Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Mediensysteme

Sentimentanalyse von Tweets mit Methoden des SemEval 2013/2014

Bachelorarbeit

Michel Büchner
geb. am: 13.05.1988 in Jena

Matrikelnummer 80030

1. Gutachter: Prof. Dr. Matthias Hagen
2. Gutachter: Dr. Andreas Jakoby

Datum der Abgabe: 24. November 2014

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weimar, 24. November 2014

.....
Michel Büchner

Zusammenfassung

Wir beschäftigten uns in dieser Arbeit mit der Sentimentanalyse eines gesamten Tweets, wir untersuchten ob die Stimmung eines Tweets positiv, neutral oder negativ ist. Dazu implementierten wir vier Sentimentanalysesysteme nach und kombinierten die Ergebnisse der einzelnen Systeme, in dem wir den Durchschnitt der Klassenwahrscheinlichkeiten der einzelnen Systeme bildeten. Der Durchschnitt entschied dann, welche Stimmung einem Tweet zugewiesen wurde. Die vier verwendeten Sentimentanalysesysteme wählten wir aus Teilnehmern des SemEval 2013 und 2014 aus, die sich mit dem gleichen Problem der Sentimentanalyse eines Tweets beschäftigten. Unser System evaluierten wir mit den Daten des SemEval 2013 und 2014. In der Evaluierung zeigte sich, das sich unser System unter den besten 5 von 50 Teilnehmern des SemEval 2014 platzieren würde.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Diese Arbeit beschäftigt sich mit der Sentimentanalyse von Tweets. Die Sentimentanalyse ist eine Methode zur automatischen Klassifizierung eines Textes. Dabei soll die Stimmung des Textes, positiv oder negativ, bestimmt werden. So lässt sich mit Hilfe der Sentimentanalyse z. B. bestimmen, ob sich ein Kunde positiv oder negativ zu einem Produkt geäußert hat.

Ziel dieser Arbeit ist es, Verfahren zur Sentimentanalyse, die von Teilnehmern des SemEval 2013 und 2014 erstellt wurden, zu untersuchen und nachzubauen. Mit Hilfe der nachimplementieren Verfahren soll ein eigenes System entstehen, das wenn möglich eine höhere Genauigkeit aufweist als die Verfahren der Teilnehmer.

Die Sentimentanalyse wird immer wichtiger da in der heutigen Zeit viele Menschen ihre Meinung online mit anderen teilen. Zum Beispiel schreiben viele Reviews zu Büchern oder Filmen, teilen Ihre politischen Ansichten zu bestimmten Themen mit oder geben Feedback zu Restaurants oder Produkten. Diese Meinungen sind besonders wichtig für Filmstudios, Parteien oder Hersteller. Zum Beispiel ist es für Hersteller wichtig, Kundenfeedback zu erhalten um neue Produkte zu erfinden oder alte zu verbessern. Parteien hingegen würden gern wissen, wie die meisten Leute zu bestimmten Themen stehen, um im Wahlkampf besonders auf diese Themen eingehen zu können oder diese zu vermeiden. Doch wenden sich die Verbraucher meistens nicht direkt an den Hersteller, sondern lieber an andere Verbraucher, um darüber zu diskutieren. Solches Feedback oder politische Meinungen findet man häufig in Foren oder in sozialen Netzwerken, z. B. auf Twitter.

Mit monatlich 284 Millionen aktiven Nutzern (Twitter, 2014b) und 500 Millionen Tweets pro Tag (Twitter, 2014a) ist Twitter eines der größten sozialen Netzwerke. Twitter wird von Privatpersonen, Unternehmen und Organisationen als Kommunikationsplattform genutzt. Geteilt werden auf Twitter kurze

Textnachrichten, sogenannte Tweets. Diese Nachrichten umfassen viele Themen, wie politische Standpunkte, Meinungen zu Produkten oder Informationen und Ansichten zu aktuellen Geschehnissen. Diese riesige Menge an Nachrichten macht Twitter zu einer guten Quelle, um Meinungen zu finden.

Die Sentimentanalyse von Tweets ist allerdings eine größere Herausforderung, als die Analyse einer Produktkritik. Das liegt daran, dass Tweets relativ kurz sind (140 Zeichen), meist umgangssprachlich geschrieben sind und Ironie, genrespezifische Ausdrücke oder Abkürzungen enthalten können. Um effektive Verfahren zu finden, die mit den spezifischen Anforderungen eines Tweets gerecht werden, nahmen 2013 38 Teilnehmer an einem Wettbewerb der "Conference on Semantic Evaluation Exercises" (SemEval) teil. Aufgabe war es ein Verfahren zu entwickeln, welches bestimmt ob ein gegebener Tweet positiv, neutral oder negativ ist. Die gleiche Aufgabe wurde 2014 erneut gestellt, 2014 nahmen 50 Teilnehmer teil.

Wir beschäftigten uns in dieser Arbeit mit der Aufgabe des SemEval 2013 und 2014. Dazu bauten wir vier Systeme der Teilnehmer nach und kombinierten diese miteinander. So erstellten wir ein Sentimentanalysesystem, dessen Genauigkeit im Bereich der fünf besten Systeme des SemEval 2013 und 2014 liegt. Unser erstelltes Sentimentanalysesystem übertraf sogar die Genauigkeit des besten Systems aus 2013.

Die Arbeit ist wie folgt aufgebaut, im Kapitel 2 wird ein Blick auf verwandte Arbeiten der Sentimentanalyse geworfen. Wir betrachten dabei aktuelle Methoden, besonders im Bereich Twitter. Im Kapitel 3 werden die Systeme der ausgewählten Teilnehmer genauer erläutert. Wie wir die Systeme der Teilnehmer nachimplementiert und kombiniert haben, wird im Kapitel 4 beschrieben. Eine Evaluierung unserer Ergebnisse erfolgt im Kapitel 5. Einen Ausblick auf zukünftige Arbeiten und eine Zusammenfassung bilden das Kapitel 6.

Kapitel 2

Verwandte Arbeiten

Die Sentimentanalyse ist ein klassisches Textklassifizierungsproblem. Bei der Sentimentanalyse versucht man allerdings nicht Themengebieten oder Autoren zu bestimmen, sondern versucht die Stimmung des Textes zu bestimmen, z. B. ob diese positiv, neutral oder negativ ist. In der Sentimentanalyse werden deshalb oft Methoden verwendet, die bereits in der Textklassifizierung eingesetzt werden. Methoden, die häufig verwendet werden, stammen vorwiegend aus den Bereichen des maschinellen Lernens, der Computerlinguistik und der Statistik. Zur Analyse eines Textes werden oft mehrere dieser Methoden kombiniert (Feldman, 2013; Pang et al., 2002; Turney, 2002).

Computerlinguistik und Statistik werden hauptsächlich für die Vorverarbeitung oder zur Erstellung von sogenannten Features eingesetzt. Die Vorverarbeitung dient dazu, die Texte die man untersuchen möchte für eine Analyse vorzubereiten. Dazu können z. B. Wörter herausgefiltert werden, die nicht wichtig für die Analyse sind oder es kann eine Rechtschreibkorrektur erfolgen. Features sind Eigenschaften des Textes, z. B. wieviele positive Wörter der Text enthält, welche Wörter im Text vorkommen oder welche Emoticons verwendet werden. Eine Zusammenfassung von oft verwendeten Vorverarbeitungsschritten und Features folgt im Kapitel 3.

