus *C*, and *B* versus *C*. Consequently, *d* is assigned to the class that wins the corresponding matches.
As SVM is predestined for text classification, this approach is a safe choice for the series of experiments in this work. Moreover, SVMs were practically applied in many research studies in the context of emotional classification and sentiment analysis with partially persuasive results [30] [62] [78]. Considering the results of the referenced research, the tendency of SVM to be the superior learning method when using unigram representation, can be observed. Sidorov et al. furthermore proved the negative influence of small corpus size and increasing class amount to SVM performance. An implementation of SVM capable of dealing with multi-class datasets is SMO, which is available in WEKA[4] and part of the experiments conducted in the context of the present work.

3.2.5.4 Proximity based classifiers

Other than the learning methods in the previous paragraphs, proximity based classifiers - in particular k-Nearest Neighbor (kNN) - are lazy learning methods, meaning that they do not train and rely on a model. As such, the methods mentioned above are eager learning methods [49]. kNN basically consists of a three-step algorithm, starting with a calculation of the distances between a test sample and all document instances in the training dataset. Subsequently, the k documents from the training set with the least distance to the particular test sample are selected. Finally, the test sample is classified to the most frequent class across the selected training documents. For example, the test sample in Figure 3.4 (dot with cross) is classified as filled square as two out of the three considered neighbors are filled squares. There is no common recommendation for the actual value of k, it rather needs to be explored during experimental procedures by comparing performance indicators produced with different values for k [88] [49]. However, low values such as $k = 1$ are naturally a bad choice as a single nearest neighbor could be an outlier in the training set. However, it is common to weight distances in such a way that close

[4] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html, 07.03.2015

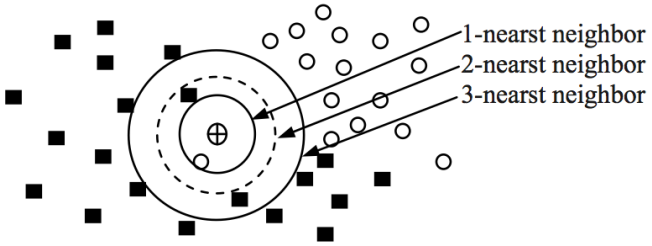

Figure 3.4: kNN classification with $k = 3$ [49]

neighbors are worth more than neighbors far away in terms of classification. Besides the proper choice of k, the performance of kNN highly depends on the distance function used for calculating distances between a test sample and training samples. As such, a BOW representation using TF-IDF measurement typically provides appropriate results when applying cosine similarity [48] [88]. Cosine similarity is the preferred choice when it comes to distance calculation for text documents, while for data mining on relational data Euclidean distance is superior [49]. Obviously, kNN trades training effort for computation time during classification as on the one hand there is no training required, but, on the other hand, all training samples need to be processed for each test sample. A possible solution in the case of unacceptable computation time is data aggregation in the pre-processing phase, which summarizes similar documents to clusters and furthermore to meta-documents. The set of meta-documents generated then represents the new training set for kNN classification [4].
In the text mining domain, Leveling investigated the effect of stopword removal in the context of IR by utilizing a kNN classifier [48]. Han et al. dealt with text categorization using various algorithms including kNN, which delivered an accuracy of more than 90% on a particular training set in combination with sophisticated feature selection [35], which emphasizes the potential of this method for classifying emotions in the present case. WEKA's implementation of kNN is called Instance-Based k Learner (IBk)[5] and is used for experiments in this work. As suggested in the literature, a proper value for k is determined in an exploratory way, and the same goes for the application of distance weights. Unfortunately, WEKA does not provide cosine similarity measurement out of the box, which makes Euclidean distance the next best choice for distance calculation.

3.2.6 Building classes

Supervised classification relies on a set of training data, usually including hand-crafted labels for each instance in the dataset. In the present case, each document in the dataset represents a negotiation message (Chapter 4). The emotional annotation of each message is achieved by the method of Multidimensional Scaling (MDS), which is described in detail in section 5.1. As the results of the examination of valence and affect with respect to Russell's circumplex model of affect [67] consist of continuous numbers between -1 and 1 for each dimension, a meaningful

[5] http://weka.sourceforge.net/doc.dev/weka/classifiers/lazy/IBk.html, 07.03.2015

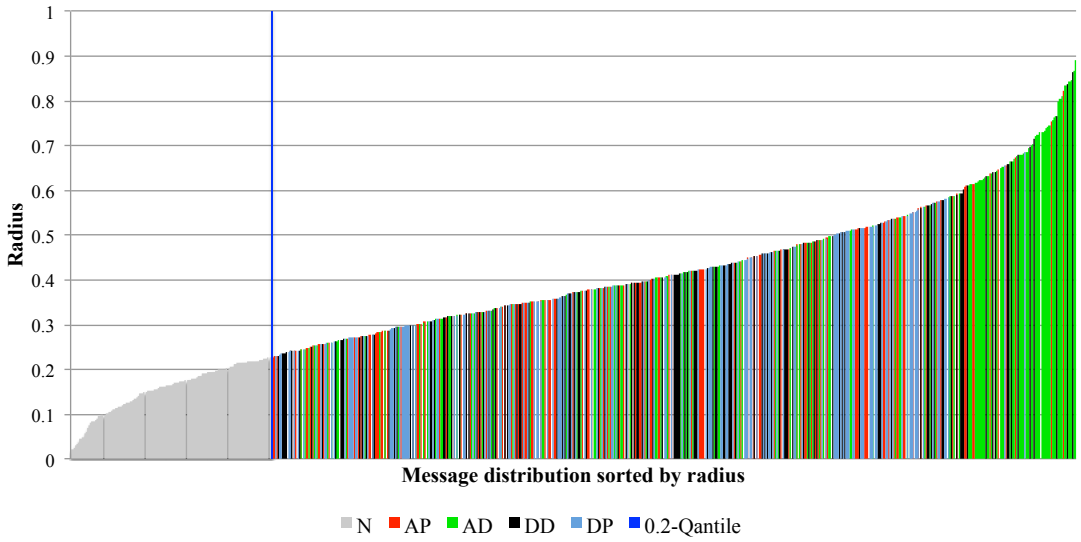

Figure 3.5: Class distribution from a radius perspective in the two-dimensional affective space

encoding into class labels is required.
An essential characteristic of Russell's model is the space of neutral affect and valence. The origin of the bipolar two-dimensional space indicates "absolute" neutrality, whereas an increasing radius in a particular angle points to certain affect concepts, for instance excitement, calm and sadness [67]. Thus, the first task is to identify a radius separating neutral messages from emotionally loaded messages. The chosen approach to this is such that, initially, the radius for each messag is calculated by applying the equation of a circle. Next, the crucial decision concerning the neutral-space radius needs to be made. Considering the rather small amount of documents in the training set (730 messages, compared with, for example, 4090 instances in [28], 2053 in [62] and 1.6 million in [30]), a balanced choice is required in order that neutral messages are neither over-represented nor under-represented, and to simultaneously keep enough messages for other classes indicating various emotional states. Therefore, a radius splitting the dataset at the 0.2-quantile was established. For the present dataset this value is 0.227, shown in Figure 3.5.

The additional consideration regarding class definition deals with the question of how - and to what granularity - emotionally loaded messages should be labeled. As such, a possible approach is to define ranges of angles in Russell's space and assign each range a particular emotion label. Then, the concepts of basic emotions by Ekman [22] or Izard [39] could be matched and embedded into the circumplex model. However, a classification into six (Ekman) and ten (Izard) emotions seems to be too granular considering the narrow training dataset and the number of remaining instances after subtracting neutral messages. Consequently, a bipolar, two-dimensional space suggests the separation into quadrants, i.e. four angle ranges of 90 degrees. According to Hippmann, each quadrant of the affective space maps to a particular group of emotions [38]. The first quadrant (activated pleasure) maps to emotions such as excitement and enthusiasm. Quadrant two (activated displeasure) covers affective expressions like anger and annoyance. In the

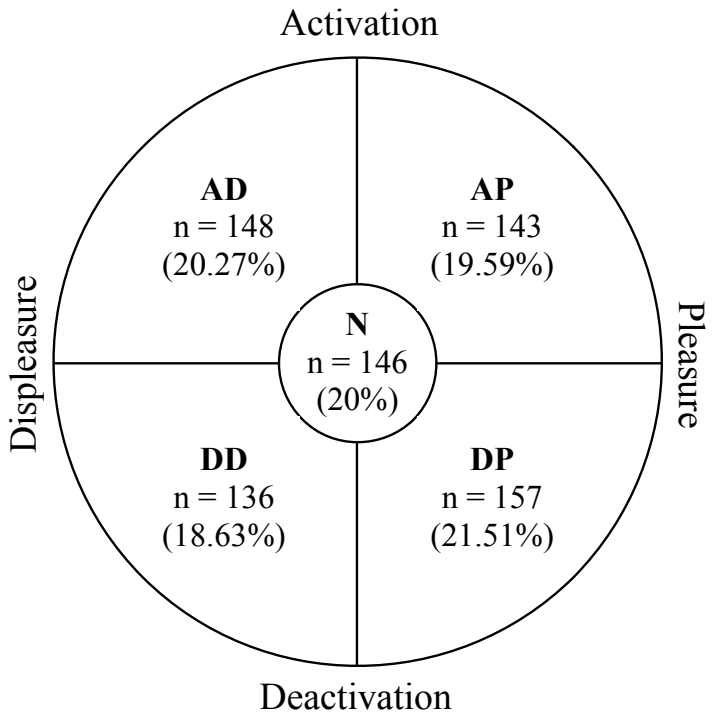

Figure 3.6: Class encoding of messages with respect to valence and activation

third quadrant (deactivated displeasure) states such as indifference and dullness can be found, and in the fourth quadrant (deactivated pleasure) serenity and relaxation are located. Those mappings are especially robust in the middle of each angle range (45, 135, 225 and 315 degrees) and become blurred towards the edges of the range. In fact, the edges of four quadrants represent the axes of the circumplex model, which themselves cover a certain range of emotions. However, classification into a neutral zone and the four quadrants seems to be a fair compromise in terms of granularity and reasonable encoding of valence and affect, particularly when considering the rather small number of training instances. Figure 3.6 illustrates the structure of class label assignment to messages of the dataset used in this work.

3.2.7 Validation of machine learning results

Last but not least, results delivered by machine learning methods need to be evaluated in order to determine the performance of the applied solutions and to estimate their future performance [88]. Therefore, several estimators for the accuracy of a classifier exist. The simplest one is the rate of correctly classified documents, shown in Equation 3.4. T stands for the total amount of documents, while C represents correctly classified instances. The inverse performance indicator - marked in Equation 3.5 - is the error rate. These two estimates are reliable if the test dataset

(T) is large and representative [33].

$$Accuracy = \frac{C}{T} \tag{3.4}$$

$$Error\ rate = \frac{T - C}{T} \tag{3.5}$$

The overall performance of the classifier can be estimated by looking at accuracy and/or error rate, although these reveal little about the characteristics of the classification errors obtained. Consequently, additional ratios are used for performance analysis. The precision of a particular class (Equation 3.6) describes the portion of correctly classified instances in comparison to all instances assigned to that class.

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

On the ohter hand, the recall of a certain class - as shown in Equation 3.7 - explains the ratio between correctly classified instances and all instances of that particular class.

$$Recall = \frac{TP}{TP + FN} \tag{3.7}$$

The F-score illustrated in Equation 3.8 is the harmonic mean of the measures of precision and recall mentioned above, and is typically used to gauge the performance of a classifier in a single number [88].

$$F\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{3.8}$$

It is the case that recall and precision depend on each other reciprocally, which results in the precision-recall tradeoff. Thus, while the overall error rate stays the same, the adjustment of threshold settings positively influences the precision, but negatively impacts the recall, or vice versa [88].

Retrieving these performance indicators relies on the concept of separate datasets for learning and for performance measuring, called the training and test dataset [88]. The holdout method typically requires mutually exclusive training and test sets. In the case of having only one dataset, it is split in the ratio 80:20 (training:test set) for example. The split needs to be considered carefully as a larger training set is beneficial for the classifier, while a larger test set ensures a more reliable estimation of accuracy [33]. The holdout method avoids estimator bias due to the fact that test data is not used in the training phase. However, proper estimates require huge and representative test and training sets. Another proven method to measure accuracy is k-fold cross-validation, which utilizes the whole dataset for training and testing. Thus, the dataset is separated into k (preferably equally sized) subsets, of which one is used for testing and the rest for training. Repeated k times, the mean of accuracy estimators for each session is delivered as the final estimator. K-fold cross-validation is widely used for estimating accuracy as it provides reliable results, where 10 has been found to be an appropriate value for k [33].

As for the this thesis, accuracy, precision, recall and F-score- which is based on precision and

recall anyways - are used in order to compare and evaluate performance of the experiment outcomes, as these measures are standard values when evaluating classification outcomes [6] [81] [5]. The method of choice for obtaining these measures is k-fold cross-validation with 10 folds, as on the one hand a meaningful split of the dataset in terms of representativeness cannot be ensured, and on the other hand the small size of the given dataset would limit the test and training dataset further.

3.2.8 Summary

To summarize, the combination of activities in the preprocessing, presentation and knowledge discovery phases result in a matrix of experiment settings. In particular, six experiment parameters were determined, of which four of them are represented in Table 3.3. The negotiation messages are used either in the original form or in the adjusted form with nouns, verbs and adjectives only. In the case of the original dataset, all selected stopword lists can be applied, while for POS tagged messages, only the 50 most frequent words, or no words at all make sense. Furthermore, stemming is inapplicable for the adjusted dataset as stemmers cannot handle the tags attached to words and this would result in potentially classification-relevant information added by the POS tagger being lost.

Setting	**Dataset**	**n-grams**	**Stopwords**	**Stemmer**
#01	original	unigram	none	none
#02	original	unigram	none	Porter
#03	original	unigram	Swish-E	none
#04	original	unigram	Swish-E	Porter
#05	original	unigram	Top 50	none
#06	original	unigram	Top 50	Porter
#07	original	uni- & bigram	none	none
#08	original	uni- & bigram	none	Porter
#09	original	uni- & bigram	Swish-E	none
#10	original	uni- & bigram	Swish-E	Porter
#11	original	uni- & bigram	Top 50	none
#12	original	uni- & bigram	Top 50	Porter
#13	POS adjusted	unigram	none	none
#14	POS adjusted	unigram	Top 50	none
#15	POS adjusted	uni- & bigram	none	none
#16	POS adjusted	uni- & bigram	Top 50	none

Table 3.3: List of combined experiment settings applied to four selected learning mechanisms

Besides the options shown in Table 3.3, optimizing feature selection is part of the empirical part of this work. The remaining experiment parameter concerns the chosen machine learning approaches. As such, settings are combined with each of the four selected learning methods, namely J48, NaiveBayesMultinomial, SMO and IBk. Results of corresponding settings are discussed in Chapter 5, where individual experiment instances are referenced with the setting

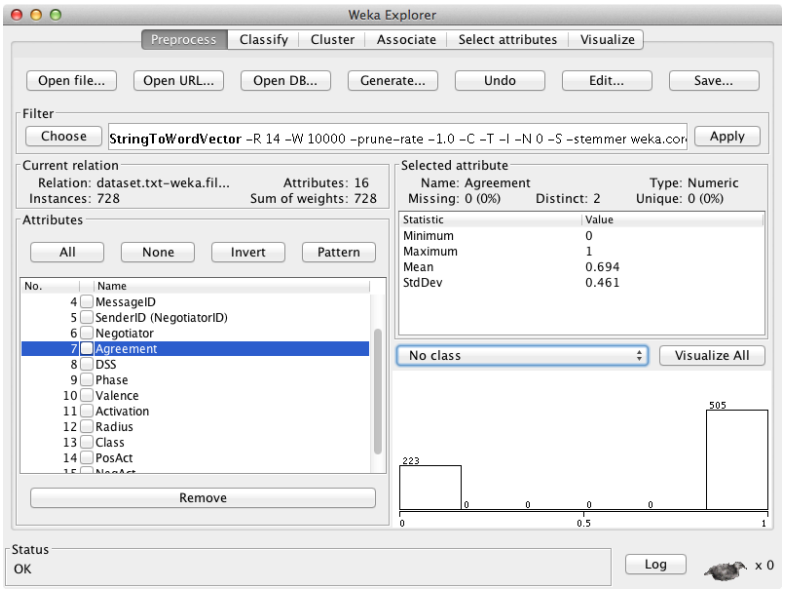

Figure 3.7: The preprocess section of WEKA including filter configuration and attributes list

number followed by the abbreviation of the learning method, e.g. #01.J48, #05.NBM, #09.SMO and #14.IBk.