Maschinelles Lernen versucht, mithilfe der Features, bestimmte Gesetzmäßigkeiten in den Texten zu erkennen und zu erlernen. Diese Gesetzmäßigkeiten werden dann angewendet, um neue Texte einordnen (klassifizieren) zu können, wie z. B. in der Sentimentanalyse in positive und negative Texte. Dabei kommen oft zwei verschiedene Ansätze zum Einsatz, überwachtes Lernen oder unüberwachtes Lernen. Der Unterschied zwischen überwachten Lernen und unüberwachten Lernen besteht hauptsächlich darin, das bei überwachtem Lernen dem System sogenannte Trainingsdaten zur Verfügung stehen, dem unüberwachtem System hingegen nicht. Trainingsdaten sind Texte, deren Stim-

mung bereits ermittelt wurde. Diese Stimmung kann manuell von Personen bestimmt worden sein, das heißt, eine Person betrachtet jeden Text und ermittelt die Stimmung des Textes. Die Stimmung kann aber auch automatisch erkannt worden sein, wenn Informationen vorliegen, die klar eine Stimmungsrichtung anzeigen. Filmkritiken z. B. können extra Wertungen enthalten, wie 0 bis 5 Sterne, diese kann man dazu nutzen, einem Text eine Stimmung automatisch zuzuweisen. Hier kann z. B. eine 0-Sternebewertung auf eine negative Stimmung hinweisen. Die Stimmungen die den Texten zugewiesen wird, nennt man auch Klassen, bei der Sentimentanalyse können die Klassen z. B. positiv, neutral und negativ sein.

2.1 Unüberwachtes Lernen

Ein Sentimentanalysesysteme das unüberwachtes Lernen nutzt, ist z. B. das System von Turney (Turney, 2002). Das System verwendete keine Trainingsdaten, da es ein unüberwachtes System ist. Das System verwendete nur die „Pointwise mutual information“ (PMI) von Wortgruppen, die ein bestimmtes Wortartmuster erfüllten. PMI ist ein Maß für die statistische Abhängigkeit zweier Terme. So wurden für den ersten Term zunächst bestimmte Wortgruppen aus dem zu analysierenden Text extrahiert, wenn sie ein bestimmtes Muster erfüllten. Die Wortgruppe “langer Film” erfüllt z. B. eines dieser Muster, da nach einem Adjektiv ein Substantiv folgt. Als zweiten Term für die Berechnung des PMI verwendete das System die Worte “schlecht” und “ausgezeichnet”, da diese häufig in Filmkritiken zur Bewertung des Filmes eingesetzt werden und das System hauptsächlich solche Kritiken analysiert. Der PMI wird dann berechnet, indem die Wahrscheinlichkeit das beide Terme zusammen vorkommen und die Wahrscheinlichkeiten das die Terme alleine vorkommen bestimmt werden und mit der Formel 2.1 berechnet werden.

$$PMI(\text{“langer Film”},\text{“schlecht”}) = log_2 \left(\frac{p(\text{“langer Film” \& “schlecht”})}{p(\text{“langer Film”}) * p(\text{“schlecht”})} \right) \tag{2.1}$$

Die Wahrscheinlichkeit, dass eine Wortgruppe vorkommt, wurde in diesem System bestimmt, indem man eine Suche bei der Suchmaschine AltaVista ausführte und den “NEAR” Operator verwendete. Dieser sucht nach zwei Wortgruppen die in einem maximalen Abstand von zehn Wörtern zueinander vorkommen. Die Anzahl der Suchergebnisse stellt so die Wahrscheinlichkeit dar, das beide Wortgruppen zusammen vorkommen. So kann die Wahrscheinlichkeit das beide Wortgruppe vorkommen wie in Formel 2.2 bestimmt werden.

$$p(\text{“langer Film” \& “schlecht”}) = \text{Suchergebnisse}(\text{“langer Film” NEAR “schlecht”}) \tag{2.2}$$

Die Stimmung einer Wortgruppe wird dann berechnet, indem der PMI der Wortgruppe mit jeweils den Worten “schlecht” und “ausgezeichnet” bestimmt werden und diese dann voneinander subtrahiert werden, wie in der Formel 2.3 zu sehen ist.

$$\begin{aligned} Sentiment(\text{“langer Film”}) = PMI(\text{“langer Film”}, \text{“ausgezeichnet”}) \\ -PMI(\text{“langer Film”}, \text{“schlecht”}) \end{aligned} \tag{2.3}$$

Letztendlich musste nur noch der Durchschnitt für den gesamten Text berechnet werden und man erhält die Stimmung des gesamten Textes. Mit diesem System erreichte Turney im Bereich Filmkritiken nur eine Genauigkeit von 65,8 % (Turney, 2002). Pang, Lee und Vaithyanathan erstellten im gleichen Jahr ein eigenes Sentimentanalysesystem und erreichten im gleichen Bereich eine Genauigkeit von 82,9 % (Pang et al., 2002), sie verwendeten allerdings überwachtes Lernen.

2.2 Überwachtes Lernen

Das Standardvorgehen bei überwachten Systemen beginnt zunächst mit dem Trainieren eines Klassifikators. Dazu werden zuerst die Texte der Trainingsdaten vorverarbeitet. Anschließend werden aus den vorverarbeiteten Texten der Trainingsdaten Features extrahiert und ein Klassifikator wird mit diesen Features und den jeweiligen Klassen der Trainingsdaten trainiert. Nachdem der Klassifikator trainiert wurde, werden die Texte untersucht, von denen wir die Stimmung analysieren wollen, diese werden auch Testdaten genannt. Testdaten sind meistens ähnliche Texte, wie die Trainingsdaten, von denen allerdings die Stimmung nicht bekannt ist, sondern erst bestimmt werden soll. Diese Testdaten werden auch wieder vorverarbeitet und es werden Features erstellt. Nun kann der trainierte Klassifikator die Texte der Testdaten mit Hilfe der Features klassifizieren, das heißt, er bestimmt die Stimmung der Texte.

Ein Klassifikator ist ein Algorithmus, der Texte anhand von bestimmten Eigenschaften (Features) in vordefinierte Klassen aufteilt. Dazu wird der Klassifikator mit klassifizierten Texten trainiert, um dann auf unklassifizierte Daten

angewandt zu werden. Oft verwendete Klassifikatoren sind
Support Vector Machine (SVM), Naive Bayes und Maximum Entropy. Ein SVM Klassifikator z. B. bildet die Trainingsdaten zunächst in einem Vektorraum als Vektoren ab. Der SVM versucht dann, in diesem Raum eine Hyperebene als Trennfläche einzufügen, die die Klassen (z. B. positiv und negativ) voneinander trennt. Die Vektoren, die dieser Ebene am nächsten liegen, nennt man Stützvektoren. Es wird versucht, den Abstand dieser Stützvektoren zur Ebene möglichst zu maximieren, um eine möglichst hohe Genauigkeit zu erhalten. Die Testdaten werden dann als Vektoren in den Vektorraum eingesetzt. Auf welcher Seite der Ebene sich ein Vektor befindet, bestimmt dann zu welcher Klasse der Text gehört (Burges, 1998).

Welchen Klassifikator man verwenden sollte, hängt von dem Einsatzgebiet, der Größe der Daten und den verwendeten Features ab. In ihrem Paper verglichen Pang, Lee und Vaithyanathan die Klassifikatoren SVM, Naive Bayes und Maximum Entropy (Pang et al., 2002). Sie stellten dabei fest, dass alle drei Klassifikatoren je nach verwendeten Features unterschiedliche Genauigkeiten erreichten. Der SVM Klassifikator erreichte aber insgesamt die höchste Genauigkeit.

2.3 Sentimentanalyse von Tweets

Derzeit gibt es viele aktuelle Sentimentanalysetechniken, die Stimmungen im Text erkennen können. Doch für welche Einsatzgebiete ist der Einsatz überhaupt sinnvoll (Karlgren et al., 2012)? Besonders bei der Betrachtung von Tweets stellt sich die Frage: Wieso ist eine Sentimentanalyse sinnvoll? Eine aktuelle Technik beschäftigt sich beispielsweise damit, die öffentliche Stimmung durch Tweets abzubilden (Bollen et al., 2009). Dazu bestimmten sie die tägliche öffentliche Stimmung und stellten einen Zusammenhang mit aktuellen Ereignissen her, z. B. dem Ölpreis oder Wahlen. Gerade zu Wahlzeiten findet man oft Meinungen und Stimmungen zu bestimmten Parteien bei Twitter, deshalb beschäftigte sich eine weitere Technik mit Tweets die während einer TV Debatte getweetet wurden (Diakopoulos and Shamma, 2010). Sie verknüpften die diskutierten Themen mit Stimmungen aus Tweets und konnten so z. B. kontroverse Themen erkennen, die Parteien helfen können, ihren Wahlkampf auf bestimmte Themen abzustimmen. Tweets liefern natürlich nicht nur Meinungen zu politischen Themen, sondern auch zu bestimmten Produkten oder Marken (Ghiassi et al., 2013; Lim and Buntine). Diese sind für Firmen extrem wichtig, um neue Produkte zu entwickeln, oder alte zu verbessern. Für Firmen ist auch wichtig, was ihre Mitarbeiter über sie denken und wie die Stimmung der Mitarbeiter ist, um gezielt auf Probleme eingehen zu können, wenn Mitar-

beiter nicht zufrieden sind (Moniz and de Jong, 2014). Allerdings setzt das eine gewisse Mitarbeiteranzahl und die Nutzung von Twitter durch die Mitarbeiter voraus, sonst erhält man nicht genügend Daten zum Auswerten.

Ein weiteres Problem bei Twitter sind besonders negative oder beleidigende Tweets, diese möchte man eventuell nicht sehen. Mit Hilfe der Sentimentanalyse wäre es möglich, diese besonders negativen Tweets gleich zu erkennen und auszublenden (Xiang et al., 2012). Ein großes Problem stellt allerdings derzeit noch die Sprache dar, die meisten Sentimentanalysesysteme können nur englische Texte analysieren, doch werden Tweets nicht nur in Englisch geschrieben, wodurch eine große Anzahl an Tweets gar nicht ausgewertet werden kann. Doch einige Techniken beschäftigen sich genau mit diesem Problem und übersetzen z. B. den untersuchten Text automatisch vom Chinesischen ins Englische (He, 2011). Außerhalb von Twitter ist es auch möglich, Produktkritiken auszuwerten. So beschäftigt sich eine Technik mit dem Erstellen einer Rangliste anhand von Produktkritiken. Dabei werden zunächst für den Nutzer wichtige Kategorien erkannt und anhand derer wird ein Ranking der Produkte erstellt (de Albornoz et al., 2011). Ein weiteres Einsatzgebiet für die Sentimentanalyse sind Ergebnisse von Suchmaschinen, so wäre es möglich, sich direkt vor den Suchergebnissen anzeigen zu lassen wie die Stimmung der Seite oder des Artikels zu dem gesuchten Thema ist (Demartini, 2011).

Wie man sieht, gibt es viele Einsatzgebiete für die Sentimentanalyse. Da die Einsatzgebiete sehr verscheiden sind, werden viele unterschiedliche Ansätze verwendet, um die Sentimentanalyse durchzuführen. Deshalb kommen auch viele verschiedene Features zum Einsatz, je nach Anwendungsgebiet sind unterschiedliche Features geeignet. Häufig verwendet Features sind Unigramme (Go et al., 2009; Kouloumpis et al., 2011), Emoticons und Großbuchstaben (Barbosa and Feng, 2010) sowie lang gezogene Wörter (Brody and Diakopoulos, 2011). Oder es werden ausgefallene Features wie Lautschrift (Ermakov and Ermakova, 2013) oder mehrsprachige automatische Übersetzung (Balahur and Turchi, 2013) verwendet.

2.4 SemEval

Mit dem Problem der Sentimentanalyse beschäftigte sich seit 2013 ein Teil des SemEval (Nakov et al., 2013; Rosenthal et al., 2014). 2013 war es die Aufgabe 2 und 2014 die Aufgabe 9. Das Ziel dieser Aufgaben war es, einen Tweet auf seine Stimmung hin zu untersuchen. Dadurch versuchten die Veranstalter, die Forschung in Richtung Sentimentanalyse für Mikroblogging zu fördern und sie versuchten, die besten Ansätze für dieses Problem zu finden. Dazu teilten sie die Aufgabe in zwei Unteraufgaben. In Aufgabe A sollte für einen gegebenen

Abschnitt in einem Tweet bestimmt werden, ob dieser positiv, neutral oder negativ ist. In Aufgabe B sollte für einen gesamten Tweet bestimmt werden, ob dieser positiv, negativ oder neutral ist. Wir beschäftigen uns in dieser Arbeit mit der Aufgabe B, also der Sentimentanalyse des gesamten Tweets. Die SemEval Aufgabe verwendete auch teilweise zusätzlich Texte als Quellen wie SMS oder Live Journal Texte. Wir haben uns allerdings auf Tweets spezialisiert und verwenden auch nur diese. Alle Teilnehmer, die an dieser Aufgabe des SemEval teilnahmen verwendeten unterschiedliche Ansätze. Unser Ziel ist es, vier dieser Systeme miteinander zu kombinieren, um die Stärken der einzelnen Systeme zu nutzen.

2.5 Kombinationstechniken

Das Kombinieren mehrerer Systeme zu einem ist eine oft verwendete Methode um schwache Systeme zu einem starken System zu vereinen
(Maclin and Opitz, 2011; Polikar, 2006; Rokach, 2010). Bei der Kombination der einzelnen Systeme gibt es mehrere Möglichkeiten. Das Bagging
(Breiman, 1996) trainiert die einzelnen Klassifikatoren mit jeweils zufälligen Teilen der Trainingsdaten. Wo hingegen das Boosting
(Freund and Schapire, 1996; Schapire, 1990) versucht, Klassifikatoren mit Teilen der Trainingsdaten zu trainieren, die von einem anderen Klassifikator falsch klassifiziert wurden. In unserem System teilen wir die Trainingsdaten allerdings nicht auf, sondern jeder Klassifikator trainiert mit allen Trainingsdaten.
Die meisten Methoden zum Kombinierung verwenden die Endaussage der Klassifikatoren, das heißt, ist der Tweet positiv, neutral oder negativ und berechnen dann einfach den Durchschnitt aus diesen (Asker and Maclin, 1997). Wir betrachten allerdings nicht die Aussage, sondern die Wahrscheinlichkeiten, mit denen ein Tweet einer bestimmten Klasse zugeordnet wird. So lassen sich z. B. Wahrscheinlichkeiten für jede Klasse unterschiedlich gewichten (Fung et al., 2006). Doch wir gewichteten jede Klasse gleich und berechnen den Durchschnitt der Wahrscheinlichkeiten für jede Klasse, bevor wir eine Endaussage treffen.

Kapitel 3

Vorgehen der SemEval-Teilnehmer

Für unser Ensemblesystem wählten wir insgesamt vier Teilnehmer des SemEval 2013 und 2014 aus. 2013 beteiligten sich 38 Teilnehmer und 2014 nahmen 51 Teilnehmer an der Unteraufgabe B, der Sentimentanalyse von Tweets teil. Bei der Auswahl der Systeme orientierten wir uns am jeweiligen Ranking der Teams. Als erstes System verwendeten wir das System vom Team NRC-Canada. Sie belegten 2013 den ersten Platz und verwenden eine große Anzahl an Features und Sentimentlexika. Als zweites System wählten wir das System vom Team GU-MLT-LT, da es 2013 den zweiten Platz belegte und uns weitere Features und ein weiteres Sentimentlexikon lieferte. Als drittes System verwendeten wir das System vom Team KLUE, da es den fünften Platz belegte und es uns auch weitere Features und ein weiteres Sentimentlexikon lieferte. Das drittplatzierte System verwendeten wir nicht, da es einen komplett anderen Ansatz wählte und die Klassifizierung anhand von vorher aufgestellten Regeln vornahm. Diese Regeln sind allerdings nicht öffentlich verfügbar, wodurch das Nachbauen des Systems nicht möglich war. Wir entschieden uns auch dagegen, das viertplatzierte System zu verwenden, da es kaum neue Features und Sentimentlexika nutzte und sehr dem ersten und zweiten System ähnelte. Als viertes System entschieden wir uns letztendlich für das System von TeamX, da es 2014 den ersten Platz belegte und ein weiteres Feature und eine große Anzahl an Sentimentlexika verwendete.
Die verwendeten Features, Vorverarbeitungsschritte und Sentimentlexika werden im Folgenden allgemein beschrieben.