3.3 Tool selection

Regarding tool selection for data mining related tasks, the author relies on the comparison study of Wahbeh et al. as a detailed analysis of various alternatives is not the focus of this work. Wahbeh et al. compared tools with respect to their performance regarding classification by measuring accuracy. The tools under investigation were WEKA[6], TANAGRA[7], KNIME[8] and Orange[9]; the evaluation was performed by applying six classification algorithms including Naive Bayes, C4.5 (decision tree) and SVM to nine datasets varying in characteristics such as attribute types (e.g. categorical, integer), number of instances, and number of attributes. Given the accuracies of their conducted experiments, Wahbeh et al. concluded that none of the examined tools is superior in terms of classification. However, WEKA was the only tool supporting all six algorithms tested on the selected datasets [2]. Those results and the fact that WEKA is capable of executing pre-processing, classification, clustering, regression, association rules and visualization makes it a safe choice for further investigation of the given dataset for this work.
In particular, WEKA is a bundle of tools and applications that are to some extent related to the field of data mining. Besides visualization and data viewer tools supporting data scientists in

[6] http://www.cs.waikato.ac.nz/ ml/weka/, 03.12.2014
[7] http://eric.univ-lyon2.fr/ ricco/tanagra/en/tanagra.html, 03.12.2014
[8] https://www.knime.org/, 03.12.2014
[9] http://orange.biolab.si/, 03.12.2014

investigating datasets in a traditional manner, WEKA provides four major components. Firstly, there is the Explorer component, which contains features required for the whole data mining process described in subsection 3.2.1. As such, a comprehensive preprocess section in the Explorer deals with analysis and preparation of source data for further mining tasks. Figure 3.7 illustrates the corresponding view including the sections with listed attributes and a separate filter section. The latter offers a huge range of filter possibilities, though the most essential ones for the present work are those for the employment of stopword removal, stemmers, tokenizers and attribute type converters. Each filter comes with individual settings, as does the "StringToWordVectorFilter", a filter bundling stopword application, stemming and tokenizing for text transformations into BOW representation, shown in Figure 3.8. Note that this filter is the central component for data preprocessing and data representation in the scope of this thesis. Once the preparation of the dataset is done according to subsections 3.2.3 and 3.2.4, the actual mining tasks, i.e. classification, are achieved in the "Classify" section of WEKA Explorer. Figure 3.9 points out four major areas: the classifier selection including the possibility to adjust settings; the test options, e.g., for result validation via k-fold cross-validation; the class selector determining the class column in the training set; and, finally, the classifier output consisting of classifier specific information and accuracy information.
Another useful utility of WEKA Explorer supports the process of attribute selection. Thus, methodologies including Information Gain (IG) analyze the value of attributes regarding the defined class attribute and can therefore advise the user to select the most valuable attributes during the process of feature selection/reduction. The attribute selection support of WEKA is employed in the experiments in the context of this work according to the methodological approach described earlier in this chapter.
The described feature set of WEKA's Explorer covers the majority of required tool support in order to conduct the empiric research of the present work. For the sake of completeness, the other three applications of WEKA will be mentioned briefly at this point. The Experimenter application helps to execute experiments in a kind of batch mode. In detail, it provides a three-step workflow, where the first step - illustrated in Figure 3.10 - consists of setting up input datasets, algorithms to be applied and the validation approach. The second phase basically involves the application of the chosen algorithms to the selected datasets, while in the last step the focus is on analysis of performance indicators across methods and datasets.
A slightly different approach compared with that of the two utilities discussed so far is implemented with the Knowledge Flow application. With this tool, users can model real work- and data-flows. Thus, in addition to typical activity nodes representing classifiers, cluster algorithms and filters, components like data sources, data sinks and visualization nodes are available for modeling.
In addition to the preceding graphical user-interface based tools, WEKA comes with a Command Line Interface (CLI) component (called "Simple CLI", which provides command line access to fundamental WEKA functionality as well. WEKA libraries can furthermore be linked and used in Java programs for customized workflows and integration with other applications.

As for the enhanced task of POS tagging, another tool is required due to the fact that WEKA does not include sophisticated Natural Language Processing (NLP) utilities. Therefore, the tool-

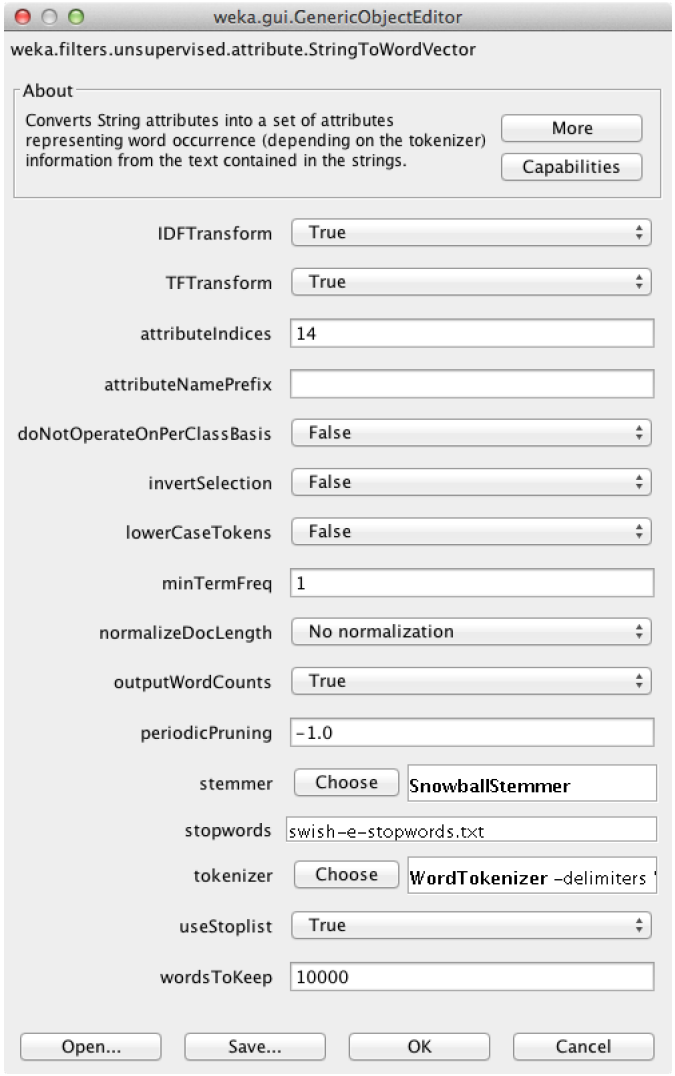

Figure 3.8: For each filter, WEKA provides individual settings, as demonstrated here for the TextToWordFilter, which handles tokenization, stemming and stopword removal

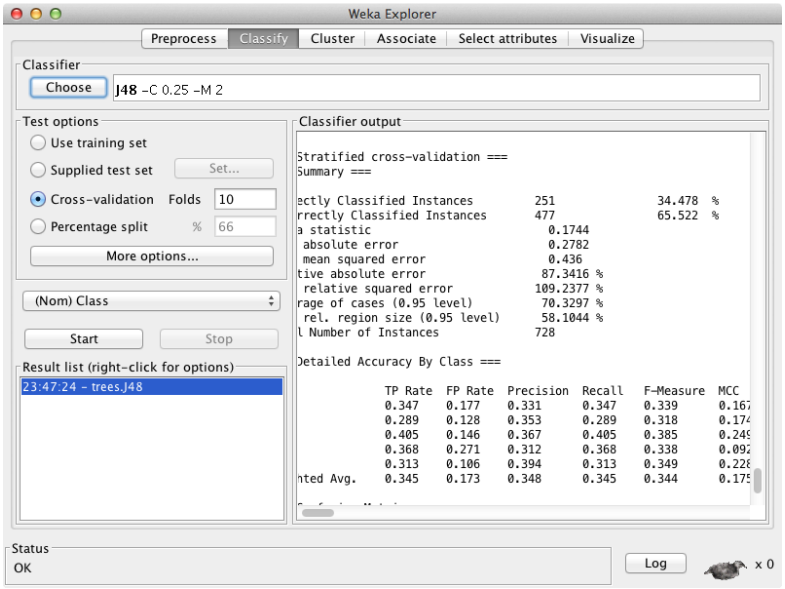

Figure 3.9: WEKA's Classify section provides the option to choose classifiers, evaluation method and column class

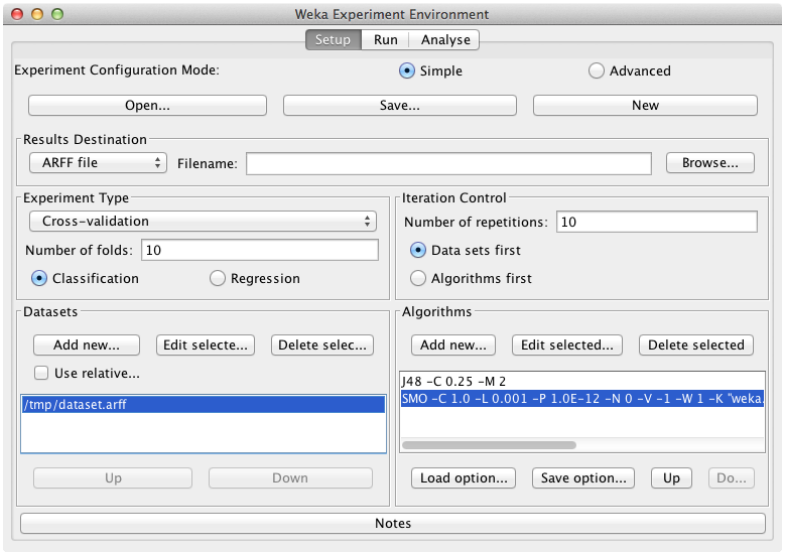

Figure 3.10: WEKA's Experimenter provides the option to conduct experiments in a kind of batch mode

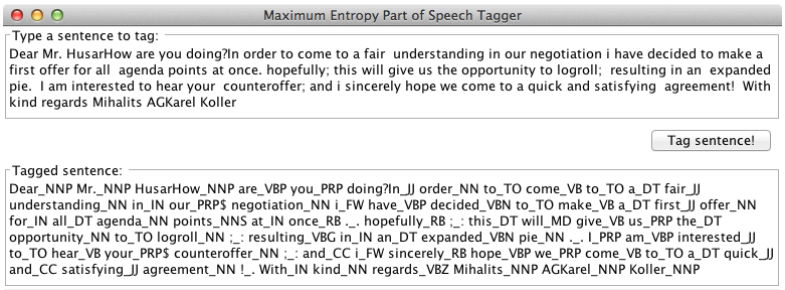

Figure 3.11: Stanford NLP Group's basic UI tool for POS tagging[13]; tags according to Santorini's POS tagging guidelines for the PTB project [70]

set of the Stanford NLP Group is used. The Stanford NLP Group is a conglomerate of experts of various disciplines such as linguistics and computer science[10]. In order to tackle problems such as Word Sense Disambiguation (WSD), sentence understanding and probabilistic tagging, they employ integrated approaches of data analysis, linguistic modeling, machine learning, and probabilistic methods. Furthermore, the Stanford NLP Group publicly provides their NLP software including the Stanford POS tagger for download[11].
Experiments conducted by Toutanoa et al., who utilized the aforementioned POS tagger, revealed 97.24% tag accuracy and an accuracy of 56.34% in terms of correctly tagging full sentences. Those accuracy values were achieved by employing methods like adjacent tag contexts, incorporation of multiple consecutive words and enhanced handling of unknown word features [83]. Although those investigations relied on the PTB Wall Street Journal dataset[12] and therefore on a different domain as in the present case, the tagger is assumed to be robust enough to detect nouns, verbs and adjectives in negotiation messages as well. Figure 3.11 shows a sample message tagged by Standford NLP Group's POS tagger, obviously assigned with all kind of tags explained in [70]. However, the practical approach to POS tagging in the context of this work is such that the tagger analyzes each negotiation message and subsequently all terms not tagged with a noun, verb or adjective tag are removed from the message, resulting in a separate input dataset for further processing. Since there are various tags for nouns, verbs and adjectives, Table 3.4 summarizes the tags, which are kept in the remaining message dataset.

3.4 Initial data preparation

After having chosen WEKA and Stanford's POS tagger as the most appropriate tools for further research, this section provides necessary steps in order to be able to use the given dataset with the mentioned utils.
Due to the fact that the dataset is provided as a spreadsheet, some transformation was required before being able to load it into WEKA. The format of choice therefore is Attribute-Relation

[10] http://nlp.stanford.edu/index.shtml, 04.03.2015
[11] http://nlp.stanford.edu/software/index.shtml, 04.03.2015
[12] http://www.cis.upenn.edu/ treebank/, 04.03.2015

Tag	Description	Examples
NN	noun, singular or mass	message, negotiation
NNS	noun, plural	messages, negotiations
NNP	proper noun, singular or mass	Austrian, Java
NNPS	proper noun, plural	Austrians, MacBooks
JJ	adjective	young, small
JJR	adjective, comparative	younger, smaller
JJS	adjective, superlative	youngest, smallest
VB	verb, base form	come, make
VBD	verb, past tense	came, made
VBG	verb, gerund or present participle	coming, making
VBN	verb, past participle	made, broken
VBP	verb, non-3^{rd} person singular present	come, make
VBZ	verb, 3^{rd} person singular present	comes, makes

Table 3.4: List of tags indicating which words in the negotiation messages are kept in the dataset for POS related experiments [70]

File Format (ARFF). An ARFF file consists of two sections, a header and a data section. The header provides information about the name of the given relation and lists the attributes and their data types available in the dataset. Below the header, the data section contains data in comma separated form[14]. Coming from a spreadsheet, the following steps were taken until data loading into WEKA was completed successfully. Firstly, the source datafile is stripped to the required minimum, i.e. only the column containing negotiation messages and the class labels will remain (detailed information can be found in Chapter 4 and subsection 3.2.6). Secondly, commas in negotiation messages were replaced by semi-colons in order to allow WEKA to distinguish between Comma-Separated Values (CSV) related commas and data related commas. In the same step, each message was surrounded by apostrophes, whereas apostrophes within messages were invalidated accordingly. Furthermore, it should be mentioned here that there were some occurrences of Custom Stylesheets (CSS) blocks in text messages that were removed before transformation as they obviously appeared there by mistake. Finally, a line break was added at the end of the resulting csv file, which was then successfully transformed to an ARFF file by the command shown in Listing 3.1.

```
java -cp weka.jar weka.core.converters.CSVLoader original.csv >
    original.arff
```

Listing 3.1: Transforming CSV to ARFF

At the initial transformation of data into the ARFF file format, it turned out that the message data field is misinterpreted by WEKA's transformation utility. Therefore, another filter employment was required in order to address the issue that text fields entered by individuals - the messages

[14] http://www.cs.waikato.ac.nz/ml/weka/arff.html, 09.12.2014

- were originally treated as nominal data. In this case, the filter turning nominal data into string was applied to the message field, as can be seen in Listing 3.2

```
weka.filters.unsupervised.attribute.NominalToString -C 2
```

Listing 3.2: Convert individual text containing fields to string data type

As for the POS adjusted dataset, Listing 3.3 points out the programmatic approach. In principle, the input file consists of all negotiation messages, where each row represents one messages. While iterating over messages, sentences and words, only the words with the corresponding tags (see Table 3.4) are kept in the output file, which is manually enriched with the class-label column after POS processing. Subsequently, the generated CSV file is processed and loaded into WEKA in a similar manner to the original dataset file described above.

```
// Define list of allowed tags
String[] allowedTags = new String[] { "NN", "NNS", "NNP", "NNPS", "JJ", "JJR"
    , "JJS", "VB", "VBD", "VBG", "VBN", "VBP", "VBZ" };
List<String> allowedTagsList = Arrays.asList(allowedTags);

// Initialize tagger
MaxentTagger tagger = new MaxentTagger(args[0]);

// File handles
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(new
    FileInputStream(args[1]), "utf-8"));
OutputStream outputStream =  new FileOutputStream("messages.out.txt");
PrintWriter printWriter = new PrintWriter(new OutputStreamWriter(outputStream
    , "utf-8"));

// Read line by line, i.e. iterate over negotiation messages
String line = "";
while ((line = bufferedReader.readLine()) != null) {

   Reader reader = new StringReader(line);

   // Tokenize sentences
   List<List<HasWord>> sentences = MaxentTagger.tokenizeText(reader);

   // Iterate over sentences
   for (List<HasWord> sentence : sentences) {

      // Tag words in sentence and iterate over tagged words
      List<TaggedWord> taggedSent = tagger.tagSentence(sentence);
      for (TaggedWord tw : taggedSent) {

         // If tag is white-listed, add to output
         if (allowedTagsList.contains(tw.tag())) {
            printWriter.print(String.format("%s_%s ", tw.word(), tw.tag()));
         }
      }
   }
   reader.close();
   printWriter.println("");
```

```
37      printWriter.flush();
38  }
39  bufferedReader.close();
```

Listing 3.3: Programmatic POS tagging and token filtering based on the Stanford POS tagger and tags representing nouns

Once the source data is in ARFF file format, further steps are undertaken with the WEKA toolkit. A step-wise description of the usage of WEKA for the particular experiments conducted in this work is beyond the scope. However, further details for handling the WEKA toolkit can be retrieved from the accompanying documentation[15].

[15] http://prdownloads.sourceforge.net/weka/WekaManual-3-7-11.pdf?download, 10.03.2015

CHAPTER 4

E-negotiation data analysis

The major aspect of this work is the extraction and analysis of emotions in negotiation records. Consequently, it is necessary to have a well-defined set of such transcripts, which are the basis of corresponding investigations. Ideally, negotiation data from real-world negotiations would be used, which would probably reveal the most natural results as there is no bias due to a laboratory environment. For this work, however, an aggregated set of experimental negotiations is considered to be sufficient to meet the desired objectives. As such, this chapter deals with the exploration of the provided dataset from various perspectives of the experiment setup, applied methods and tools, and characteristics of the records gained.

4.1 Negotiation experiment

As already mentioned in the introduction section of this chapter, the dataset was generated through a laboratory experiment conducted by Mitterhofer et al. across four Universities in The Netherlands, Austria and Germany [53]. In total 224 subjects were part of the experiment, which was designed to consider the different support system components of the chosen e-negotiation system. For the scope of this work, the dataset used is the same as that used by Hippmann, consisting of negotiations with and without decision support enabled [38]. Under this constraint, 114 people - in fact, students with average negotiation experience from participating in negotiation courses - were asked to represent one of two fictitious companies in the aviation sector, one located in Austria the other in Ukraine. Based on a participant assessment with 95 respondents, the average age of participants was 25.29 years with subjects' age ranging from 22 up to 46. The gender distribution was almost equal, marginally tending towards a female majority. The English and negotiation skills where examined based on self-reporting on a five point Likert scale ranging from 1 (no skills) up to 5 (excellent skills). The average English language competence was found to be quite good (3.95 on average), indicating low bias regarding potential language issues during the negotiations, which were conducted in English. However, the negotiation skills were estimated to be moderate, indicated by an average of 2.59. In terms of nationality, the majority of participants were from The Netherlands (45%) and Austria (23%), complemented by

students from 18 countries around the world including Finland, Hungary, Italy, Sweden, Bulgaria, France, China, Iraq and Russia - each group making up between one and four percent of the test group [38].
As a company representative, participants had to negotiate with their opponent on the subject of a possible joint venture in a bilateral negotiation setup. A total of seven issues were put on the table: the future revenue shares, the occupation of the board, establishing of a secrecy clause, the contract duration, the payment of common workers, the court of jurisdiction, and how Ukrainian workers were to be compensated. Each negotiating party had to stick to a given set of preferences regarding each issue, with the predefined positions designed slightly in opposition to one another [53].
In order to familiarize themselves with the system and avoid any bias caused by incorrect tool utilization, subjects were briefed and introduced to the selected e-negotiation tool one week before the actual negotiation. In particular, the chosen tool for guiding the electronic negotiations was Negoisst [74], which is described in detail in section 4.2. Both general and party-dependent negotiation case information was provided the day before the actual experiment period started. The negotiations were scheduled to last for a maximum of two weeks, though, negotiators were allowed to conclude or abort negotiations at any time within the given time period. In order to avoid scenarios where negotiation partners could communicate outside the NSS, each dyad was setup such that the two parties were located in different universities [38].
As mentioned at the beginning of this section, the experiment was not explicitly conducted for this thesis but for a larger research project, therefore there are several aspects that must be considered when analyzing results for the purpose of this work. For instance, Negoisst's decision support component was used for investigations regarding the impact on negotiation outcomes [53]. Thus, some negotiations were accomplished with DSS enabled, which allowed negotiators to compare offers via utility values throughout the negotiation process.

4.2 E-negotiation system

The experiment outlined in the previous section was conducted using the web-based NSS Negoisst. Written in Java it follows a client-server architecture that relies on a three-tier approach (data, application and presentation layer) with a rather small, i.e. dumb in terms of business logic, client accessible through any web browser. However, Negoisst is an e-negotiation tool that aims for the achievement of two essential objectives in the context of electronic negotiations, namely the unambiguous exchange of messages, and the system's provision of intuitive, flexible and user-friendly interaction for the negotiator [74]. Thus, exchanged artifacts should be as structured as necessary, but as unstructured as possible in order to ensure the user's full control over the system.
A core concept of an NSS is that messages can be sent between negotiators. Such messages consist of various characteristic fields, e.g., time, sender, recipients and type, and are of a semi-structured nature. Derived from plain messages, an evolving business contract is part of every negotiation. This contract is represented as a versioned document resulting in the final business contract document at the end of a successful negotiation. Taking those aspects into account, it makes sense to map negotiations to the so-called DOC.COM framework. The framework

connects the concepts of versioned documents, message records, negotiation hierarchies and partners accordingly as illustrated in Figure 4.1 [75]. Negotiators have the ability to semantically enrich the content of messages by adding category-value pairs to written text. Those tuples are specific to contracts, branches or the actual negotiation, and typically represent the issues under negotiation.

Negoisst uses the DOC.COM framework as a foundation and aims to fulfill the requirements of an e-negotiation system mentioned above. As such, it supports semi-structured message exchange with the ability to specify the type of each message. The supported types in Negoisst reflect the way statements can be expressed according to Searle's Theory of Speech Acts [76]. Searle's theory deals with the sort of speech used to give intended meaning to statements and defines the following speech acts: assertive (descriptive), commissive (expressing intention to perform actions), directive (attempt to make others to perform actions), expressive (reveal personal psychological state) and declarative (institutionalized acts, e.g. baptism). Derived from those speech acts, a message can be of the type request, offer, counter-offer, accept, reject, question or clarification, depending on the sender's intention of the message. A predefined negotiation protocol organizes negotiations such that the way of exchanging messages of various type follow a certain process. For example, a request message must be answered by a message of the type reject, counter-offer or accept.

In addition, authors of messages have to position a particular sent message in the red or the green message area, limited by the constraints set by, for example, message type. These message areas support negotiators in keeping track of context and status of a negotiation by dividing communicated messages into formal (red) and informal (green) kinds of messages. The idea behind this split boils down to the fact that messages are either legally relevant (messages in the red area) or simply help to evolve the discussion.

Besides document management and communication support, Negoisst includes a decision support component. The purpose of this component is not only to evaluate offers during the negotiation, but also to measure preference structures of negotiators. The latter is supposed to help negotiators be clear about their expectations concerning their preferences and therefore to avoid conflicts. By numerical utility values and graphical representations, negotiating parties can obtain the negotiation progress and take action accordingly [17].

Finally, Negoisst provides the possibility of satisfiability checks. This feature compares current results of negotiations to objectives and resources (available stock, budgets, terms of business, etc.) of the corresponding negotiation party. By considering different criteria of satisfiability, it is possible to monitor the progress of satisfiability as a contract document evolves.

Figure 4.2 points out some of the concepts with respect to message exchange in the web Graphical User Interface (GUI) of Negoisst. In particular, general attributes of a message such as title and recipients are visible in *a)*. The area marked *b)* reflects the concept of message types as the author can choose here the intended typo of the message. The possibility of enriched message text is illustrated in *c)*. The text blocks on a gray background indicate issues and their values predefined for the current negotiation. The utility value [41] in *d)* is the measure of fit to the negotiators preferences according to the current status of the contract and is therefore related to decision support as well as satisfiability checks. In *e)*, the category-value tuples representing negotiation issues are listed and editable so that the negotiator can make adjustments throughout

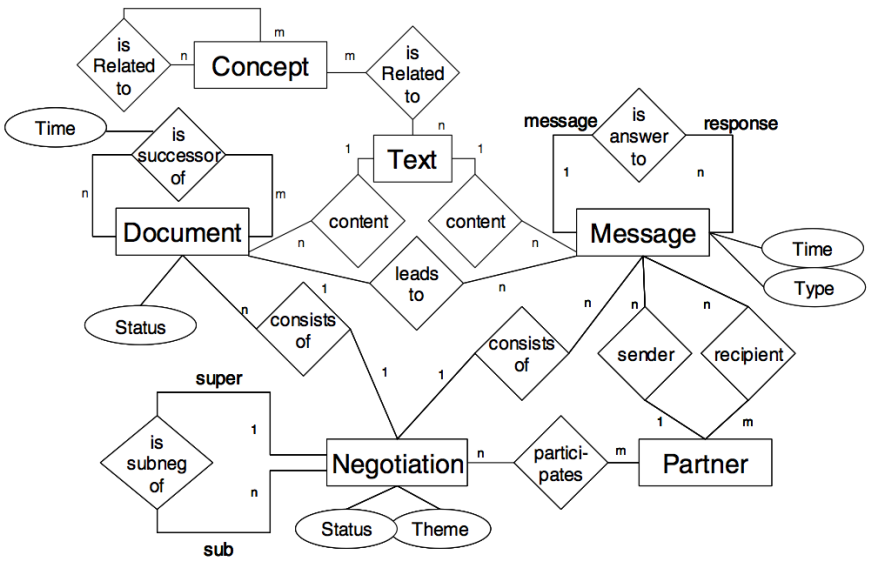

Figure 4.1: DOC.COM framework showing the relationships between the major concepts involved in e-negotiations [74]

the negotiation process.

4.3 Negotiation transcript description

After describing the experiment setup in the previous section, this paragraph describes the output of Negoisst, the NSS utilized for the experiment. The output data is principally available as a spreadsheet and consists of 42 data columns, where only twelve are originally came from the experiment described in section 4.1. The remaining columns are part of an enhanced experiment explained in section 5.1. Seven out of the twelve relevant data fields contain IDs, that uniquely identify negotiation, message and negotiator for each message exchanged. An additional field for indicating the rank of the negotiator in a certain negotiation is also attached, i.e. “1” for the initiator, and “2” for the responder. The flag “Agreement” illustrates whether a negotiation was successful (“1”) or failed (“0”). Next to this, the “DSS” flag exhibits the usage of the decision support component in a certain negotiation experiment setting as described in section 4.1. The negotiation system is furthermore capable of keeping track of the phase, during which a message was posted. As such, the three-phase model of e-commerce is employed. This approach suggests that a corresponding transaction starts with a phase of getting to know each other, i.e. the searching phase. In the next step - the negotiating phase - the relevant details are discussed and offers exchanged. In the final fulfilling phase, the negotiators arrange concluding issues such as logistics and payment [74]. Last but not least the actual content of a message is stored in a separate field. However, as those messages are enriched by concrete values for the negotiated issues, these texts contain special labels with suggested attribute values as well.

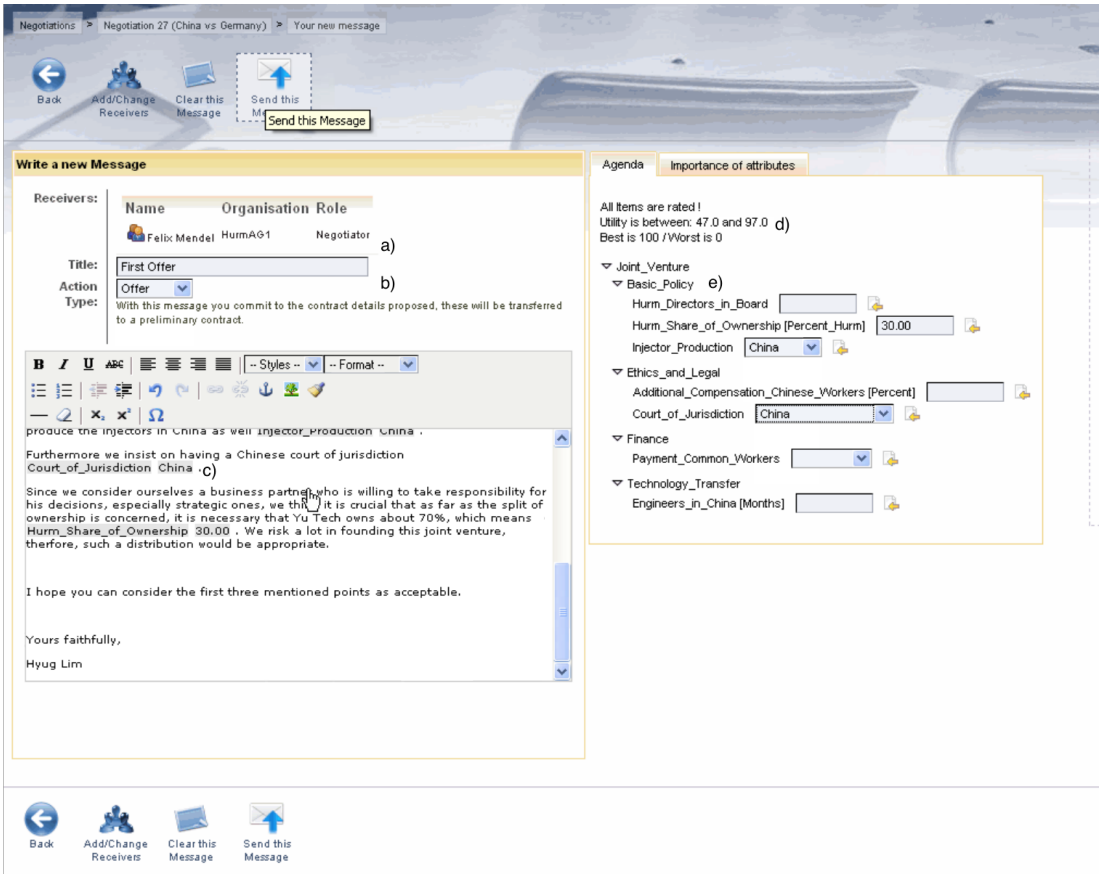

Figure 4.2: Sample screenshot of message compiling section of Negoisst including the concepts described in this section[1]

The number of distinct negotiators was actually 114, which is expected due to the experiment setup described in 4.1. As the negotiations of the experiment were set up in a bilateral form, the number of involved parties, i.e. persons, per negotiation is two. In total, 730 messages were exchanged in 57 negotiation sessions. Out of those 57 negotiations, 38 were successful meaning that this set of negotiations ended with an agreement between the negotiating parties. The rest of the negotiations (19) failed without an agreement. In terms of DSS, 32 negotiations were conducted with decision support, while the remaining 25 negotiations took place without it.

CHAPTER 5

Results & discussion

The following results chapter actually deals with the findings revealed as a result of the approach described in Chapter 3. As such, it not only contains results gathered from the implementation of machine learning methodologies, but also contains insights concerning the dataset under investigation. Additionally, the results are discussed and interpreted in order to promote ideas for future work in that arena.

5.1 Preliminary assessment towards emotions

The provided dataset not only contains contains data generated directly from the negotiation experiment, but also additional data resulting from further assessment of the affect involved. The approach selected approach to do so was MDS, which is described in subsection 2.4.1, as researched by Hippmann [38]. Therefore, a total of 69 raters in three independent groups initially sorted the negotiation messages assigned to those groups. The criteria for sorting was emotional similarity, whereas raters were not only to rank the messages according to that criteria, but also form decks of messages that were perceived to convey the same emotional state. Having done so, the raters had to describe each deck in terms of associated emotion. Those textual explanations can be found in 26 columns in the dataset, and since the number of raters for each group was slightly different, those columns contain missing values as well.
As the next step, a similarity matrix (i.e. a cross-tabulation of messages) for each rater was created [11], indicating which messages were assigned to the same deck. Averaging the matrices over the rates of each group resulted in three similarity matrices with values ranging form 0 to 1. The author furthermore decided on a two-dimensional space, which was most appropriate considering the obtained stress values for models from one to five dimensions. This resulting space was found to be a concept very similar to Russell's theory of an affective circumplex from Russell [67].
Russell suggests that emotional states can be depicted in a two-dimensional bipolar space, which arose from the idea that emotions principally consist of the factors pleasantness-unpleasantness

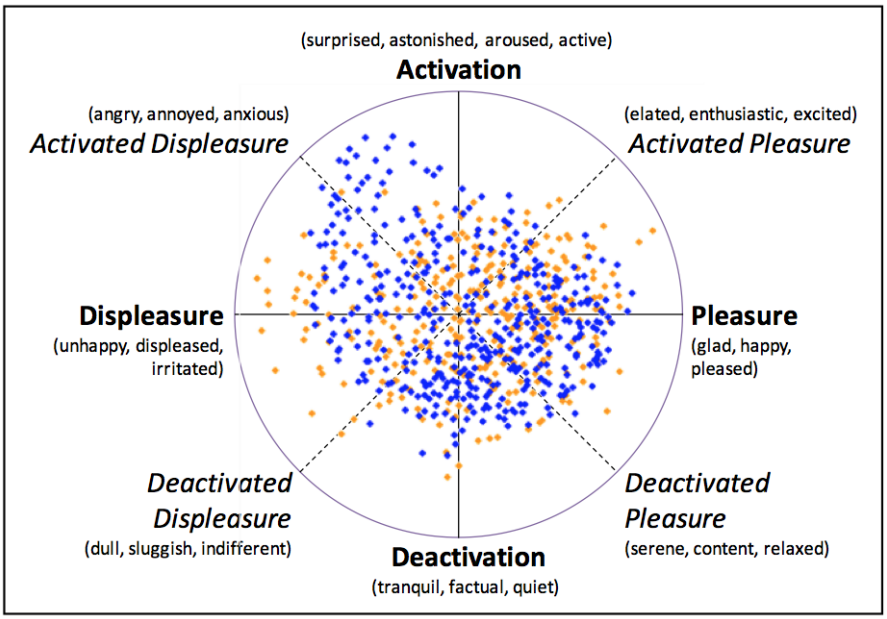

Figure 5.1: Affective space of messages (dots) in the given dataset including dimensional poles according to Russell's circumplex model of affect [38]

and attention-rejection [73]. A series of experiments employing different measurement methods and approaches delivered highly similar models of a two-dimensional space, which is put on the the axes "pleasure" and "arousal". Consequently, specific emotions can be accurately approximated by a vector of pleasure and arousal resulting in a circumplex structure. Russell and Barrett further examined the concept of the circumplex model by the concept of core affect, which uses the terms pleasure and activation for the two bipolar dimensions [68]. Core affect refers to a form of elementary affective feelings, where an individual is in a state of core affect at any point in time, even if it is only a neutral state. Pleasure being an indication of a subject's positivity, this dimension is also named positive-negative or hedonic tone, or can also be referred to as valence. Activation is defined as the subjective perception of mobilization, i.e. whether an individual is willing to act or to rest. Other terms for activation, then, are arousal, activity or tension [16,68]. However, rotating the dimensions of by 45 degrees results in a model described by Watson and Tellegen. The dimensions in that case are positive affect (low-high) and negative affect (low-high) [86].