3.1 Features

Features sind Eigenschaften von Texten, mit denen ein Text beschrieben werden kann. Sie werden verwendet um den Klassifikator zu trainieren und um einen Text zu klassifizieren. Es gibt viele verschiedene Features, die je nach Einsatzgebiet verwendet werden. Einige werden fast immer benutzt, andere fast gar nicht oder nur in bestimmten Gebieten. Die Wahl der Features kann das Ergebnis der Sentimentanalyse stark beeinflussen.

Wort N-Gramme Wort N-Gramme werden von allen Features am meisten verwendet. Sie kommen in fast jedem Sentimentanalysesystem vor und sind leicht zu bestimmen. Dabei werden alle Wörter oder Wortgruppen des zu analysierenden Textes als Feature genommen. Werden Wortgruppen verwendet, so können diese zusammenhängen, das heißt nur Worte die hintereinander im Text vorkommen. Wortgruppen können aber auch nicht zusammenhängen, das heißt, Wörter im Text können beliebig zu einer Wortgruppe kombiniert werden. Wie viele Wörter zu einer Wortgruppe zusammengefasst werden, hängt dabei vom N in N-Grammen ab, so werden z. B. bei Trigrammen (3-Gramme) drei Wörter benutzt. Wie die Wörter und Wortgruppen als Feature verwendet werden, ist von System zu System unterschiedlich, so kann das Vorkommen oder das Nichtvorkommen als Feature genommen werden. Das heißt, kommt z. B. das Wort “Hallo” in einem Text vor, so wird diesem Feature, “Hallo” ist dabei das Feature, eine 1 oder „true“ zugewiesen, kommt es nicht vor, eine 0 oder „false“. Eine weitere Möglichkeit ist, die Anzahl wie oft das Wort im Text vorkommt dem Feature zuzuweisen. Das heißt, kommt das Wort “Hallo” dreimal im Text vor so wird dem Feature eine 3 zugewiesen. So entsteht eine riesige Anzahl an Features, da jedes Wort des Textes ein Feature darstellen kann. Um diese Anzahl zu verringern und Mehrfacheinträge zu minimieren, ist eine gute Vorverarbeitung wichtig. Da so falsche Schreibweisen oder Groß- und Kleinschreibungen nicht als unterschiedliche Features verwendet werden.

Zeichen N-Gramme Ähnlich wie bei Wort N-Grammen, können auch nur Zeichen, anstelle von Wörtern, als Feature verwendet werden. Dabei werden nur die einzelnen Zeichen und nicht die ganzen Wörter verwendet. Auch hier können wieder unterschiedlich viele Stellen verwendet werden. Zusammenhängende Trigramme vom Wort “Hallo” sind z. B. “Hal”, “all” und “llo”. Dabei macht es allerdings keinen Sinn, Unigramme zu verwenden, da man sonst einfach das ganze Alphabet als Features verwendet. Man kann auch hier wieder das Vorkommen oder die Anzahl dem Feature zuweisen.

Sentimentlexika Eine weitere häufig verwendete Featuregruppe sind Sentimentlexika. Diese bestehen meist aus einer Liste von Wörtern, denen ein bestimmter Sentimentwert zugeordnet ist. Dieser Wert kann aus Zahlen, z. B. von -5 bis 5, bestehen oder aus Worten, z. B. "positiv" oder "negativ". Mithilfe dieser Sentimentlexika ist es dann möglich, verschiedene Features zu erstellen, z. B. kann man für alle Wörter im Text die zugehörigen Werte bestimmen und diese aufsummieren, diese Summe kann dann als Feature genommen werden. Welche Sentimentlexika häufig verwendet werden, wird im Abschnitt 3.3 genauer beschrieben. Durch eine gute Vorverarbeitung kann hier auch die Trefferanzahl der Wörter im Sentimentlexikon erhöht werden. Da Wörter wie "Hallllo" wahrscheinlich im Sentimentlexikon nicht gefunden werden, das Wort "Hallo" hingegen schon, obwohl es die gleiche Bedeutung hat und sich nur in der Schreibweise unterscheidet.

Cluster Viele Wörter kommen immer im gleichen Kontext vor, das heißt, vor und nach einem Wort stehen teilweise oftmals die gleichen Wörter wie bei einem ähnlichen Wort. Dem Wort "der" können z. B. Wörter wie "Hund", "Stein" oder "Stift" folgen, dem Wort "ein" können auch die Wörter "Hund", "Stein" oder "Stift" folgen. Das heißt, beide Wörter, "der" und "ein", kommen im gleichen Kontext vor. Deshalb ist es möglich, diese Wörter zusammenzufassen. Um solche Wörter zusammenfassen zu können verwendet man "Brown clustering" (Brown et al., 1992). Mithilfe dieses Algorithmus fasste Owoputi 216.856 Wörter, aus insgesamt 56.345.753 Tweets, in 1000 Clustern zusammen (Owoputi et al., 2013). Da die Wörter in den jeweiligen Clustern oft im gleichen Kontext vorkommen, lassen sich die IDs der Cluster gut als zusätzliches Feature verwenden.

Wortstamm Wie in der Vorverarbeitung können Wortstämme auch als Feature benutzt werden. Wortstämme können so entweder als Wort N-Gramme, falls die Rückführung auf die Wortstämme in der Vorverarbeitung erfolgt ist, oder als eigens Feature, wenn keine Vorverarbeitung auf Wortstämme erfolgt ist, genutzt werden.

Wortart Wie im Abschnitt 3.2 beschrieben, kann man für jedes Wort in einem Satz seine Wortart bestimmen. Ist ein Wort z. B. ein Substantiv oder ein Adjektiv. Es ist möglich, das Vorkommen oder Nichtvorkommen einer Wortart im Text, wie bei N-Grammen, als Feature zu verwenden oder die Anzahl, wie oft eine Wortart vorkommt, kann als Feature verwendet werden.

Großbuchstaben Werden Wörter in Texten nur in Großbuchstaben geschrieben, z. B. "HALLO", so haben sie meistens eine besondere Bedeutung. Im Web gilt ein so geschriebenes Wort auch als Schreien. Deshalb haben solche Wörter

eine größere Bedeutung als normal geschriebene. Die Anzahl solcher Wörter in einem Text stellt auch ein mögliches Feature dar.

Satzzeichen Wie Wörter in Großbuchstaben deuten auch Satzzeichen auf besondere Bedeutungen hin. Gerade dann, wenn diese mehrfach wiederholt werden z. B. "!!!" oder "???". Das Wiederholen dieser Satzzeichen soll meistens die Aussage des Satzes verstärken. Sätze mit solchen Satzzeichen könnten so die Sentimentaussage des Textes beeinflussen. Deshalb kann die Anzahl solcher wiederholten Satzzeichen ein weiteres Feature sein.

Hashtags Hashtags kommen in fast jedem Tweet vor und sollen auf bestimmte Themen oder auch Stimmungen hinweisen. Deshalb sind auch Hashtags wichtig für die Sentimentanalyse, z. B. kann "#happy" direkt auf eine positive Stimmung hinweisen. So kann z. B. die Anzahl und Art der Hashtags ein Feature darstellen.

Emoticons Im Web, besonders in Chats und Tweets, findet man oft Emoticons. Diese sollen eine bestimmte Stimmung ausdrücken und sind daher wichtig für die Sentimentanalyse. Da sie direkt auf die Sentimentaussage des Textes hinweisen können. Deshalb kann z. B. die Anzahl positiver oder negativer Emoticons ein Feature sein.

In die Länge gezogene Wörter Möchte man einem Wort mehr Ausdruck oder eine höhere Bedeutung verleihen. So kann man ein Wort in Großbuchstaben schreiben, wie oben beschrieben, oder man zieht es in die Länge, man verwendet also einen oder einige Buchstaben mehrfach hintereinander.
"Halllllo" würde man z. B. verwenden um auf sich aufmerksam zu machen. Deshalb kann auch die Anzahl solcher Wörter ein Feature darstellen.

Wortanzahl Ein weiteres mögliches Feature ist die Anzahl an Wörtern im Tweet. Durch die beschränkte Anzahl an Zeichen in Tweets sollte sich die Anzahl an Wörtern für Tweets nicht stark unterscheiden. Anders bei z. B. Reviews, die sehr unterschiedliche Zahlen an Wörtern haben können.