Taking this into consideration, Hippmann enhanced the MDS outcomes by rotating the spaces generated by the three assessment groups such that their axes (i.e. valence and activation) were aligned. Consequently, the data fields "Valence", "Activation", "PosAct" and "NegAct" in the dataset indicate the underlying emotional state of each message, as the values for each dimension reach from 1 (high) to -1 (low). Figure 5.1 illustrates the dimensional configuration according to Russell as well as to Watson and Tellegen, and marks out each message in the affective space.

In contrast to the graphic in Figure 5.1, Figure 5.2 displays messages (dots), split into portions

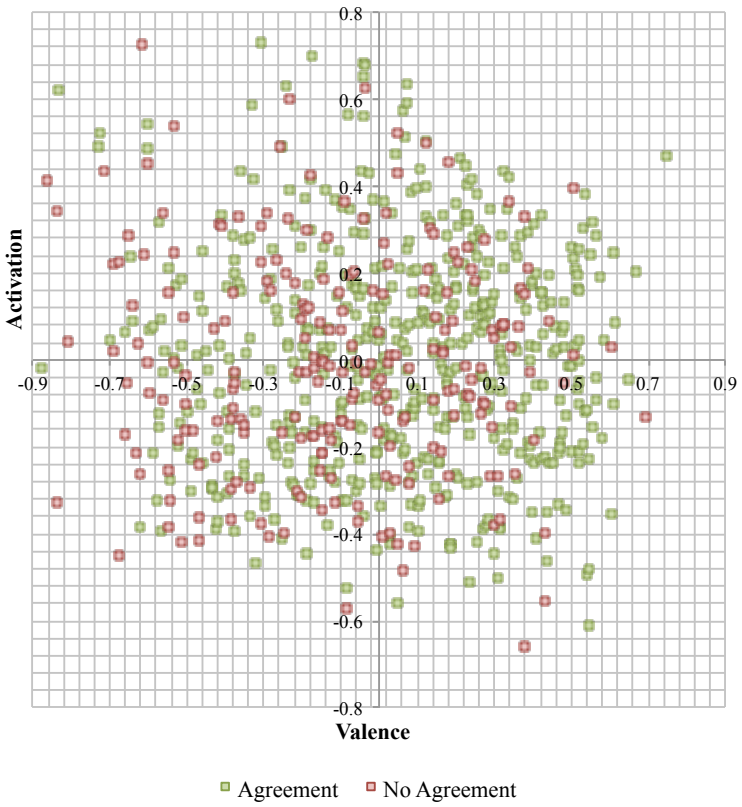

Figure 5.2: Representation of messages over the space of the circumplex model of affect, split into messages of negotiations ending with agreement and without agreement.

of messages of successful and unsuccessful negotiations. The space, however, is of the aforementioned two-dimensional kind of activation and valence, i.e. the circumplex model of affect.

Another relevant issue that can be observed at this stage is the structure of evaluation messages formulated by participants in the assessment groups. As mentioned above, the assessment groups in the MDS experiment were of different sizes ranging from 21 up to 26. This causes a lot of structural missing values in the data set. In turn, four columns representing the assessment group with the lowest number of participants contain 480 missing values (66%), while the smallest assessment group caused one column consisting of 235 missing values (32%). The other instances of missing values are negligible as in only one case the 1% mark is reached. The number of distinct values in the assessment values, which represent the deck descriptions given by the test subjects, range from 8 up to 31.

5.2 Standard descriptive statistics

In this section some information about the nature of the investigated dataset in textual and visual form is provided. Considering the affective dimensions Hippmann extracted from the negotiation transcript, valence values range from -0.878 up to 0.748 with a standard deviation of 0.332 over the whole dataset. Compared to the dataset containing only successful or only unsuccessful messages, this value is slightly higher. Furthermore, Table 5.1 reveals that the range between the minimum and maximum value is larger for negotiations ending in an agreement. Regarding quantiles (0.25, 0.5, 0.75), valence for messages in unsuccessful negotiations is generally lower than the same value measured for successful negotiations. The same trend is valid for the mean value: the mean over the full dataset is zero as this is a result of the MDS method. Detailed figures dealing with descriptive statistics of valence for the given dataset can be obtained in Table 5.1. The distribution of messages over the valence dimension with respect to the negotiation outcome (i.e. agreement and no agreement) can be obtained in Figure 5.3. Visually, the graph suggests a normal distribution of message valence for the whole dataset. The situation for the dataset split into successful and unsuccessful negotiations is rather unclear, because the share of messages of unsuccessful negotiations tends to decrease as the valence values increases. The application of unpaired t-tests validates the visual prediction by comparing the corresponding means [58].

Statistical key figure	**Messages of negotiations**		
	All	**Agreement**	**No Agreement**
n	730	505	225
Mean	0.000	0.047	-0.105
Standard deviation	0.332	0.324	0.326
Minimum	-0.878	-0.878	-0.863
Maximum	0.748	0.748	0.693
0.25-quantile	-0.224	-0.184	-0.359
Median	0.019	0.074	-0.096
0.75-quantile	0.258	0.295	0.148

Table 5.1: Statistical key figures for valence

In Table 5.2 one can find the corresponding variables and hypothesis for a valence related t-test.

t-test variables	$\mu_{agreement} = \mu_{no-agreement}$
Mean	0.047, -0.105
Standard deviation	0.324, 0.326
Degrees of freedom	728
p-value	$1.012 * 10^{-08}$

Table 5.2: Summary of t-test of means of valence

By employing an unpaired t-test, the means of valence for messages of successful and unsuccess-

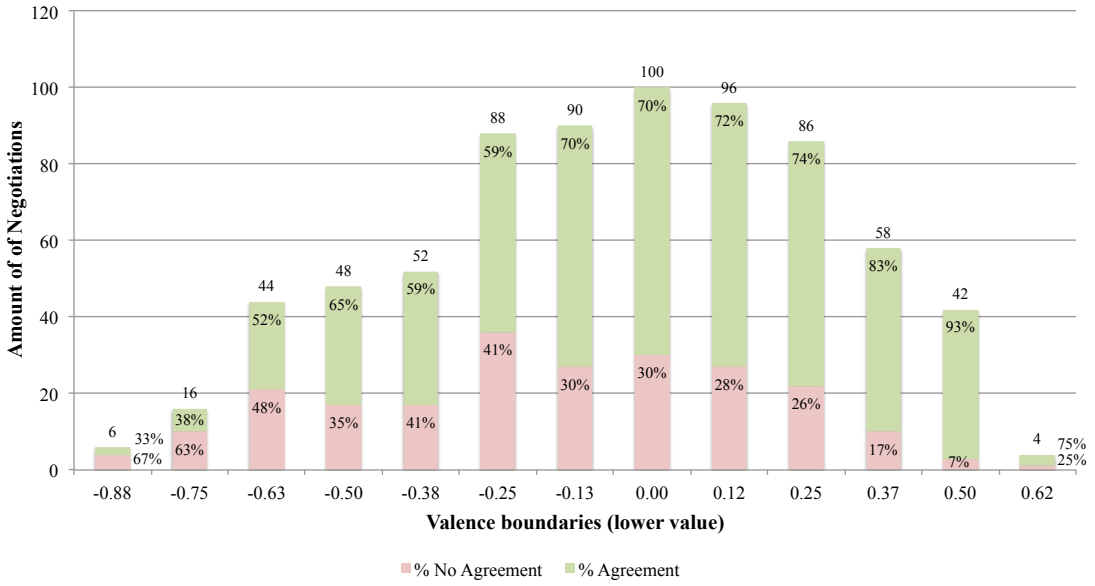

Figure 5.3: Distribution of messages over the dimension of valence

ful messages can be compared with respect to the equality of the means. Under the assumption of a significance level of 5%, the examined means differ from each other significantly (p-value = $1.012 * 10^{-08}$). Therefore, the findings of the applied t-test support the visual presumption that the share of messages from unsuccessful negotiations decreases with increasing valence values. Concerning activation, some values were shown to deviate from the statistical figures for valence. In particular, the mean of activation for unsuccessful negotiations is slightly lower than for successful ones and all quartiles show a higher value for negotiations without agreement than for successfully concluded negotiations. The values for activation range from -0.657 to 0.732, which is the actual range for activation values of messages of successful negotiations. Table 5.3 displays additional statistical data of activation values in the given dataset. Similar to the illustration of the distribution of valence values, Figure 5.4 shows the histogram of activation values, proposing a normal distribution for the whole dataset. However, the distribution of messages between successful and unsuccessful negotiations regarding activation values seems to be flatter than for valence. For the sake of assessment of the means for activation of messages, the same unpaired t-test as above is applied, which is summarized in Table 5.4. The obtained p-value of 0.558 reveals that there is no significant difference between the means of the two data subsets. As such, in contrast to the explorations for valence shown, activation means for messages of successful and unsuccessful negotiations tends to be the same.

As the class for each negotiation message is derived from the two values of valence and activation, the findings of the present and the preceding section could potentially indicate bias. While activation tends to be balanced with respect to the outcome of an negotiation, valence in the given dataset perhaps induces bias as the means of successful and unsuccessful negotiations differ significantly. This is particularly problematic due to the fact that the number of messages

Statistical key figure	Messages of negotiations		
	All	Agreement	No Agreement
n	730	505	225
Mean	0.000	-0.004	0.009
Standard deviation	0.256	0.248	0.271
Minimum	-0.657	-0.657	-0.570
Maximum	0.732	0.732	0.631
0.25-quantile	-0.191	-0.191	-0.185
Median	-0.006	-0.012	0.014
0.75-quantile	0.173	0.170	0.194

Table 5.3: Statistical key figures for activation

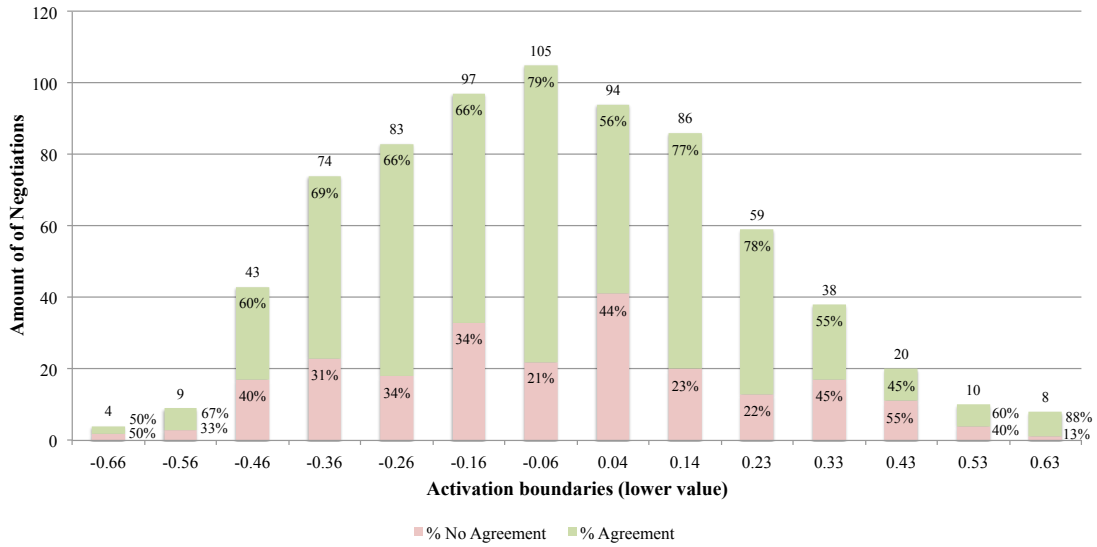

Figure 5.4: Distribution of messages over the dimension of activation.

of negotiations ending with an agreement is more than twice as high as those resulting without agreement. Thus, the separation along the axis of valence could cause distortion, as messages of successful negotiation are far more numerous in the given dataset.

5.3 Learning results

This section contains the results of learning algorithms applied to negotiation messages according to section 3.2. Graphs and tables in this and subsequent sections correspond to the naming schema defined in subsection 3.2.8 (e.g. #01.NBM stands for NaiveBayesMultinomial applied in combination with experiment setting one). The results in this section are discussed from several different perspectives. Firstly, performance and findings are considered individually by

t-test variables	$\mu_{agreement} = \mu_{no-agreement}$
Mean	-0.004, 0.009
Standard deviation	0.248, 0.271
Degrees of freedom	728
p-value	0.558

Table 5.4: Summary of t-test of means of activation

learning approach. As such, those subsections handle algorithm specific aspects, trends and findings. Secondly, results across applied learning approaches are compared accordingly. Finally, the influence of attribute selection and other pre-learning activities are discussed separately as they turned out to have a direct impact on results. However, before going into detailed result discussion of the results, preliminary notes valid across all experiments are set out below.

5.3.1 Preliminary notes

Basically, experiments were conducted with predefined parameter settings given in section 3.2. Some variables, though, required empirical examination in order to optimize the settings. In particular, this is true for the number of features selected for classification and algorithm dependent settings. The former was not only done for the sake of result optimization, but also for reasons of time and resource complexity: it is simply not feasible to work with settings consisting of vast numbers of attributes (> 45,000), such as uni- and bigrams. (Table 5.5 illustrates the number of extracted features per experiment setting.) In addition, it could be observed, that to a certain extent, a decreasing number of features improved performance of learners continuously. As a result, experiments were conducted with the top 1000, 200, 100 and 50 features according to discrimination power determined by Information Gain (IG). Regarding algorithm dependent

Experiment setting	#01	#02	#03	#04	#05	#06	#07	#08
No. of attributes	6939	5491	6523	5200	6831	5405	49118	46128
Experiment setting	#09	#10	#11	#12	#13	#14	#15	#16
No. of attributes	48687	45822	49010	46041	7146	7072	46786	46712

Table 5.5: Number of attributes available for each experiment setting

configuration, the approach to find proper parameter settings was another exploratory activity. Most notably, the optimal settings for continuous numerical parameters (e.g., a parameter indicating intensity of pruning for the J48 algorithm), but also for discrete numerical parameters (e.g., number of nearest neighbors for IBk), were approached individually for each experiment run. Clearly, this could have resulted in a local optimum in terms of algorithm performance indicators rather than a global optimum. However, the effort of searching for the global optimum for each experiment run manually would have blown the scope of the present work. Furthermore, it is not expected that the further adjustment of classifier configuration settings would drastically change the structure of the results obtained. Results described in the upcoming sections are ex-

tracted based on best accuracy, e.g., 01.SMO was applied with the four different feature space sizes as mentioned above, and several classifier configuration settings - the values for accuracy, precision, recall and F-score were taken from the result-set with the best accuracy observed.
In addition to the overall performance analysis of learning algorithms, pair-wise comparisons of variations in preprocessing and representation settings are conducted in order to evaluate the influence of stemming, stopword removal, n-grams and dataset adjustments. As such, pairs of experiment settings are selected, which only differ in a single aspect. For instance, comparison of #01 and #02 reveal the impact of stemming, as does #03 and #04, #05 and #06 etc. The full list of pair-wise comparisons is recorded in Table 5.6.

Aspect	**Comparison**	**Compared settings**	**No. of pairs**
Stemming	none vs. Porter	#01 vs. #02, #03 vs. #04, #05 vs. #06, #07 vs. #08, #09 vs. #10, #11 vs. #12	6
Stopwords	none vs. Swish-E	#01 vs. #03, #02 vs. #04, #07 vs. #09, #08 vs. #10	4
Stopwords	none vs. Top 50	#01 vs. #05, #02 vs. #06, #07 vs. #11, #08 vs. #12, #13 vs. #14, #15 vs. #16	6
Stopwords	Swish-E vs. Top 50	#03 vs. #05, #04 vs. #06, #09 vs. #11, #10 vs. #12	4
n-grams	unigram vs. uni- & bigram	#01 vs. #07, #02 vs. #08, #03 vs. #09, #04 vs. #10, #05 vs. #11, #06 vs. #12, #13 vs. #15, #14 vs. #16	8
Dataset	original vs. POS adjusted	#01 vs. #13, #05 vs. #14, #07 vs. #15, #11 vs. #16	4

Table 5.6: Overview of pair-wise comparison of experiment instances in order to derive tendencies regarding the application of various aspects in the preprocessing and representation phases of the text analysis framework (see 3.2.2)

5.3.2 Probability based classification

WEKA's NaiveBayesMultinomial implementation is straightforward to use as it does not have any parameters to adjust. As such, only variations in selected features impact the results of the 16 individual experiment settings. The best results were obtained with 100 features in seven out of the 16 cases, while the limitation of 200 features delivered superior results in six experiment runs. 1000 features (best results in one case) and 50 features (two cases) seem to be feature spaces that were respectively too diversified and too narrow. The best classification results in terms of accuracy employing the Naive Bayes approach were achieved in experiment setting #08.NBM, i.e. utilizing uni- and bigrams without stopword removal, but with stemming on 200 features of the original dataset. #08.NBM furthermore delivered the best values for recall

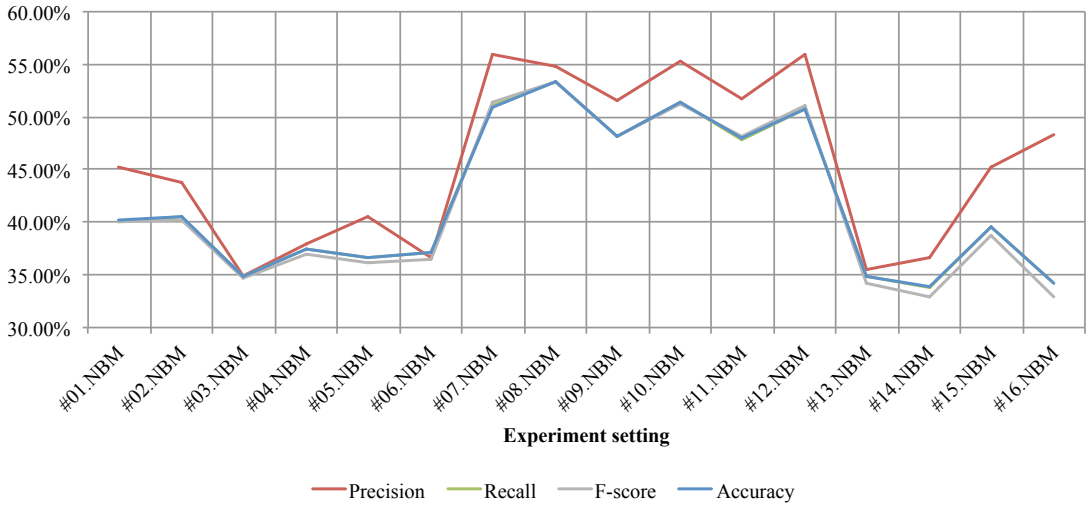

Figure 5.5: Performance measures of recognizing emotions in negotiation messages using Naive Bayes approach (WEKA's NaiveBayesMultinomial) across defined experiment settings

(53.30%) and F-score (53.30%). Precision, however, was best achieved with setting #07.NBM (56%) followed by #12.NBM (55.90%). Figure 5.5 graphically summarizes the results obtained for accuracy, precision, recall and F-score across the 16 defined experiment settings. In total, four experiment settings delivered values above 50% for all four performance measures (#08.NBM, #07.NBM, #10.NBM and #12.NBM), of which the top three are recorded in Table 5.7. Recalling Table 3.3, the four experiments settings mentioned therein have three aspects in common: employing the original dataset, using uni- and bigrams, and applying Porter's stemmer. Leaving Porter's stemmer aside, settings 07.NBM to 12.NBM delivered remarkable results, taking into account that the remaining settings hardly reach the 40% mark. Thus, the employment of bigrams seems to be responsible for enhanced performance, which is further reasoned in subsection 5.3.6. The worst results, on the other hand, were observed when using POS adjusted datasets, especially in combination with top 50 stopword removal. Considering the 16 results, the most reliably recognized class is AD with regards to F-score (twelve out of 16 times), followed by DP (three out of 16). The class delivering the highest F-score (59.10%) for setting #08.NBM was AD.

Concerning the impact of preprocessing and representation activities, unambiguous trends can be observed. Stemming improves results generated with NaiveBayesMultinomial (in six out of six cases), while stopword removal has a negative impact regardless of the stopword list applied (in four out of four (Swish-E) and six out of six cases (top 50) no stopword removal delivered better results). However, in three out of four cases the Swish-E stopword list performed better than the extracted top 50 list. Furthermore, using uni- and bigrams clearly improves classifier performance, which was observed in each of the eight cases. The POS adjusted dataset, though, decreased performance compared to the original dataset in every case.

Setting	Accuracy	Precision	Recall	F-score	Top class	No. of features
#08.NBM	53.29%	54.80%	53.30%	53.30%	AD (59.10%)	200
#10.NBM	51.37%	55.30%	51.40%	51.30%	AD (58.50%)	100
#07.NBM	50.96%	56.00%	51.00%	51.40%	AD (55.00%)	100

Table 5.7: Top three results from the perspective of accuracy, achieved by the application of NaiveBayesMultinomial; besides the performance measures, the number of selected features used for classification and the best class prediction (F-score) is outlined

5.3.3 Decision tree based classification

The decision tree implementation used for the experiments of the present work was WEKA's J48. Typically for decision tree modeling, the question of the intensity of pruning is the main factor influencing performance. As such, the essential configuration parameters in the corresponding WEKA implementation are "confidendeFactor" and "minNumObj". The former enforces stronger pruning when a lowver value is set, and the latter defines how many instances need to be in one leaf node of the induced tree (the higher the value, the stronger the pruning effect). It is reasonable to assume that a higher feature space would benefit from settings, that entail stronger pruning. This effect was not observed: each experiment run, therefore, required a manual search for a (local) optimum of settings. In terms of feature space, the best results were obtained six times considering 50 features, three times with 100 features, in five cases with 200 features, and in two cases 1000 features turned out to be the best choice. Taking into account accuracy, recall and F-score, settings performing the best way were: #04.J48 (accuracy = 40.15%, recall = 40.10%, F-score = 39.90%), #08.J48 and #06.J48 in that order, as can be seen in Table 5.8. From a precision perspective, those three settings deliver the top three results, with #08.J48 delivering the top value of 40%. The measures obtained suggest that the decision tree approach performs best with the unmodified dataset after stemming is employed. In terms of POS-based adjustments of data, however, the performance of the J48 classifier was negatively impacted. As such, #13.J48, #14.J48 and #16.J48 reveal the worst results regarding accuracy (lower than 33%), recall (lower than 33%) and F-score (lower than 31%). In regard to precision, #14.J48 (35.30%) performed better than #05.J48 (32.80). The findings mentioned along with further details are illustrated in Figure 5.6, which shows performance measures of J48 in combination with the defined experiment settings. Within the set of 16 experiment results, the class most often obtained with the highest F-score is AD (8), followed by DD (4), AP (3) and DP (1). In this regard, an F-score of 46.3% was achieved for the setting with the highest accuracy (#04.J48, 40.15%).

Clear statements can be made concerning the influence of stemming and POS tagging. The former was found to be beneficial for classifying with J48 as, in six out of six cases of pair-wise comparison, stemmed datasets surpassed their unstemmed equivalents. The latter, though, led to worse results than when the original dataset was employed in each of the four outlined matches. Stopword removal based on an extracted top 50 words list was less effective than both the Swish-E based settings (better results in zero out of four scenarios), and no stopword removal (only in one out of six cases). Removing words occurring on the Swish-E list improves performance

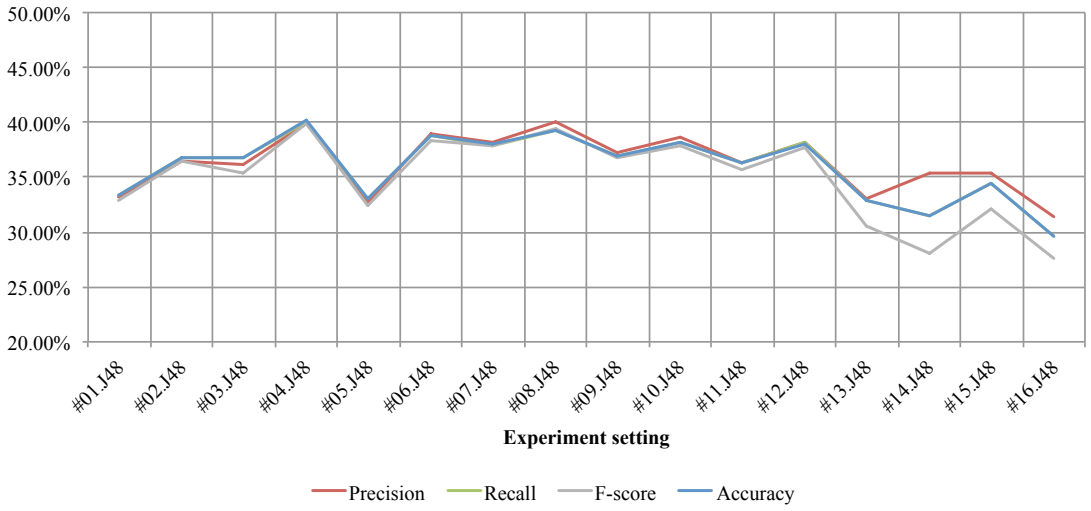

Figure 5.6: Accuracy, precision, recall and F-score observed by applying the decision tree implementation J48 across 16 predefined experiment settings

measures in two cases, while in the remaining two cases it does not. The situation regarding n-grams is also rather ambiguous. The application of unigrams and bigrams as features delivers better results than pure unigram representation in five out of eight cases.

Setting	Accuracy	Precision	Recall	F-score	Top class	No. of features
#04.J48	40.14%	39.90%	40.10%	39.90%	AD (46.30%)	200
#08.J48	39.32%	40.00%	39.30%	39.40%	AP (40.40%)	100
#06.J48	38.76%	38.90%	38.80%	38.40%	AD (44.60%)	50

Table 5.8: Best results achieved by J48, including the class with the highest F-score and size of feature space used in the particular experiment run

5.3.4 Support Vector Machine based classification

In order to analyze emotions in negotiation messages using SVM, WEKA's SMO algorithm was utilized. During experiments with SMO utilizing polynomial kernel, it turned out that three parameters have a crucial impact on classification performance, namely "buildLogisticModels", "filterType" and a complexity factor "c". No general rule about how these parameters should be combined could be derived, which led again to an approach of trying out promising combinations. However, the chance of obtaining optimal results in this case is higher, since the values for the former two parameters are discrete and finite, and the impact of c was observed to be comprehensible. Concerning feature selection, most results relied on 50 features (seven), while 100 features and 200 features generated top results for five and four experiment settings respectively. However, the best results were obtained using a feature space size of 100, as Table 5.9

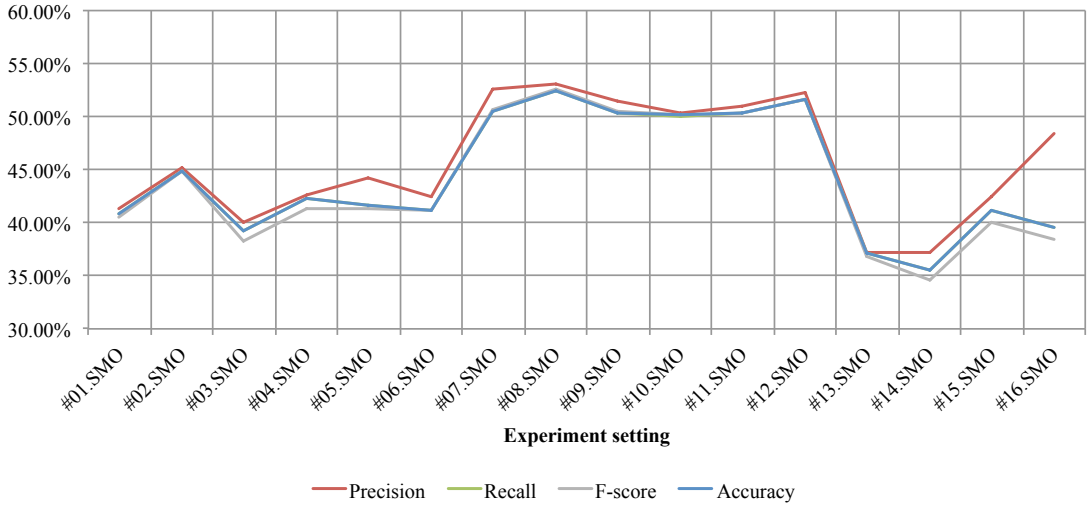

Figure 5.7: Illustration of accuracy, precision, recall and F-score achieved by SMO learning algorithm in 16 defined experiment settings

indicates. Taking the table into account, #08.SMO was found to produce the best classification results not only in terms of accuracy (52.47%), but also for the remaining three performance measures. Close to #08.SMO, #12.SMO and #07.SMO deliver above 50%-performance, where it should be noted that all experiment instances using uni- and bigram representation on the original dataset turned out to yield above 50% for all four performance measures (see Figure 5.7). Similar to Naive Bayes, observed results are outstandingly good in cases of bigram utilization on the original dataset (experiment settings 07.SMO to 12.SMO), which can be reasoned with feature selection as described in subsection 5.3.6. The worst classifiers were obtained in experiments relying on unigrams without stemming, especially on the POS adjusted dataset. It is, therefore, #13.SMO, #014.SMO and #03.SMO that are at the end of the performance ranking of the SVM classifier approach. Even so, accuracy values of 37.12%, 35.48% and 39.18% for the worst three experiment results are still moderate in context of experiments of this work. Regarding quality of class predictability, SMO based results revealed AD to be predictable relatively reliably, as in eleven out of 16 experiments AD was the class with the highest F-score. In the remaining five experiment runs, DD surpassed other classes three times and AP and DP were shown to have the highest F-score once each.
The employment of uni- and bigrams has a positive impact on the SMO classifier, as confirmed by pair-wise comparison (improvement in all eight scenarios). POS tagging and subsequent filtering clearly has a negative impact in terms of accuracy, as not in one single case using this approach showed improved accuracy of a classifier. Furthermore, the utilization of stopword removal tends to be of no value for performance improvement. Neither the Swish-E list (in no cases), nor the top 50 list (only in one out of six cases) seriously leveraged classification results to higher levels. However, stemming at least tends to improve results, as the top ranked experiments include stemming and in four out of six cases stemming had a positive impact on classifier

performance.

Setting	Accuracy	Precision	Recall	F-score	Top class	No. of features
#08.SMO	52.47%	53.10%	52.50%	52.60%	AD (57.50%)	100
#12.SMO	51.64%	52.30%	51.60%	51.70%	AD (56.80%)	100
#07.SMO	50.55%	52.60%	50.50%	50.60%	AD (54.10%)	100

Table 5.9: Top three classification results, best class recognition (F-score) and the corresponding choice of feature space size, accomplished by WEKA's SVM implementation SMO

5.3.5 Proximity based classification

The critical questions with respect to optimization of kNN algorithms are related to the value of k and the kind of distance weighting. As such, WEKA's IBk implementation provides adjustment settings accordingly ("KNN" and "distanceWeighting"). The values for k that delivered the best results varied randomly from one up to more than 20, where distance weighting did not improve results in every case. In fact, exploration made in search of the best algorithm configuration led to results that did not reach 39% accuracy in any of the cases. This is illustrated by the results from the best experiment run (#10.IBk), which reached 38.39% accuracy, 40.60% precision, 38.50% recall and an F-score of 38.50%. The next best results were achieved by #08.IBk (36.71% accuracy) and #01.IBk (36.58% accuracy) - as seen in Table 5.10 - where the precision scores of #08.IBk and #01.IBk, however, were surpassed by #12.IBk (39.10%) and #16.IBk (40.40%). Generally, IBk most often achieved best results when reducing the feature space to 50 attributes (eleven times), while 100 and 200 attributes revealed the highest accuracy in two cases each; for a feature space size of 1000, this happened only once. The lowest performance of IBk was obtained for #13.IBk, #14.IBk and #15.IBk, which delivered accuracy values between 30.82% and 32.88%. This suggests the conclusion that POS adjusted datasets cause performance measures to drop. Additionally, #11.IBk was ranked in the bottom three experiment instances in terms of precision and F-score, which can be seen in Table 5.8 alongh with the aforementioned results. The quality of classification for the sixteen experiment instances from a class perspective was best for DD (13), followed by DP (2) and AD (1). The corresponding F-score of DD classification for experiment #10.IBk was 49%.
Viewing the pair-wise comparison, stopword removal using the Swish-E list improves IBk results in three out of four cases, while the top 50 list negatively influences the results in five out of six scenarios. Accordingly, the top 50 stopword list performed worse in every case compared to the Swish-E list. The employment of the Porter stemmer improved classification in combination with uni- and bigrams in three cases, but otherwise only in one case, which makes a rate of four out of six. Not surprisingly when looking at the bottom ranked results, the POS dataset reduced results in three cases. However, in the case of application of uni- and bigrams in combination with the top 50 stopword list, the POS adjusted dataset achieved better results. As for the aspect of n-grams, an improvement could be observed in five of the eight cases when both uni- and bigrams were used for representation.