Wortbedeutung Viele Wörter können je nach Kontext unterschiedliche Bedeutungen haben. Zum Beispiel kann mit einem Blatt, ein Blatt an einem Baum gemeint sein oder ein Blatt Papier. Deshalb ist es für die Sentimentanalyse wichtig, welchen Sinn ein Wort hat. Mithilfe von
"Word Sense Disambiguation"-Systemen lassen sich so Wahrscheinlichkeiten bestimmen, mit denen ein Wort eine bestimmte Bedeutung hat. Als Feature können dann die einzelnen Wortbedeutungen mit ihren jeweiligen Gewichtungen verwendet werden.

Negierungen In einigen Sätzen in einem Text könne Negierungswörter vorkommen, diese dienen dazu, die Aussage eines Textes umzukehren. So kann man schnell erkennen, dass der Satz "Der Film ist nicht gut." eine negative Aussage hat. Durch das Wort "nicht" wird hier die positive Aussage des Satzes, durch das Wort "gut", umgekehrt. Solche Sätze zu erkennen, ist in der Sentimentanalyse besonders wichtig, da Sätze sonst falsch gewertet werden. Eine Möglichkeit, solche negativen Abschnitte zu finden beschreiben Pang, Lee und Vaithyanathan in ihrem Paper (Pang et al., 2002). So wird zunächst nach Negierungswörtern gesucht, wie z. B. "nicht". Hat man ein solches Wort gefunden, so wird nach dem nachfolgenden Satzzeichen gesucht. Alle Wörter die nun in diesem Bereich liegen, vom Negierungswort zum Satzzeichen, werden vom Negierungswort negiert. So lassen sich alle Wörter in diesem Bereich gesondert markieren, z. B. ist es möglich, ihnen ein Suffix anzuhängen, so wird aus "gut" "gut_NEG". Dieses Suffix kann im Sentimentanalysesystem Einfluss auf die Wort N-Gramme haben, sodass "gut_NEG" ein eigenes Wort darstellt. Außerdem kann man die Zahl solcher negierten Abschnitte als Feature verwenden.

Wie man sieht, gibt es viele verschiedene Features, die alle unterschiedlichen Einfluss auf das Sentimentanalysesystem haben können. Einige haben großen Einfluss wie, z. B. Sentimentlexika und einige haben nur geringen Einfluss auf das Gesamtsystem, wie z. B. großgeschriebene Wörter oder Satzzeichen. Den Einfluss einzelner Features kann man gut im Paper von NRC-Canada sehen (Mohammad and Turney, 2010).

3.2 Vorverarbeitung

Viele Texte, die in der Sentimentanalyse verwendet werden, können sich stark voneinander unterscheiden. Nicht nur im Inhaltlichen, sondern auch im Aufbau. So können Texte formell oder salopp geschrieben sein und können Slangwörter enthalten. Das macht eine einheitliche Analyse meist nicht so einfach. Außerdem können Texte bestimmte Wörter oder Zeichen enthalten, die für die Analyse nicht wichtig sind, sondern störend wären. Besonders Tweets können URLs oder Benutzernamen beinhalten, die oftmals nicht wichtig sind. Deshalb werden die Texte vorverarbeitet, bevor sie zur Analyse verwendet werden. Dabei gibt es viele unterschiedliche Möglichkeiten, die oft miteinander kombiniert werden.

Kleinschreibweise Um den Text zu vereinheitlichen, werden oft alle Zeichen in Kleinschreibweise konvertiert. Das ist für N-Gramm Features von Vorteil, da man Dopplungen vermeiden kann. Zum Beispiel wenn ein Wort am Satz-

anfang steht und ein Wort im Satzinneren, durch die unterschiedliche Groß- und Kleinschreibweise würden die Wörter so mehrmals als Feature verwendet, obwohl es das gleiche Wort ist. Allerdings können durch die Kleinschreibweise wichtige Informationen verloren gehen, die für einige Features benötigt werden, z. B. benötigt das Feature "Großbuchstaben" die Information, ob ein Buchstabe groß oder klein geschrieben wurde. Deshalb ist es manchmal sinnvoll, eine weitere Version des Textes aufzuheben, die nicht in Kleinschreibweise konvertiert wurde.

Ersetzen oder löschen unwichtiger Wörter Oft kommen in Texten Wörter oder Zeichen vor, die für die Sentimentanalyse keine Bedeutung haben, wie z. B. URLs. Da diese unwichtigen Wörter nicht benötigt werden, können sie ganz gelöscht oder durch einen Platzhalter ersetzt werden. So kann eine sogenannte Stopwortliste erstellt werden, mit Wörtern, die man nicht beachten will. So können z. B. Artikel wie "der", "die", "das" auf einer solchen Liste stehen. Auch Zahlen können unwichtig sein und ersetzt oder gelöscht werden.

Kürzen von Wörtern Wie auch schon bei der Konvertierung in Kleinschreibweise ist es sinnvoll, Wörter die in die Länge gezogen wurden z. B. "Hallllo" zu kürzen. Denn ohne Kürzen wären "Hallllo" und "Hallo" zwei unterschiedliche Wörter, was z. B. die Trefferquote bei der Verwendung von Sentimentlexika verringern kann. Allerdings werden solche Wörter auch als Feature verwendet, weshalb man bei der Nutzung von lang gezogenen Wörtern als Feature die Wörter nicht kürzen sollte.

Rechtschreibkorrektur In Texten kann es häufiger vorkommen, das man sich verschreibt, besonders bei Tweets. Da die Texte so kurz sind, liest man sie meist nicht noch einmal durch, um Fehler zu erkennen. Durch Rechtschreibfehler kann es allerdings wieder zu Mehrfacheinträgen oder Falscheinträgen bei der Featureerstellung kommen. Deshalb ist es eventuell sinnvoll, eine Rechtschreibkorrektur durchzuführen. Allerdings kann es dabei zu Problemen kommen, da die Korrektur automatisch erfolgt und es oft mehrere Möglichkeiten gibt, wie der Autor das Wort eigentlich schreiben wollte. Dadurch kann es zu Fehlern kommen, die den Sinn eines Wortes komplett ändern. Außerdem hat der Autor das Wort vielleicht mit Absicht falsch geschrieben. So können Wörter verloren gehen, die für einige Features wichtig wären, wie z. B. in die Länge gezogene Wörter.

Wortstamm Um weitere Mehrfachverwendungen von Wörtern zu vermeiden, lassen sich Wörter auch auf ihren Wortstamm zurückführen. So kann man unterschiedliche Zeitformen zusammenfassen. Zum Beispiel haben die Wörter "finden" und "gefunden" den gleichen Wortstamm "find". Allerdings kann es

dabei auch vorkommen, dass die Bedeutung des Wortes verloren geht. Zum Beispiel haben die Worte "auffinden" und "empfinden" eine unterschiedliche Bedeutung, allerdings den gleichen Wortstamm "find".

Wortartmarkierung In einem Satz kann jedes Wort eine unterschiedliche Bedeutung haben, z. B. kann ein Wort ein Substantiv oder Adjektiv sein. Diese Wortarten können für die Analyse ein wichtiger Faktor sein. Deshalb ist es manchmal wichtig, alle Wörter vor der Analyse mit ihrer Wortartmarkierung zu versehen. Diesen Vorgang nennt man auch Part-of-speech Tagging.

Tokenisierung Da man in der Sentimentanalyse teilweise mit einzelnen Wörtern oder Wortgruppen anstatt des ganzen Textes arbeitet, z. B. bei der Verwendung von N-Grammen, werden Texte vor der Analyse in sogenannte Token zerlegt, häufig werden die Wörter und Wortgruppen dabei an den Leerzeichen getrennt.

Wie man sieht, sollte man bei der Wahl der Vorverarbeitungsschritte beachten, welche Features man später erstellen möchte, da durch die Wahl der falschen Vorverarbeitungsschritte Informationen verloren gehen können die für einige Features wichtig sind.

3.3 Sentimentlexika

Sentimentlexika sind sehr wichtig für die Sentimentanalyse, da sie einzelnen Wörtern Sentimentwerte zuweisen und so wichtig für die Featureerstellung sind. Anhand der Sentimentwerte lässt sich ermitteln, ob ein Wort positiv oder negativ ist. Außerdem haben sie einen großen Einfluss auf das Gesamtergebnis der Sentimentanalyse, wie man im Paper von NRC-Canada sehen kann (Mohammad and Turney, 2010). Allerdings gibt es viele verschiedene Sentimentlexika, die sich alle stark unterscheiden können, z. B. im Wortumfang und in der Art der Wörter. Einträge in Sentimentlexika sind meistens Wörter oder Wortgruppen.

Bing Liu's Opinion Lexicon Bing Liu's Opinion Lexicon
(Hu and Liu, 2004) enthält ca. 6800 Wörter. Es wurde hauptsächlich mithilfe von Texten aus Produkt- und Filmreviews erstellt. Es enthält auch falsch geschriebene und Slangwörter.

MPQA Subjectivity Lexicon Das MPQA Subjectivity Lexicon
(Wilson et al., 2005) enthält ca. 8200 Einträge aus dem Bereich Nachrichten. Es enthält hauptsächlich formale Einträge, das heißt, es enthält z. B. keine falsch geschriebenen Wörter.

AFINN-111 Das AFINN-111 Lexikon (Nielsen, 2011) wurde manuell erstellt und enthält 2477 Einträge. Das Lexikon wurde speziell für die Sentimentanalyse von kurzen Texten in sozialen Medien erstellt.

General Inquirer Das General Inquirer Lexikon (Stone et al., 1966) enthält mehr als 11.000 Einträge. Es enthält auch hauptsächlich formale Einträge.

NRC Emotion Lexicon Das von NRC-Canada manuell erstellte Lexikon enthält ca. 14.000 Wörter (Mohammad and Turney, 2010).

NRC Hashtag Lexicon Das NRC Hashtag Lexicon wurde von NRC-Canada speziell für die SemEval 2013 Twitteraufgabe automatisch erstellt (Mohammad et al., 2013). Zur Erstellung suchte NRC von April bis Dezember 2012, nach Tweets mit positiven und negativen Hashtags. Dazu erstellten sie eine Liste aus 32 positiven und 38 negativen Wörtern, wie z. B. "#good", "#excellent", "#bad" und "#terrible". So erstellte NRC ein Set aus 775.000 Tweets. Für die einzelnen Wörter und Wortpaare der Tweets wurden dann Wertungen mithilfe des PMI berechnet. Positive Werte zeigen, dass ein Wort oder Wortpaar eine positive Aussage hat, negative das es eine negative hat. So wurde ein Lexikon mit 54.129 Unigrammen, 316.531 Bigrammen und 308.808 nicht zusammenhängenden Wortpaaren erstellt.

NRC Sentiment140 Lexicon Wie auch das NRC Hashtag Lexicon wurde auch dieses Lexikon von NRC-Canada speziell für die SemEval 2013 Twitteraufgabe automatisch erstellt (Mohammad et al., 2013). Das Lexikon wurde ähnlich erstellt wie das NRC Hashtag Lexicon, allerdings wurden hier nicht positive und negative Hashtags verwendet, sondern Emoticons. So wurde ein Lexikon mit 62.468 Unigrammen, 677.698 Bigrammen und 480.010 nicht zusammenhängenden Wortpaaren erstellt.

SentiWordNet Das SentiWordNet Lexikon (Baccianella and Sebastiani, 2010) erweitert das WordNet Lexikon um Sentimenteinträge. Das WordNet Lexikon enthält ca. 150.000 Einträge. Auch das SentiWordNet enthält hauptsächlich formale Einträge.

Wie man sieht, gibt es viele verschiedene Sentimentlexika, diese unterscheiden sich hauptsächlich in den Wortarten, die sie enthalten. Einige können direkt mit falsch geschrieben Wörtern verwendet werden, andere enthalten diese Wörter nicht, weshalb eine Vorverarbeitung von falschgeschrieben Wörtern erfolgen muss. Deshalb sind für die Sentimentanalyse von Tweets Sentimentlexika gut geeignet, die bereits falsch geschriebene Wörter und Slangwörter enthalten.

3.4 Teilnehmer

In Tabelle 3.1 wird dargestellt, welche Features, Vorverarbeitungsschritte und Sentimentlexika die einzelnen Teilnehmer verwendeten. Allerdings werden nicht alle Features so verwendet, wie im Abschnitt zuvor beschrieben. Die Besonderheiten der einzelnen Systeme werden im Folgenden genauer erläutert.

3.4.1 NRC-Canada

Das Team NRC-Canada (Mohammad et al., 2013) verwendete in ihrem System, wie in Tabelle 3.1 zu sehen, eine große Anzahl an verschiedenen Features und erstellten eigene Sentimentlexika. Die verwendeten Features nutzten sie, um einen linearen SVM-Klassifikator zu trainieren. Um nicht zu viele Informationen zu verlieren, setzten sie nur wenige Vorverarbeitungsschritte ein, wie in Tabelle 3.1 zu sehen. Sie ersetzten auch nur URLs und Benutzernamen durch Platzhalter. Die meisten Features setzten sie wie im Abschnitt zuvor beschrieben ein, Besonderheiten bei der Verwendung der Features sind die folgenden.

Wort und Zeichen N-Gramme Wie in der Tabelle 3.1 zu sehen, verwendete das Team NRC-Canada nicht nur Wort N-Gramme, sondern auch Zeichen N-Gramme. Für N-Gramme betrachtete NRC das Vorkommen oder Nichtvorkommen. NRC verwendeten für Wort N-Gramme zusammenhängende und nichtzusammenhängende 1- bis 4-Gramme. Für Zeichen verwendeten sie zusammenhängende 3- bis 5-Gramme.

Sentimentlexika Als Sentimentlexika verwendete NRC die Sentimentlexika, wie in Tabelle 3.1 zu sehen. Mithilfe dieser Sentimentlexika erstellte NRC vier Features, für jedes Sentimentlexikon und jeweils für positive und negative Wertungen. Die Features sind die Anzahl der positiven (negativen) Wörter des Tweets, die Summe aller positiven (negativen) Wertungen des Tweets, die größte positive (negative) Wertung und die Wertung des letzten positiven (negativen) Wortes des Tweets. So werden insgesamt 40 (5 * 2 * 4) Features erstellt.

Satzzeichen Für Satzzeichen wurde nicht nur die Anzahl wiederholter Satzzeichen als Feature verwendet, sondern auch, ob das letzte Zeichen ein Ausrufezeichen oder Fragezeichen ist.

Emoticons Für die Emoticon-Features wurde nicht die Anzahl der Emoticons als Feature verwendet, sondern das Vorkommen oder Nichtvorkommen von positiven und negativen Emoticons. Zusätzlich wurde noch bestimmt, ob das letzte Zeichen ein Emoticon ist.

3.4.2 GU-MLT-LT

Das Team GU-MLT-LT (Günther and Furrer, 2013) verwendete, wie in der Tabelle 3.1 zu sehen, mehrere Vorverarbeitungssysteme, allerdings nur wenige Features. Wie auch bei NRC-Canada wird ein linearer SVM-Klassifikator mit den Features trainiert. Anders als die anderen Teilnehmer verwendeten sie in ihren Features nicht immer den Tweet, der durch alle Vorverarbeitungsschritte gelaufen ist, sondern legten insgesamt drei Versionen an. Alle Versionen wurden allerdings zuvor in Token zerlegt. Version 1, der Roh-Tweet, nutzte gar keine Vorverarbeitungsschritte außer Tokenisierung. Bei der 2. Version, der normalisierten Version, wurden alle Wörter in Kleinschreibweise konvertiert und es wurden alle Zahlen durch 0 als Platzhalter ersetzt. In der 3. Version, die kollabierte Version, wurde die normalisierte Variante verwendet und dabei zusätzlich noch alle Wörter gekürzt, in denen Buchstaben mehr als zweimal hintereinander vorkamen. Die unterschiedlichen Versionen setzte GU-MLT-LT bei der Featureerstellung wie folgt ein.

Wort N-Gramme Für Wort N-Gramme verwendeten sie Unigramme des normalisierten Tweets.