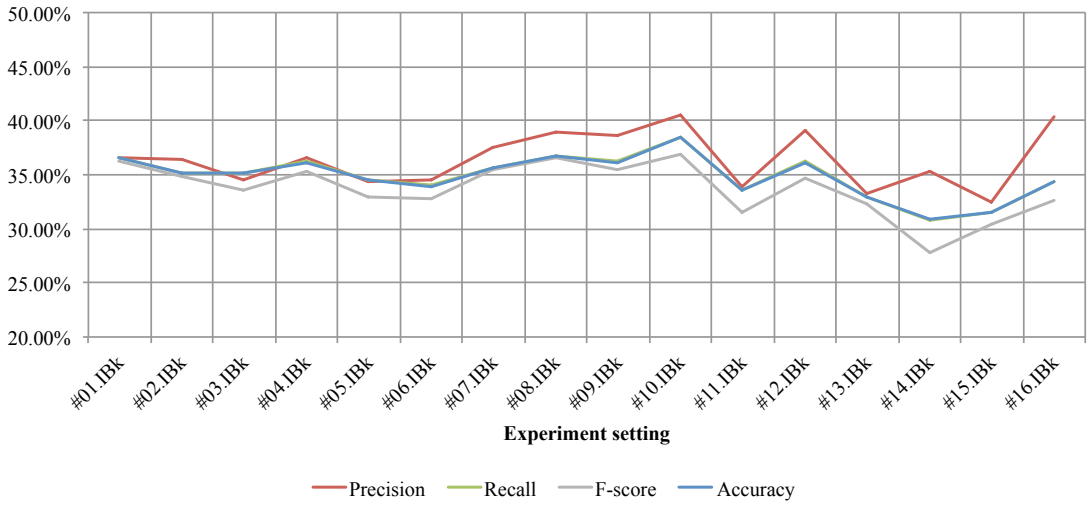

Figure 5.8: Performance measures accuracy, precision, recall and F-score drawn over 16 defined experiment settings, which were employed in combination with WEKA's kNN implementation IBk

Setting	Accuracy	Precision	Recall	F-score	Top class	No. of features
#10.IBk	38.49%	40.60%	38.50%	36.90%	DD (49.00%)	50
#08.IBk	36.71%	39.00%	36.70%	36.50%	DD (40.10%)	1000
#01.IBk	36.58%	36.60%	36.60%	36.20%	AD (44.90%)	50

Table 5.10: Top IBk results ranked by accuracy, including number of utilized features and the best classified class with respect to F-score

5.3.6 Feature selection

As results outlined in this section strongly rely on selected features used for classification, this subsection deals with the corresponding attribute ranking based on Information Gain (IG). Due to the varying experiment setups of this thesis, each of the 16 experiment groups has its own feature list. The reason for this is evidently the impact on features due to the application of stemmers, stopword removal, n-gram variations and POS tagging. The illustration and analysis of feature lists for each of the 16 experiment setups is too comprehensive to include in context of this work, therefore, only the features of the experiment setups of #01, #07 and #13 are compared. The choice of these three feature-sets is made due to the pureness and understandable meaning of words and pairs of words as stemming and stopword removal have no impact on the considered features. In turn, the selected settings allow the comparison of effects of bigrams in the feature space and POS related adjustments in the dataset. Setting #01, therefore, represents the group of #01 to #06, experiment setting #07 stands for setups #07 to #12, and POS related settings #13 to #16 are represented by #13.
Observing the features of #01 in Table 5.11, it is somewhat surprising that the majority of the

Features of #01		Features of #07		Features of #13	
IG	**Feature**	**IG**	**Feature**	**IG**	**Feature**
0.0607	not	0.0604	not	0.0499	am_VBP
0.0494	am	0.0514	you for	0.0437	Thank_VB
0.0404	Thank	0.0509	am	0.0376	nice_JJ
0.0388	I	0.047	I am	0.0336	is_VBZ
0.0372	You	0.0431	for your	0.0294	have_VBP
0.037	of	0.0424	I	0.0287	be_VB
0.036	for	0.0395	Thank	0.0273	looking_VBG
0.0358	nice	0.0389	Thank you	0.0265	happy_JJ
0.0336	this	0.0376	nice	0.0225	RRB
0.0336	is	0.0374	You	0.017	Have_NN
0.0273	forward	0.0367	forward to	0.0168	disappointed_JJ
0.0273	looking	0.0366	looking forward	0.0164	Sincerely_NN
0.0267	hope	0.0338	you want	0.0163	cooperative_JJ
0.0265	happy	0.027	are not	0.0139	proposition_NN
0.0262	reject	0.0264	this negotiation	0.013	Morning_NN
0.0211	concerns	0.0238	you dont	0.0128	save_VB
0.0197	disappointed	0.0203	glad to	0.0128	revenue30_NN
0.0163	cooperative	0.0178	your willingness	0.0127	remain_VB
0.0128	??Good	0.0176	minimum of	0.0122	I_VBP
0.0128	revenue30	0.0166	cannot make	0.0122	job_NN
No. of features (IG > 0.0): 40		No. of features (IG > 0.0): 95		No. of features (IG > 0.0): 23	

Table 5.11: Top ten and ten additionally selected features and their IG value with respect to the class attribute (N, AP, AD, DD, DP), produced by WEKA's InformationGainAttributeEval algorithm. The table shows attributes extracted from experiment setup #01, #07 and #13. The bottom line gives the number of features, where IG was determined to be higher than zero.

top ten features (i.e. all features except "Thank" and "nice") seem to be rather neutral in terms of emotional states. Clearly, as the features consist of unigrams only, the context of a word gets lost, e.g., "not" and "I" hardly indicate class membership without detailed knowledge of messages in the training set. However, as might be expected, features such as "hope", "happy", "reject" and "disappointed" are still on the list of features, which have IG values above zero. The appearance of "??Good" and "revenue30" furthermore indicates some experiment specific noise, or indeed, bias. The former is likely to be an issue caused by some special characters that were unhandled in the process of transformation from Negoisst to WEKA. The latter points out that the particular negotiation scenario influences the learning performance by the enrichment of negotiation messages with attribute values of negotiation issues. Surely, one could argue, a message containing the offer or request for a future revenue share of 30 is likely to be associated with a particular emotional class. However, such criteria produce models that are not even domain specific, but case specific. In total, 40 attributes of the prepared dataset for #01 were found

to have an IG value above zero, which does not mean, however, that additional features selected for learning cannot improve performance.
The top ten list of features with bigrams added to the feature space is similar to the list obtained for #01, though, the variations of the phrase "Thank you for your" (i.e. "Thank you", "you for", "for your") probably suggest that this phrase has quite strong power of discrimination regarding the five classes in the present case. Similarly, derivations of the phrase "looking forward to" appear in the feature list of #07. Furthermore, a major consideration for using bigrams was the capture of negations consisting of word combinations such as "not happy". Indeed, these kinds of negations could be observed in the feature list of #07, for instance "are not", "cannot make" and "you dont". Bottom line, experiment setup #07 results in 95 uni- and bigrams having an IG value above zero, which is more than double the value of #01. Obviously, the enhanced list of meaningful features - caused by introducing bigrams into the feature space - drastically increases performance of Naive Bayes Multinomial (NBM) (Figure 5.5) and SMO (Figure 5.7) learners, and generally tends to improves classification results, as revealed in pair-wise comparison of experiments in subsection 5.3.7.
In contrast to #01 and #07, features of experiment setup #13 consists of nouns, verbs and adjectives only. As such, it may be noted that words like "of", "for" and "this" are missing in the top 10 list. Instead, potentially meaningful tokens such as "looking" and "happy" made it into the top ten. Additionally, "Have_NN" and "RRB" reveal issues regarding incorrect tagging and tokenizing of POS tagged texts. The number of features with an IG value above zero is outstandingly low compared to compared to feature lists from #01 and #07, dropping to just 23.
The features outlined in Table 5.11 explain why the application of stopword removal does not boost classifier performance as predicted in subsection 3.2.3. Stopword removal was applied in order to neglect very frequently used words as their contribution to class discrimination was expected to be low. However, taking the top ten features of #01 into account, six out of the ten tokens appear on the Swish-E stopword list. The situation is even worse considering the top 50 stopword list: as illustrated in Table 5.12, eight words that appear on the top 50 stopword list are also on the top ten features list of #01. Thus, the employment of stopword removal approaches

Rank	Word	Frequency
3	of	4553
4	I	2801
6	you	2552
9	for	2324
12	is	2064
13	this	1520
22	not	930
45	am	463

Table 5.12: The eight words appearing in the top 50 stopword list, which are part of the top ten features list of #01 as well

can be understood to negatively influence classifiers due to the fact that determining features in terms of IG are stripped out in the preprocessing phase. Taking the selected three represen-

tative experiment settings and the findings regarding stopword removal into account, then, the number of features with positive IG values with respect to the given classes, tends to correlate with classification performance. In particular, NBM and SMO performance across experiment settings confirms this hypothesis, bearing in mind the superior results for settings #07 to #12. Furthermore, pair-wise comparison of according experiments supports this suggestion, observed in comparisons of n-gram and POS related aspects outlined in the following subsection.

5.3.7 Performance comparison & results reasoning

After viewing the results of classification individually by algorithm, this subsection compares and summarizes the outcome across the four applied learning approaches. As such, Table 5.13 shows the best result for each algorithm and thus reveals that NBM surpasses the other approaches in each of the four considered performance measures. Only experiments utilizing SMO get close to the top result achieved by Naive Bayes. A level below those two results, the best performances of J48 and IBk settled on around 40% accuracy. However, the F-score achieved for particular classes is still quite high, for example, the class with the highest F-score in experiment #10.IBk was DD (49.00%).

Setting	Accuracy	Precision	Recall	F-score	Top class	No. of features
#08.NBM	53.29%	54.80%	53.30%	53.30%	AD (59.10%)	200
#04.J48	40.14%	39.90%	40.10%	39.90%	AD (46.30%)	200
#08.SMO	52.47%	53.10%	52.50%	52.60%	AD (57.50%)	100
#10.IBk	38.49%	40.60%	38.50%	36.90%	DD (49.00%)	50

Table 5.13: Best results achieved by each learning method, selected by accuracy; for each result, the class with the highest F-score and the number of utilized features is outlined

A fundamental finding in this regard is the similarity in the performance of Naive Bayes and SVM when measured in accuracy. Figure 5.9 illustrates the superior results of both approaches in experiment scenarios #07 to #012, i.e. in settings where the original dataset with uni- and bigram representation was used. Corresponding feature selection revealed the discrimination power of bigrams and their ability to enlarge the list of meaningful features, which magnified performance as shown in subsection 5.3.6. In the remaining scenarios, the SVM approach outperforms other algorithms, which of the relative performance varies considerably in experiments #01 to #06. In turn, the decision tree approach and kNN were neither capable of drastically leveraging performance by using uni- and bigrams, nor by any other preprocessing step. However, each of the four algorithms stay significantly above the baseline, which is chosen under the assumption of an informed guess. Taking Figure 3.6 into account, one could assume that the share of messages assigned to each class is exactly 20%: the ideal case of equal distribution. In the present case, however, the share of DP slightly surpasses the remaining classes and reaches a value of 21.51%. Supposing that a learning method - or any individual - does not include in any additional information, but knows the actual share of classes in the considered dataset (and is, therefore, capable of making an informed guess), 21.51% would be the best value of accuracy,

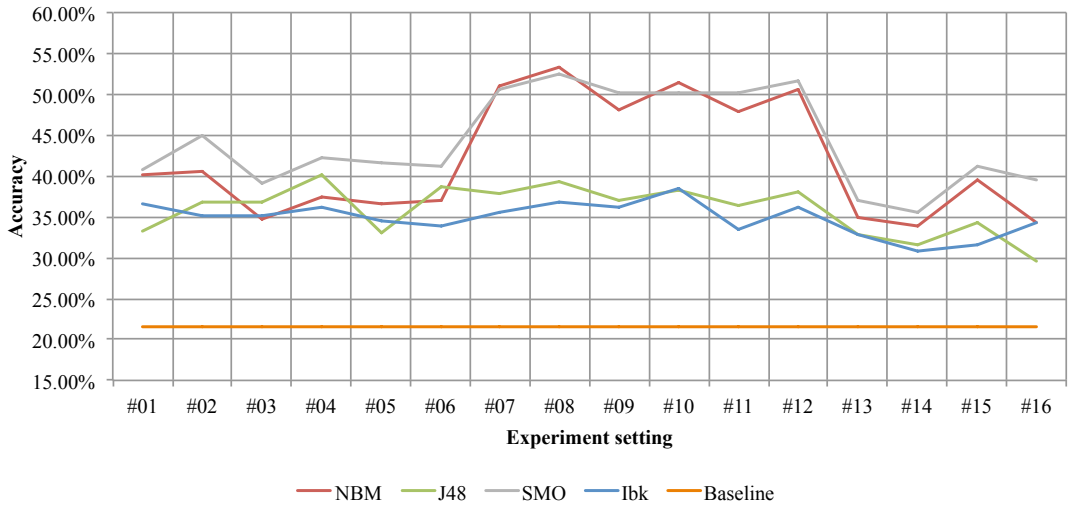

Figure 5.9: Comparison of accuracy of the four examined learning methods throughout the series of conducted experiments, showing the superior performance of NBM and SMO in experiment settings #07 to #12; the baseline is the share of the most frequent class (DP) in the original dataset of the this work (21.51%)

given that all instances are classified as DP. As shown in Figure 5.9, the utilization of a learning method outperforms the informed guess approach, i.e. the baseline, in any case.

Another aspect which was explored in the subsections of algorithm-specific results concerns the impact of preprocessing and representation related activities. To this end, Table 5.14 illustrates the summarized results of pair-wise comparison regardless of the particular learning algorithm. In terms of stemming, a tendency towards improved results can be seen as, in 20 out of 24 cases, experiment settings with a stemmed dataset delivered better results. Consequently, stemming seems to be beneficial for text analysis in the domain of text-based negotiations, even though stemming was found to be application dependent in subsection 3.2.3 and was therefore uncertain. Clearly not recommended is the application of stopword removal. Although the Swish-E stopword list achieved better results compared to the top 50 stopword list, applying no stopword removal at all performs better than using Swish-E (in eleven of 16 scenarios) or the top 50 list (in 21 out of 24 cases). An explanation for this behavior is given in subsection 5.3.6. A similar reasoning can explain the failure of POS adjusted datasets, which neglect many words (everything except nouns, verbs and adjectives) that are potentially essential for class determination. In this regard, a correlation between the performance and number of features with above-zero IG values (subsection 5.3.6) can be observed. As such, the corresponding assumption about classification capability of special word categories made in subsection 3.2.4 cannot be confirmed. Finally, the employment of bigrams in addition to unigrams had a positive impact in 26 out of 32 cases, especially in Naive Bayes and SVM related experiments as mentioned above. Subsection 5.3.6 in this chapter outlines potential causes for the observed performance improvement.

Aspect	Comparison	Success rate (RHS)
Stemming	none vs. Porter	20/24
Stopwords	none vs. Swish-E	5/16
Stopwords	none vs. Top 50	3/24
Stopwords	Swish-E vs. Top 50	3/16
n-grams	unigram vs. uni- & bigram	26/32
Dataset	original vs. POS adjusted	1/16

Table 5.14: Impact of preprocessing and representation aspects by pair-wise comparison according to Table 5.6; Success rate describes the amount of cases in which the comparison value on the right-hand side won the match

5.3.8 Class specific performance

Bearing in mind that the ultimate objective of NSS enhancement is to indicate the critical emotional development of negotiations, not all classes considered in this work are equally important. As Griessmair and Koeszegi have shown, negotiations drifting to a negative communication style typically result in termination of the negotiation process. In order to remediate in such cases of potentially failing negotiations, it may be assumed that in a real-world scenario it is more important to detect unpleasant and eventually aroused messages than those conveying positive or neutral emotions. Consequently, the classification performance of classes AD and DD are especially interesting in the scope of this work.
In this regard, detailed results on the class level of the best experiment results of NBM and SMO can be obtained in Table 5.15. The precision for activated displeasure reached more than 62% in both cases, while the recall value for NBM (56.10%) surpasses the recall values of the SMO equivalent by almost three percentage points. These numbers result in F-scores close to 60%. Furthermore, AD was the class best classified by the top result of J48 (#04.J48) as shown in Table 5.16. An F-score of 46.30%, however, is considerably below the performance of NBM and SMO. The classification performance obtained by IBk regarding AD was the worst, as the F-score did not even reach 36% resulting from the poor recall value of 29.70%.

	#08.NBM			#08.SMO		
Class	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
N	63.60%	43.20%	51.40%	50.70%	47.30%	48.90%
AP	53.90%	48.30%	50.90%	55.60%	51.70%	53.60%
AD	62.40%	56.10%	59.10%	62.20%	53.40%	57.50%
DD	44.40%	58.10%	50.30%	44.50%	53.70%	48.70%
DP	49.50%	60.50%	54.40%	51.80%	56.10%	53.80%
Weighted avg.	54.80%	53.30%	53.30%	53.10%	52.50%	52.60%

Table 5.15: Class specific performance measures of the top results achieved by NBM and SMO

However, the performance values for class AD are quite high compared to those of other classes, a tendency confirmed by another finding resulting from pair-wise experiment comparisons: by

aggregating the numbers of best individual class detection in terms of F-score for each of the 16 experiment instances across the four algorithms, it was revealed that class AD outperformed the other classes in 33 out of 64 cases. A possible explanation for the superior results for class AD is that individuals in emotional states found in the second quadrant (activated displeasure) of Russell's circumplex might use characteristic words and phrases in their messages, which therefore makes it easier for classifiers to assign the right class. Recalling Figure 3.5, it is possible to observe that, in addition, messages labeled with class AD mostly tend to occur on the right-hand side of the spectrum, i.e. the perceived emotional intensity of those messages is rather high. Thus, one could argue that highly emotionally loaded messages are easier for classifiers to categorize.