Wortstamm Zur Erstellung der Wortstämme verwendete GU-MLT-LT die Token des kollabierten Tweets.

Cluster Die IDs der von Owoputi erstellten Cluster bestimmte GU-MLT-LT von allen drei Versionen und verwendeten jeweils das Vorkommen der IDs im Tweet als Feature.

Sentimentlexika Als Sentimentlexikon verwendete GU-MLT-LT nur das SentiWordNet. Zur Erstellung des Features nahm GU-MLT-LT die kollabierten Token des Tweets und berechneten die Summe aller Wertungen.

Negierung Negierung verwendete GU-MLT-LT nicht direkt als Feature, sondern wendete die Negierung auf die normalisierten Token und die Wortstämme an.

3.4.3 KLUE

Das Team KLUE (Proisl et al., 2013) verwendete von allen vier betrachteten Systemen die wenigsten Features, nutzte dabei allerdings ein Feature das die anderen Teilnehmer nicht verwenden. Als Klassifikator verwendete KLUE, nicht wie die anderen Teilnehmer einen linearen SVM-Klassifikator, sondern einen Maximum-Entropy-Klassifikator. KLUE reduzieren auch als einziges System alle Wörter bereits in der Vorverarbeitung auf ihren Wortstamm, wodurch

die Bedeutung einiger Wörter verloren gehen kann. Weitere Besonderheiten bei der Featureerstellung sind die folgenden.

Wort N-Gramme Anders als die anderen Teilnehmer verwendete KLUE nicht das Vorkommen oder Nichtvorkommen der N-Gramme, sondern wie häufig das N-Gramm im Tweet vorkommt. Da Tweets aber relativ kurz sind, kommt es nicht so häufig vor das ein N-Gramm pro Tweet mehr als einmal vorkommt. Damit ein N-Gramm allerdings überhaupt als Feature verwendet wird, muss es in mindestens fünf Tweets vorkommen. Verwendet werden Uni- und Bigramme.

Sentimentlexika Wie in Tabelle 3.1 zu sehen, verwendete KLUE das AFINN-111 Lexikon, allerdings erweiterten sie dieses um 343 Wörter. Diese zusätzlichen Wörter wurden bestimmt, indem KLUE ähnliche Wörter, wie die im AFINN-111 Lexikon enthaltenen, in Textkorpora von Google und Wikipedia suchten. Dazu suchten sie in den Textkorpora nach Wörtern die zusammen mit einem Wort des AFINN-111 Lexikons vorkommen und berechneten dementsprechende Wertungen. Letztendlich verwendeten sie nur Wörter deren Wertung größer als 2,5 oder kleiner als -2,5 war. Zusätzlich nutzte KLUE eine Liste mit 212 Emoticons und 95 Internetslangabkürzungen von Wikipedia und vergaben per Hand die Wertungen -1 (negativ), 0 (neutral) und 1 (positiv). Mithilfe dieser Sentimentlexika erstellten sie die folgenden vier Features für jedes Sentimentlexikon. Die Anzahl an positiven Wörtern im Tweet, die Anzahl an negativen Wörtern im Tweet, die Anzahl an Wörtern im Tweet, die im Sentimentlexikon vorkommen, und der Durchschnitt aller Wertungen.

Negierung Negierung wird nicht direkt als Feature verwendet, sondern alle Wertungen des Sentimentlexikafeatures, die im Negierungsbereich liegen werden mit -1 multipliziert, was die Aussage der Wertungen umkehrt. Positive Wörter werden so zu negativen und umgekehrt.

3.4.4 TeamX

TeamX (Miura et al., 2014) orientiert sich bei der Featurewahl am System von NRC-Canada, allerdings werden weniger Features als bei dem System von NRC-Canada verwendet, dafür werden allerdings die meisten Sentimentlexika genutzt. Wie auch bei NRC-Canada und GU-MLT-LT wird ein linearer SVM Klassifikator mit den Features trainiert. Allerdings erfolgt vor der endgültigen Klassifizierung eine Gewichtung der möglichen Klassen. Mehr zu der Gewichtung der Klassen im Abschnitt 4.2. Auch werden die verwendeten Sentimentlexika jeweils unterschiedlich behandelt, da unterschiedliche Part-of-speech Tagger verwendet werden. Die Besonderheiten bei der Behandlung der Sentiment-

lexika und der Features sind die folgenden.

Wortartmarkierung Bei der Vorverarbeitung werden zwei unterschiedliche Part-of-speech Tagger verwendet, um die Wortarten zu bestimmen. Der "Stanford POS Tagger" wird für formale Sentimentlexika und für das Wortsinnfeature verwendet. Der "CMU ARK POS Tagger" wird für formlose Sentimentlexika, N-Gramm und Cluster Features verwendet. Der "CMU ARK POS Tagger" wurde extra für Tweets entwickelt und kommt daher besser mit formloser Sprache zurecht, so können Abkürzungen oder Wörter die häufig bei Twitter verwendet werden besser erkannt werden.

Wort und Zeichen N-Gramme Wie bei NRC-Canada werden nicht nur Wort- sondern auch Zeichen N-Gramme verwendet. Auch werden zusammenhängende und nicht zusammenhängende 1- bis 4-Gramme für Wörter und zusammenhängende 3- bis 5-Gramme für Zeichen verwendet.

Sentimentlexika Die verwendeten Sentimentlexika werden wie in Tabelle 3.1 zu sehen in zwei Gruppen geteilt, in formal (x^F) und formlos (x^L). Die verwendeten Tweets, die mit diesen Sentimentlexika verwendet werden, unterscheiden sich darin, dass die Wortarten bei der Vorverarbeitung, von zwei unterschiedlichen Part-of-speech Taggern bestimmt wurden. Die zwei Lexikagruppen unterscheiden sich darin, das in den formalen Lexika nur Wörter vorkommen, die auch in Wörterbüchern zu finden sind. In den formlosen Lexika kommen hingegen auch Slangwörter, Abkürzungen oder falsch geschriebene Wörter vor. Mithilfe aller verwendeten Sentimentlexika, erstellten TeamX vier Features, für jedes Sentimentlexikon und jeweils für positive und negative Wertungen. Die Features sind die Anzahl der positiven (negativen) Wörter des Tweets, die Summe aller positiven (negativen) Wertungen des Tweets, die größte positive (negative) Wertung und die Wertung des letzten positiven (negativen) Wortes des Tweets. So werden insgesamt 40 (7 * 2 * 4) Features erstellt.

Features	**TeamX**	**NRC-Canada**	**GU-MLT-LT**	**KLUE**
Wort N-Gramme	x	x	x	x
Zeichen N-Gramme	x	x		
Lexika	x	x	x	x
Cluster	x	x	x	
Wortstamm			x	
Wortarten		x		
Großbuchstaben		x		
Satzzeichen		x		
Hashtags		x		
Emoticons		x		
Langgezogene Wörter		x		
Negierungen	x	x	x	x
Wortbedeutung	x			
Wortanzahl				x
Vorverarbeitung				
Kleinschreibweise	x		x	x
Unwichtige Worte	x	x	x	x
Kürzen von Wörtern			x	
Rechschreibkorrektur	x			
Wortstamm				x
Wortartmakierung	x	x		
Tokenisierung	x	x	x	x
Sentimentlexika				
Bing Liu's Opinion Lexicon	x^{L}	x		
MPQA Subjectivity Lexicon	x^{F}	x		
AFINN-111	x^{L}			x
General Inquirer	x^{F}			
NRC Emotion Lexicon		x		
NRC Hashtag Sentiment L.	x^{L}	x		
Sentiment140 Lexicon	x^{L}	x		
SentiWordNet	x^{F}		x	

Tabelle 3.1: Vergleich der verwendeten Features, Vorverarbeitungsschritte und Lexika.

Kapitel 4

Unser Vorgehen

Um unser Ensemblesystem zu erstellen, implementierten wir zunächst die Systeme von NRC-Canada, GU-MLT-LT, KLUE und TeamX nach. Danach kombinierten wir die Ergebnisse der einzelnen Systeme zu einem Endergebnis.