	#04.J48			**#10.IBk**		
Class	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
N	30.20%	28.80%	29.50%	32.90%	34.90%	33.90%
AP	40.00%	33.60%	36.50%	50.00%	21.00%	29.60%
AD	42.60%	50.70%	46.30%	44.40%	29.70%	35.60%
DD	43.70%	45.60%	44.60%	36.80%	73.50%	49.00%
DP	43.10%	42.00%	42.60%	38.90%	35.70%	37.20%
Weighted avg.	39.90%	40.10%	39.90%	40.60%	38.50%	36.90%

Table 5.16: Class specific performance measures of the top results achieved by J48 and IBk

While class AD is handled quite well compared to other classes, the situation for the other relevant class DD (deactivated displeasure) seems to be the opposite. Although DD was best recognized in 20 of the 64 scenarios, most of the 20 cases (13) were achieved in experiments conducted with IBk, which overall performed quite badly. As well as IBk, J48 was also unable to achieve an F-score above 50% for the class DD; the F-score of 49% generated in experiment #10.IBk, however, is moderate. When looking at the detailed values of #10.IBk in Table 5.16, it turns out that this F-score value was achieved by an uncommonly high recall of more than 73% in combination with the low precision value of 36.80%. Thus, the reason for the superior F-score values for DD achieved by IBk tend to be the result of a systematical bias (i.e., assigning a lot of negotiation messages to DD, which results in a high recall rate). Going back to the best results of NBM and SMO as outlined in Table 5.15, the class DD attained the lowest values across all classes in both scenarios and, as in the IBk results previously mentioned, the value for recall is significantly higher than for precision.
Regarding the classification performance of the relevant class DD, another specific experiment setting worth being looked at is #12.SMO, as for this experiment the F-score for DD is the highest compared to the DD F-score values of the top three results of NBM and SMO. Table 5.17 shows that besides an acceptable F-score value for class AD (56.80%), an F-score for DD of more than 52% could be achieved. The significance of this value, though, is doubtful; it is substantially leveraged by the good recall value (60.30%), while precision stayed far below 50% (45.80%). This confirms the observed tendency of high recall coupled with low precision with respect to class DD.
For the sake of completeness, it should be mentioned that while the class DP was classified in the

	#12.SMO		
Class	**Precision**	**Recall**	**F-score**
N	51.40%	48.60%	50.00%
AP	54.00%	47.60%	50.60%
AD	60.80%	53.40%	56.80%
DD	45.80%	60.30%	52.10%
DP	49.00%	49.00%	49.00%
Weighted avg.	52.30%	51.60%	51.70%

Table 5.17: Class specific performance of #12.SMO with adequate F-score for class DD

best way seven times and AP four times, in not one single case was the neutral class the class with the highest F-score. This can be explained by the suggestion that features typically indicating neutral messages are rare and/or hard to identify. This, along with the findings regarding classes other than the neutral one, implies that emotionally loaded messages are easier to classify than neutral ones.

5.4 Recommendations

Ultimately, the considerations, approaches and findings mentioned above can be put together in order to derive a recommended methodology for emotion recognition in text-based negotiations. Thus, in terms of preprocessing, the application of a stemmer such as Porter is highly recommended as in more than 80% of comparable experiments, performance was improved in stemming based scenarios. Furthermore, the best results in the series of conducted experiments were achieved when Porter was utilized. In this regard, it appears that the reduction of features and simultaneous impact on weighted term counts in the BOW model positively influences classifier performance. Contrarily, stopword removal should be avoided completely. Both, the application of the Swish-E stopword list and a list of the 50 most frequent tokens tend to decrease results, such as in the case in 80% of the comparable experiments observed. Additionally, removing all terms except nouns, verbs and adjectives is not recommended as classification performance of POS adjusted datasets was poor throughout the vast majority of experiments. The latter two methods should be disregarded as they not only drop "noise" in textual messages, but also remove highly discriminating features required for meaningful classification. In contrast to stopword removal and POS related approaches, the utilization of bigrams in the feature space introduces additional powerful features, used for message discrimination with respect to given classes. The positive effect of bigram employment is backed up not only by the pair-wise comparison of results (in 26 of 32 obtained scenarios), but also by superior performance of NBM and SVM in experiments settings such as #07 to #12, which rely on uni- and bigram features. Regarding further preprocessing and representation aspects, no clear recommendations can be derived from the present results due to the lack of comparable options. However, the former concerns tokenization, which was reasonably limited to a predefined delimiter set consisting of whitespaces, the dot (.), other boundary signs (,;:?!+$%-/\), brackets (()[]<>) and quotes (" ' " ');

for the latter, Bag-Of-Words (BOW) in combination with weighted feature count based on Term Frequency/Inverse Document Frequency (TF-IDF) was found to be the best choice as explained in subsection 3.2.4. Concerning the choice of learning method, the decision between the Naive Bayes and the SVM approach is quite unclear. Although, NBM achieved the best classification result with an accuracy of 53.29%, it outperformed SMO in only three of sixteen experiments. If a single choice must be made, an SVM approach such as SMO is recommended, as it tends to deliver more robust models than NBM as results throughout the conducted experiment settings showed minimal fluctuation.
1

CHAPTER 6

Conclusion & outlook

This concluding chapter summarizes the major outcomes and findings of the present work, especially with respect to the research questions outlined in Chapter 1. Additionally, the subsequent section 6.2 points out limitations of this work and section 6.3 provides starting points for further research in the field of emotion detection in text-based negotiations.

6.1 Main findings

The present work investigates options for automated emotion recognition in text-based negotiations. Before going into empiric research, it was necessary to complete some pre-work based on the first research question concerning possibilities for automated emotion recognition. As the given dataset contains an indication regarding the emotional direction of each message generated by the application of Multidimensional Scaling (MDS), supervised classification was chosen as the data mining approach of choice. However, the provided dataset does not contain a dedicated class label, but rather quantitative values for valence and activation according to Russell [67] [68]. As such, a class for each negotiation message was derived based on radius and location of messages in the bipolar, two-dimensional space, resulting in five classes: N (neutral), AP (activated pleasure), AD (activated displeasure), DD (deactivated displeasure) and DP (deactivated pleasure). Considering a common text analysis framework [4], 16 experiment settings were extracted, which differ regarding preprocessing, document representation and knowledge discovery. Those 16 experiment settings are combinations of promising choices regarding dataset preparation (original and POS adjusted), n-gram utilization (unigram and uni- and bigram), stopword removal (Swish-E list, list of top 50 words and an empty stopword list), and stemming (no stemming and Porter [65]). Regardless of those pre-learning steps, document instances are represented as Bag-Of-Words (BOW) using the weighted feature count based on Term Frequency/Inverse Document Frequency (TF-IDF). Furthermore, the 16 settings are combined with representatives of classifier families, which were to some extent proven to perform in the scope of text mining. In particular, classifiers related to decision trees, probability, SVM and

proximity were chosen and WEKA selected as the tool to provide the implementations of each classifier family. Thus, Naive Bayes Multinomial (NBM), J48, Sequential Minimal Optimization (SMO) and Instance-Based k Learner (IBk) were the methods applied in order to ascertain the emotional type of each negotiation message.
The empirical part of this thesis addresses the second research question, dealing with the selection and especially the implementation of appropriate learning methods on negotiation data. The execution of the corresponding experiments revealed some tendencies regarding applied text analysis steps, explored via pair-wise comparison of experiment settings. Thus, it was possible to show that stemming impacts results in a positive way. In contrast to stemming, stopword removal was found to be disadvantageous for classification performance, no matter which stopword list was applied. Feature selection analysis has shown, that words in the stopword lists and features conveying high Information Gain (IG) values overlap, which results in decreased discrimination power, causing lower accuracy. Besides the application of stemming, the utilization of bigrams together with unigrams led to increased classification performance. Most clearly, this was shown in combination with the Naive Bayes and SVM approach. The observed behavior is due to additional, meaningful phrases considered in the feature selection. Not only are negations consisting of two words (e.g., "are not") taken into account in bigram scenarios, but also derivatives of phrases like "looking forward to" are part of feature lists used for classification. Furthermore, POS tagging in the form in which it was applied in the present work is not beneficial for the performance of classifiers, due to the removal of words that are not nouns, verbs or adjectives. The neglect of all other words and tokens led to a lot of features meaningful in terms of IG begin dropped.
Regarding performance of examined classifiers - estimated by 10-fold cross-validation -it turned out that two of the four methodologies explicitly surpass the remaining algorithms in certain scenarios. Considering the baseline of 21.51%, which is the share of the most frequent class, DP, and, as such, the class all document instances would be assigned to in an informed guess scenario, the best results for each of the four chosen algorithms are clearly above this benchmark. However, while the best results for IBk (38.49%) and J48 (40.14%) settle around 40% accuracy, NBM (53.29%) and SMO (52.47%) exceeded the 50% mark. These remarkable results were achieved with the application of Porter's stemmer on the original dataset, and uni- and bigram features. Generally, in relation to IBk and J48, NBM and SMO performed outstandingly well in scenarios that included the original dataset and bigram utilization. This behavior is mainly caused by the fact that particular bigrams convey high values of IG and enlarge the list of powerful features accordingly.
Emotional indication could be an essential capability of potential next generation Negotiation Support Systems (NSSs) with respect to negotiation outcomes. As such, the identification of messages conveying aroused and especially unpleasant emotions is a key component in this regard. The class specific results for AD and DD are, consequently, of special interest. Breaking down the best results achieved by NBM and SMO to the class level, negotiation messages labeled AD (activated displeasure) were handled in the best way, as shown by the F-score values of 59.10% (NBM) and 57.50% (SVM). This finding is confirmed by the fact that AD was the best recognized class in terms of F-score in the majority of the experiment instances. The emotional intensity of messages located in the second quadrant (activated displeasure) is not only

perceived to be high by raters in the context of MDS, but also explains the superior performance of classifiers for this class. The performance regarding class DD (deactivated displeasure), on the other hand, was found to be quite poor, especially due to the fact that better F-score values were achieved particularly by high recall values.

6.2 Limitations

Clearly, the results recorded in the present work are insufficient with respect to the integration of components indicating the emotional class of textual negotiation messages in systems such as NSSs. Still, based on the present work, several starting points for improving learning performance in the given context can be offered. However, before discussing these improvements, fundamental constraints concerning the utilized dataset should not go unmentioned; regardless of the machine learning method selected to measure emotions in negotiation messages, the given dataset could cause problems for various reasons. Firstly, the fact that humans make spelling errors and typing errors could negatively influence the performance of learning methods as Sidorov et al. has shown as well [78]. The messages suffer additionally from missing line breaks and white spaces. Secondly, the laboratory experiment potentially introduces some bias to the results, as real-world decision-makers representing their company may behave differently to students playing a role. Such a limitation was considered by other studies as well, for example by Hine et al. [37]. The ratio of 2:1 between successful and failed negotiations is also something that should be kept in mind for further investigations, since Mohammad has shown the importance of a well distributed training/test dataset [54]. Even though the negotiation outcome was not directly involved in the learning processes of the conducted experiments, the valence of successful and unsuccessful negotiation messages deviates considerably. This is problematic due to the fact that the number of messages of successful negotiations (505) is much higher than the number of messages from negotiations ending without an agreement (225).
Regarding domain specificity, results are likely to be tied not only to the domain of negotiations, but also to the given case (described in Chapter 4). The latter assumption is based on the fact that, in terms of Information Gain (IG), tokens representing attribute values of negotiated issues are important for class discrimination. Corresponding features and trained models are hardly transferable to negotiations dealing with other issues and are therefore useless for classification. In order to get rid of case-specific features and bias, it is advised that case-specific words and phrases in document instances in the training dataset are sorted out or replaced by generalized artifacts. However, proof of the assumptions regarding domain specific issues are pending and require a separate, independent dataset consisting of negotiations messages labeled accordingly.

6.3 Future work

Potential space for improvement concerns the decisions made during the preprocessing phase of text analysis. As such, the set of delimiting characters used for tokenizing could be optimized or, to go even further, the application of more sophisticated, machine learning based approaches as suggested by Nugues [55] or Fares et al. [23] could ultimately raise classification performance. Concerning stemming, experiments were executed using Porter [65] or neglecting stemming

completely. Since stemming was found to be beneficial for learning performance in most of the cases, it is proposed that this preprocessing step be given more attention in future work. For example, a heavier stemmer such as the Paice/Husk stemmer [61] could boost learning performance as it did for Madariaga et al. [18]. In terms of stopword removal, methods applied in the present experiments led results to drop compared to those attained without stopword removal. However, although the overall performance suffers with the application of stopword removal, the impact of this preprocessing step to specific emotional classes could be worth investigating. Even though adjustments regarding activities in the preprocessing phase of text analysis are likely to raise performance measures, more potential could be identified in relation to text representation. Due to the rise in performance caused by the introduction of bigrams, this may be an indication that n-gram representation with $n > 1$ can further boost performance. In this regard, inclusion of n-grams with $n > 2$ into the feature space may possibly lead to better classification performance. As results in subsection 5.3.6 have shown, trigrams such as "looking forward to" and "thank you for" convey unused discrimination power, but are separated into derivations of uni- and bigrams. Consequently, an option for further exploration would be to not only use increased n-values, but also to completely neglect n-grams with low n-values such as unigrams. Another aspect to consider with respect to future work is feature selection. In particular, the number of features selected for the best results of each experiment instance turned out to vary unpredictably. Thus, special care should be taken when selecting features for certain scenarios, i.e., size and type of features need to be tailored to given prerequisites and the applied learning method. This is surely nothing new, but more extensive approaches than the one employed - namely selecting predefined numbers of features ranked by IG - were not applicable in the scope of this work. Also related to document representation, the poor performance achieved by POS adjusted datasets is explained by the removal of all tokens except nouns, verbs and adjectives (see subsection 5.3.6). In turn, tagging the original dataset without removing any words - especially in combination with stemming - is perhaps more promising than the chosen approach in the present thesis.
During the phase of knowledge finding, involving the application of the selected learning methods, configuration of WEKA's algorithm implementations was a crucial task with respect to performance optimization. Except NBM, each method provides options that should ideally influence classifier performance in a positive way. Finding the optimal combination of parameters is to some extent similar to a local search. Therefore, the results obtained by the present experiments might be local optima rather than global ones. In particular, numerical parameters for J48 and IBk require time-consuming search routines, which were not possible to complete in many times given the time constraints of the present work. In this respect, tool-support such as the built-in parameter optimization component of WEKA[1], could help to decrease the corresponding effort, at least to some extent.

The approaches to improve performance of emotional classification mentioned above are quite minor variations of what has been done in the experiments of the present thesis. However, there are other methodologies that differ more drastically from the approach implemented in this work. As such, a performance boost could potentially be achieved by the inclusion of semantic enrich-

[1] http://weka.wikispaces.com/Optimizing+parameters, 22.03.2015

ment of documents and words. Bloehdorn and Hotho [10], and Chaffar and Inkpen [14] conducted experiments in this domain, applying the WORDNET database. The former paper deals with extensive activities concerning document representation, e.g., dealing with synonymous and polysemous words and exploiting ontologies like WORDNET generalizations of concepts found in documents. Employing such methods were found to significantly improve results compared to simple term stem representation and are likely to increase classifier performance in the domain of emotion detection in text-based negotiation as well.
For the experiments of this thesis, each document in the training dataset was labeled with one of five predefined classes (see subsection 3.2.6). According to Sidorov et al., performance - measured in precision - decreases as the number of classes increases [78]. Therefore, class declaration could be done in a way that results in a lower number of classes. One possible way could be to implement a step-wise approach similar to Ghazi et al. [28]: the first step classifies documents with respect to emotional neutrality (neutral versus non-neutral), while further steps determine association in terms of the valence (displeasure vs. pleasure) and the activation (deactivation versus activation) of non-neutral document instances. In the most granular form, such an approach results in a series of binary classification tasks, which tend to reach higher accuracy values than single classifications into five classes. Based on these considerations, multi-label classification is another possible option, in which classifiers do not take only one label into account, but could, for instance, perform classification with respect to three binary labels: neutrality, valence and activation[2].
Compared to similar research in the field of sentiment analysis, classification performance achieved in the scope of this work is rather poor. This may be caused in part by the rather moderate number of document instances in the training set (730), but also has something to do with the characteristics of documents. Many experiments in corresponding research rely on Twitter data [30] [78], news headlines [54] or sentences extracted from blog posts [28] [14]. Thus, the size of each document measured in character count is limited and rather low, and the variety of decisive words used in one document is not expected to fluctuate immensely. Contrarily, a negotiation message usually consists of a collection of sentences, which could vary widely with respect to emotional load. For instance, a negotiator may start a message in a polite way, being thankful for previous offers and messages, but then switches to a more aggressive style. Clearly, such ambiguous messages raise the chances for classifiers to assign messages to the wrong class. This, in turn, suggests approaches of breaking down negotiation messages into smaller chunks (e.g. by dividing into sentences or predefined fractions) before applying learning methods and afterwards aggregating results accordingly.
Finally, in the context of knowledge discovery it may be worth conducting investigations regarding further learning methods in addition to the four algorithms applied in the scope of this work. Besides the option to explore the applicability of completely different data mining disciplines like association rule learning and clustering, supervised learning approaches can be further examined. Particularly, two classification approaches are promising candidates as they were already utilized in text mining scenarios with promising results. On the one hand, there is the relatively new technique of boosting, which relies on the concept of composite classifiers and is represented, for example, by AdaBoost [71] [10]. On the other hand, Maximum En-

[2] A corresponding extension to the WEKA toolkit is available under http://meka.sourceforge.net, 23.03.2015

tropy (MaxEnt) classifiers turned out to perform well in the text classification context and can even compete with Naive Bayes and SVM approaches in certain scenarios [62] [30].

List of figures

2.1 Emotion measurement approaches 10

3.1 Text analysis framework 21
3.2 Simple decision tree based on two numerical features and two classes 29
3.3 Linear hyperplane separating class instances 30
3.4 kNN classification with $k = 3$ 31
3.5 Class distribution from a radius perspective 32
3.6 Class encoding of valence and activation 33
3.7 WEKA's preprocess section 36
3.8 WEKA's filter configuration (TextToWordFilter) 38
3.9 WEKA's Classify section 39
3.10 WEKA's Experimenter application 39
3.11 POS tagger from the Stanford NLP Group 40

4.1 DOC.COM framework 48
4.2 Negoisst sample screenshot 49

5.1 Affective space and emotional poles 52
5.2 Representation of messages on two-dimensional space 53
5.3 Distribution of messages over valence 55
5.4 Distribution of messages over activation 56
5.5 Performance measures of Naive Bayes approach across experiment settings 59
5.6 Performance measures of decision tree approach across experiment settings 61
5.7 Performance measures of SVM approach across experiment settings 62
5.8 Performance measures of IBk approach across experiment settings 64
5.9 Comparison of accuracy of the four examined learning methods 68

List of tables

3.1 Spreadsheet of words in documents 21
3.2 Selection of stopword lists 22
3.3 List of combined experiment settings 35
3.4 Whitelist of tags for POS related experiments 41

5.1 Statistical key figures for valence 54
5.2 Summary of t-test of means of valence 54
5.3 Statistical key figures for activation 56
5.4 Summary of t-test of means of activation 57
5.5 Number of attributes attributes for each experiment setting 57
5.6 Overview of pair-wise comparison of experiment instances 58
5.7 Top three results from accuracy perspective, achieved by the application of NBM . 60
5.8 Top three results from accuracy perspective, achieved by application of J48 61
5.9 Top three results from accuracy perspective, achieved by application of SMO . . . 63
5.10 Top three results from accuracy perspective, achieved by application of IBk 64
5.11 Top ten and ten additionally selected features and their IG value 65
5.12 Words appearing in both the top 50 stopwords and the top ten features list of #01 . 66
5.13 Best results achieved by each learning method, selected by accuracy 67
5.14 Impact of preprocessing and representation aspects by pair-wise comparison 69
5.15 Class specific performance measures of the top results achieved by NBM and SMO 69
5.16 Class specific performance measures of the top results achieved by J48 and IBk . . 70
5.17 Class specific performance of #12.SMO with adequate F-score for class DD 71

Bibliography

[1] Hervé Abdi. *Encyclopedia of Measurement and Statistics*, chapter Kendall Rank Correlation, pages 509–511. SAGE Publications, Inc., 0 edition, 2007.

[2] Mohammed N. Al-Kabi Abdullah H Wahbeh, Qasem A. Al-Radaideh and Emad M. Al-Shawakfa. A comparison study between data mining tools over some classification methods. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Artificial Intelligence*, 1:18–26, 2011.

[3] Wendi L. Adair and Jeanne M. Brett. The negotiation dance: Time, culture, and behavioral sequences in negotiation. *Organization Science*, 16(1):33–51, 2005.

[4] Charu C. Aggarwal and Cheng Xiang Zhai. *An Introduction to Text Mining*. Springer US, 2012.

[5] Ameeta Agrawal and Aijun An. Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 346–353, Washington, DC, USA, 2012. IEEE Computer Society.

[6] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 579–586, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[7] Yas A. Alsultanny. Comparison between data mining algorithms implementation. In Hocine Cherifi, JasniMohamad Zain, and Eyas El-Qawasmeh, editors, *Digital Information and Communication Technology and Its Applications*, volume 167 of *Communications in Computer and Information Science*, pages 629–641. Springer Berlin Heidelberg, 2011.

[8] Shilpa Arora, Elijah Mayfield, Carolyn Penstein-Rosé, and Eric Nyberg. Sentiment classification using automatically extracted subgraph features. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 131–139, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[9] L.F. Barrett. Feelings or words? understanding the content in self-report ratings of experienced emotion. *Journal of Personality and Social Psychology*, 87(2):266–281, August 2004.

[10] Stephan Bloehdorn and Andreas Hotho. Boosting for text classification with semantic features. In Bamshad Mobasher, Olfa Nasraoui, Bing Liu, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis*, volume 3932 of *Lecture Notes in Computer Science*, pages 149–166. Springer Berlin Heidelberg, 2006.

[11] I. Borg and P. Groenen. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[12] Max Bramer. Text mining. In *Principles of Data Mining*, Undergraduate Topics in Computer Science, pages 329–343. Springer London, 2013.

[13] A. Carlson, C. Cumby, J. Rosen, and D. Roth. The snow learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, 5 1999.

[14] Soumaya Chaffar and Diana Inkpen. Using a heterogeneous dataset for emotion analysis in text. In Cory Butz and Pawan Lingras, editors, *Advances in Artificial Intelligence*, volume 6657 of *Lecture Notes in Computer Science*, pages 62–67. Springer Berlin Heidelberg, 2011.

[15] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990.

[16] Roddy Cowie, Naomi Sussman, and Aaron Ben-Ze'ev. Emotion: Concepts and definitions. In Roddy Cowie, Catherine Pelachaud, and Paolo Petta, editors, *Emotion-Oriented Systems*, Cognitive Technologies, pages 9–30. Springer Berlin Heidelberg, 2011.

[17] Alexander Dannenmann and Mareike Schoop, editors. *Conflict Management Support in Electronic Negotiations*, number Paper 31, 2010.

[18] Ricardo Sánchez de Madariaga, José Raúl Fernández del Castillo, and José Ramón Hilera. A generalization of the method for evaluation of stemming algorithms based on error counting. In Mariano Consens and Gonzalo Navarro, editors, *String Processing and Information Retrieval*, volume 3772 of *Lecture Notes in Computer Science*, pages 228–233. Springer Berlin Heidelberg, 2005.

[19] Ann Devitt and Khurshid Ahmad. Is there a language of sentiment? an analysis of lexical resources for sentiment analysis. *Language Resources and Evaluation*, 47(2):475–511, 2013.

[20] Yadolah Dodge. Bayes' theorem. In *The Concise Encyclopedia of Statistics*, pages 30–31. Springer New York, 2008.

[21] Daniel Druckman and Mara Olekalns. Emotions in negotiation. *Group Decision and Negotiation*, 17(1):1–11, 2008.

[22] Paul Ekman. *Basic Emotions*, pages 45–60. John Wiley Sons, Ltd, 2005.

[23] Murhaf Fares, Stephan Oepen, and Yi Zhang. Machine learning for high-quality tokenization replicating variable tokenization schemes. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 7816 of *Lecture Notes in Computer Science*, pages 231–244. Springer Berlin Heidelberg, 2013.

[24] Ingo Feinerer. Analysis and algorithms for stemming inversion. In Pu-Jen Cheng, Min-Yen Kan, Wai Lam, and Preslav Nakov, editors, *Information Retrieval Technology*, volume 6458 of *Lecture Notes in Computer Science*, pages 290–299. Springer Berlin Heidelberg, 2010.

[25] C. Fellbaum. *WordNet. An Electronic Lexical Database.* The MIT Press, 1998.

[26] Dan Flickinger. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6:15–28, 3 2000.

[27] Raymond A. Friedman and Steven C. Currall. Conflict escalation: Dispute exacerbating elements of e-mail communication. *Human Relations*, 56(11):1325–1347, 2003.

[28] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. Hierarchical approach to emotion recognition and classification in texts. In *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence*, AI'10, pages 40–50, Berlin, Heidelberg, 2010. Springer-Verlag.

[29] Alastair J. Gill, Darren Gergle, Robert M. French, and Jon Oberlander. Emotion rating from short blog texts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1121–1124, New York, NY, USA, 2008. ACM.

[30] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[31] Michele Griessmair and Sabine T. Koeszegi. Exploring the cognitive-emotional fugue in electronic negotiations. *Group Decision and Negotiation*, 18(3):213–234, 2009.

[32] Patrick Groenen and Michel van de Velden. Multidimensional scaling. Report / econometric institute, erasmus university rotterdam, E, http://hdl.handle.net/1765/1274, May 2004.

[33] G.K. Gupta. *Introduction to Data Mining with Case Studies*. Prentice-Hall of India Private Limited, 2006.

[34] Margaret A. Hafer and Stephen F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11–12):371 – 385, 1974.

[35] Eui-Hong(Sam) Han, George Karypis, and Vipin Kumar. Text categorization using weight adjusted k-nearest neighbor classification. In David Cheung, GrahamJ. Williams, and Qing Li, editors, *Advances in Knowledge Discovery and Data Mining*, volume 2035 of *Lecture Notes in Computer Science*, pages 53–65. Springer Berlin Heidelberg, 2001.

[36] Jeffrey T. Hancock, Christopher Landrigan, and Courtney Silver. Expressing emotion in text-based communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 929–932, New York, NY, USA, 2007. ACM.

[37] Michael J. Hine, Steven A. Murphy, Michael Weber, and Gregory Kersten. The role of emotion and language in dyadic e-negotiations. *Group Decision and Negotiation*, 18(3):193–211, 2009.

[38] Patrick Hippmann. *Multi-Level Dynamics of Affective Behaviors in Text- Based Online Negotiations: Impacts on Negotiation Success and Impacts of Decision Support*. PhD thesis, University of Vienna, 2014.

[39] Carroll E. Izard. *The face of emotion / Carroll E. Izard*. Appleton-Century-Crofts New York, 1971.

[40] Reza Karimpour, Amineh Ghorbani, Azadeh Pishdad, Mitra Mohtarami, Abolfazl AleAhmad, Hadi Amiri, and Farhad Oroumchian. Improving persian information retrieval systems using stemming and part of speech tagging. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, GarethJ.F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 89–96. Springer Berlin Heidelberg, 2009.

[41] Gregory E Kersten. E-negotiation systems: Interaction of people and technologies to resolve conflicts. In *UNESCAP Third Annual Forum on Online Dispute Resolution, Melbourne, Australia*, 2004.

[42] Gregory E. Kersten and Hsiangchu Lai. Negotiation support and e-negotiation systems: An overview. *Group Decision and Negotiation*, 16(6):553–586, 2007.

[43] Sabine T. Koeszegi, Eva-Maria Pesendorfer, and Rudolf Vetschera. Data-driven phase analysis of e-negotiations: An exemplary study of synchronous and asynchronous negotiations. *Group Decision and Negotiation*, 20(4):385–410, 2011.

[44] Sabine T. Koeszegi, Katharina J. Srnka, and Eva-Maria Pesendorfer. Electronic negotiations - a comparison of different support systems. *Die Betriebswirtschaft*, 66(4):441–463, 2006.

[45] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[46] Richard S. Lazarus, Klaus R. (Ed) Scherer, Angela (Ed) Schorr, and Tom (Ed) Johnstone. Relational meaning and discrete emotions. *Appraisal processes in emotion: Theory, methods, research. Series in affective science.*, pages 37–67, 2001.

[47] Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 45–53, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[48] Johannes Leveling. On the effect of stopword removal for sms-based faq retrieval. In Gosse Bouma, Ashwin Ittoo, Elisabeth Métais, and Hans Wortmann, editors, *Natural Language Processing and Information Systems*, volume 7337 of *Lecture Notes in Computer Science*, pages 128–139. Springer Berlin Heidelberg, 2012.

[49] Bing Liu. *Opinion Mining and Sentiment Analysis*. Data-Centric Systems and Applications. Springer Berlin Heidelberg, 2011.

[50] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[51] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[52] J.R. Méndez, E.L. Iglesias, F. Fdez-Riverola, F. Díaz, and J.M. Corchado. Tokenising, stemming and stopword removal on anti-spam filtering domain. In Roque Marín, Eva Onaindía, Alberto Bugarín, and José Santos, editors, *Current Topics in Artificial Intelligence*, volume 4177 of *Lecture Notes in Computer Science*, pages 449–458. Springer Berlin Heidelberg, 2006.

[53] Ronny Mitterhofer, Daniel Druckman, Michael Filzmoser, Johannes Gettinger, Mareike Schoop, and Sabine T. Koeszegi. Integration of behavioral and analytic decision support in electronic negotiations. In *HICSS'12*, pages 610–617, 2012.

[54] Saif Mohammad. Portable Features for Classifying Emotional Text, 2013.

[55] Pierre M. Nugues. Counting words. In *Language Processing with Perl and Prolog*, Cognitive Technologies, pages 123–167. Springer Berlin Heidelberg, 2014.

[56] Amer Obeidi, Keith W. Hipel, and D. Marc Kilgour. The role of emotions in envisioning outcomes in conflict analysis. *Group Decision and Negotiation*, 14(6):481–500, 2005.

[57] Akinori Okada and Yoshiko Takeuchi. Nonmetric multidimensional scaling of differences among type. *Psychonomic Science*, 25(4):197–198, 1971.

[58] Michael O'Mahony. *Sensory Evaluation of Food - Statistical Methods and Procedures*. Marcel Dekker Inc, Jänner 1986.

[59] Nazlia Omar, Mohammed Albared, Tareq Al-Moslmi, and Adel Al-Shabi. A comparative study of feature selection and machine learning algorithms for arabic sentiment classification. In Azizah Jaafar, Nazlena Mohamad Ali, ShahrulAzman Mohd Noah, AlanF.

Smeaton, Peter Bruza, ZainabAbu Bakar, Nursuriati Jamil, and TengkuMohdTengku Sembok, editors, *Information Retrieval Technology*, volume 8870 of *Lecture Notes in Computer Science*, pages 429–443. Springer International Publishing, 2014.

[60] Andrew Ortony and Terence J. Turner. What's basic about basic emotions? *Psychological Review*, 93(3):315–331, 1990.

[61] Chris D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, November 1990.

[62] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[63] Eva-Maria Pesendorfer, Andrea Graf, and Sabine T. Koeszegi. Relationship in electronic negotiations: Tracking behavior over time. *Zeitschrift für Betriebswirtschaft*, 77(12):1315–1338, 2007.

[64] Eva-Maria Pesendorfer and Sabine T. Koeszegi. Hot versus cool behavioural styles in electronic negotiations: The impact of communication mode. *Group Decision and Negotiation*, 15(2):141–155, 2006.

[65] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[66] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[67] James A Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[68] James A Russell and Lisa Feldman Barrett. Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. *Journal of personality and social psychology*, 76(5):805, 1999.

[69] David Sander, Didier Grandjean, and Klaus R. Scherer. A systems approach to appraisal mechanisms in emotion. *Neural Networks*, 18(4):317 – 352, 2005. Emotion and Brain.

[70] Beatrice Santorini. Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing). Technical report, Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA, 1990.

[71] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[72] KR Scherer, EB Roesch, and T Bänziger. Preliminary plans for exemplars: Theory humaine preliminary plans for exemplars: Theory humaine deliverable. Technical report, University of Geneva, 2004.

[73] H. Schlossberg. The description of facial expressions in terms of two dimensions. *Journal of Experimental Psychology*, 44:229–237, 1952.

[74] Mareike Schoop, Aida Jertila, and Thomas List. Negoisst: a negotiation support system for electronic business-to-business negotiations in e-commerce. *Data Knowledge Engineering*, 47(3):371 – 401, 2003. The language/action perspective.

[75] Mareike Schoop and Christoph Quix. Doc.com: A framework for effective negotiation support in electronic marketplaces. *Comput. Netw.*, 37(2):153–170, August 2001.

[76] John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*, volume 20. Cambridge University Press, 1969.

[77] Khaled Shaban, Otman Basir, and Mohamed Kamel. Document mining based on semantic understanding of text. In JoséFrancisco Martínez-Trinidad, JesúsAriel Carrasco Ochoa, and Josef Kittler, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, volume 4225 of *Lecture Notes in Computer Science*, pages 834–843. Springer Berlin Heidelberg, 2006.

[78] Grigori Sidorov, Sabino Miranda-Jiménez, Francisco Viveros-Jiménez, Alexander Gelbukh, Noé Castro-Sánchez, Francisco Velásquez, Ismael Díaz-Rangel, Sergio Suárez-Guerra, Alejandro Treviño, and Juan Gordon. Empirical study of machine learning based approach for opinion mining in tweets. In Ildar Batyrshin and Miguel González Mendoza, editors, *Advances in Artificial Intelligence*, volume 7629 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2013.

[79] Marina Sokolova and Guy Lapalme. How much do we say? using informativeness of negotiation text records for early prediction of negotiation outcomes. *Group Decision and Negotiation*, 21(3):363–379, 2012.

[80] Marina Sokolova and Stan Szpakowicz. Language patterns in the learning of strategies from negotiation texts. In Luc Lamontagne and Mario Marchand, editors, *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, pages 288–299. Springer Berlin Heidelberg, 2006.

[81] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, New York, NY, USA, 2008. ACM.

[82] Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[83] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, pages 252–259, 2003.

[84] F.S. von Thun. *Miteinander reden*. Miteinander reden / Friedemann Schulz von Thun. Rowohlt-Taschenbuch-Verlag, 2006.

[85] Richard E. Walton. Interpersonal peacemaking; confrontations and third-party consultation. Reading, Mass: Addison-Wesley, 1969.

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

[87] B. Weiner and S. Graham. *An attributional approach to emotional development. In Izard, Carroll E. and Kagan, Jerome and Zajonc, Robert B.: Emotions, cognition, and behavior*. Cambridge University Press, 1984.

[88] Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang. *Overview of Text Mining*. Texts in Computer Science. Springer London, 2010.

[89] Hwanjo Yu. Support vector machine. In LING LIU and M.TAMER ÖZSU, editors, *Encyclopedia of Database Systems*, pages 2890–2892. Springer US, 2009.

[90] Ding-Xuan Zhou and Kurt Jetter. Approximation with polynomial kernels and svm classifiers. *Advances in Computational Mathematics*, 25(1-3):323–344, 2006.