4.1 Implementierung

Wir versuchten die Systeme der Teilnehmer so genau wie möglich nachzuimplementieren. Allerdings war uns das nicht immer möglich, da einige Features in den Papern der Teilnehmer nicht genau beschrieben waren oder uns einige Daten nicht zur Verfügung stehen. Zum Beispiel konnten wir für das System von KLUE nicht das erweiterte AFINN-111 Lexikon verwenden, da dies nicht öffentlich verfügbar ist. Deshalb mussten wir das normale
AFINN-111 Lexikon verwenden. Auch stand uns nicht das Emoticon Lexikon von KLUE zur Verfügung, weshalb wir unser eigenes System erstellten, um positive und negative Emoticons zu erkennen. Dazu untersuchten wir, mit welchem Zeichen ein Emoticon endet, da das letzte Zeichen meist die Stimmung des Emoticons aufzeigt. Endet ein Emoticon z. B. auf ")", weist dies meist auf das Emoticon ":-)" hin, welches eine positive Stimmung ausdrückt. Wir untersuchten auch, mit welchem Zeichen das Emoticon beginnt, da Emoticons auch spiegelverkehrt geschrieben werden können, z. B. "(-:". So konnten wir erkennen ob ein Emoticon eine positive oder negative Stimmung ausdrückt und vergaben Sentimentwertungen, -1 für negative und 1 für positive Emoticons. Mit diesen Wertungen konnten wir die Features wie bei dem System von KLUE bestimmen.

Auch konnten wir den von TeamX verwendeten
"Word Sense Disambiguator" nicht nachimplementieren, da der verwendete "Word Sense Disambiguator" "UKB" in der Programmiersprache C++ geschrieben ist und wir Java verwendeten. Eine Alternative für Java wäre der

"Word Sense Disambiguator" "BabelNet", allerdings haben wir es nicht geschafft "BabelNet" in unser System zu integrieren. Deshalb haben wir uns dazu entschieden, dieses Feature im System vom TeamX nicht zu verwenden. TeamX nahm außerdem eine Gewichtung seiner klassifizierten Wahrscheinlichkeiten vor, wir verwendeten diese Gewichtungen allerdings nicht, da die in ihrem Paper angegebenen Gewichtungen in unseren Tests zu einer Verschlechterung des Systems geführt hätte. Des Weiteren wurden einige Features in den Papern der Teilnehmer nur kurz in einem Satz erklärt, was teilweise Platz für Interpretationen ließ. Wir versuchten trotzdem, jedes Feature so genau wie möglich nachzuimplementieren.

Bei der Erstellung der Features mussten wir genau darauf achten, welche Datenstrukturen wir in der Programmierung verwendeten. Durch die riesige Anzahl an Features kann es ansonsten schnell zu Speicher- und Laufzeitproblemen kommen. Bis auf das Wortsinnfeature von TeamX hatten wir keine Probleme bei der Implementierung der Features.

Bei der Wahl des Klassifikators entschieden wir uns, für alle Systeme einen linearen SVM Klassifikator zu verwenden, da alle Systeme, außer das System von KLUE, diesen bereits einsetzten. Außerdem zeigten unsere Tests, das unsere nachimplementierte Version des KLUE Systems eine höhere Genauigkeit mit einem SVM Klassifikator erreichte. Wir verwendeten als Klassifikator die Java Library "LIBLINEAR" (Fan et al., 2008) und für die Evaluierung und Klassifizierung die Library von "WEKA" (Hall et al., 2009). Für den linearen SVM Klassifikator wählten wir als SVM-Typ "L2-regularized logistic regression" und setzen den "Cost" Parameter für NRC auf 0,5, für GU-MLT-LT auf 0,15 und für KLUE und Team X auf 0,05. Den "Cost" Parameter bestimmten wir, indem wir die Systeme, in 0,05 Schritten, mit unterschiedlichen Werten testeten und uns diese Werte die höchste Genauigkeit lieferten. Wir verwendeten "LIBLINEAR", da dieser Klassifikator für Dokumentklassifizierung mit sehr vielen Features eine sehr hohe Genauigkeit bei kurzer Trainingszeit bietet. "WEKA" verwendeten wir, da es eine gute Möglichkeit bietet, die erstellten Features zwischenzuspeichern, zu klassifizieren und zu evaluieren. Wir wählten als SVM-Typ "L2-regularized logistic regression", da dieser Typ es uns als einziger ermöglichte, die Wahrscheinlichkeiten der einzelnen Klassen auszulesen, welche wir für die Kombination der einzelnen Systeme benötigen.

In Tabelle 4.1 sieht man den Vergleich der Originalsysteme mit der jeweiligen nachimplementierten Version. Als Wertung wurde der Durchschnitt der positiven und negativen F1-Scores verwendet. Wie der F1-Score berechnet wird, haben wir im Kapitel 5 genauer beschrieben. Wie man sieht, hat sich die Genauigkeit der Systeme von GU-MLT-LT und KLUE durch unser Vorgehen stark erhöht, die Genauigkeit von NRC ist nahezu gleich geblieben und die

Genauigkeit von TeamX hat sich verschlechtert. Dies könnte damit zusammenhängen, das wir den Wortsinn nicht als Feature bestimmen konnten.

System	SemEval 2013	Unsere Version
NRC	69,02	69,52
GU-MLT-LT	65,27	67,58
KLUE	63,06	67,12

System	SemEval 2014	Unsere Version
TeamX	72,12	70,09

Tabelle 4.1: Vergleich zwischen SemEval-Teilnehmer und nachimplementiertem System auf den SemEval 2013 und 2014 Testdaten.

4.2 Kombination

Nachdem wir alle Systeme nachimplementiert haben, versuchten wir nun, die Ergebnisse der einzelnen Systeme zu einem Ergebnis zu kombinieren, dazu testeten wir verschiedene Ansätze. Unsere Tests zeigten, dass jedes System auf unterschiedliche Ergebnisse kommt. Teilweise lieferte sogar nur ein System die richtige Antwort und alle anderen Systeme lagen falsch. Manchmal sogar absolut falsch, das heißt, es wurden z. B. Tweets die eigentlich positiv sind als negativ erkannt. Wir untersuchten deshalb verschiedene Kombinierungsansätze. Zunächst untersuchten wir, wie wir die Trainingsdaten auf die einzelnen Systeme aufteilen können, dazu verwendeten wir verschiedene Methoden wie Bagging und Boosting. Doch zeigte sich das dies keine Verbesserung brachte, deshalb entschieden wir uns dafür, alle Trainingsdaten für jedes System zu verwenden.

Nun suchten wir weiter nach der besten Methode, um die einzelnen Ergebnisse der Systeme zu kombinieren. Dazu untersuchten wir verschiedene Möglichkeiten, z. B. verwendeten wir das Ergebnis, das die meisten Systeme vorhersagten oder wir verwendeten immer das Ergebnis eines bestimmten Systems außer es sagt eine bestimmte Klasse vorher, dann wählten wir die Mehrheit der anderen Systeme. So versuchten wir, besonders häufige Fehlvorhersagen auszugleichen.

Doch letztendlich entschieden wir uns, nicht die Vorhersagen der Systeme, also positiv, neutral oder negativ zu verwenden, sondern die Wahrscheinlichkeiten der einzelnen Klassen. Denn es ist auch möglich, sich vom Klassifikator eine Wahrscheinlichkeit für die jeweilige Klasse geben zu lassen. Das heißt, die Aussage des Klassifikators ist nicht, dass der Tweet positiv ist, sondern

der Tweet ist z. B. mit einer Wahrscheinlichkeit von 50% positiv, mit einer Wahrscheinlichkeit von 30% neutral und mit einer Wahrscheinlichkeit von 20% negativ. So lässt sich besser einschätzen wie sicher der Klassifikator mit seiner Aussage ist. Denn es macht einen Unterschied ob ein Tweet mit einer Wahrscheinlichkeit von 80% positiv ist oder mit einer Wahrscheinlichkeit von 50%. Diese Informationen gehen allerdings verloren wenn man nur die Vorhersage des Klassifikators betrachtet und nicht die Wahrscheinlichkeiten. Dieses Vorgehen verwendete auch das System vom Team X. Sie berechneten durch mehrere Tests Gewichtungen für die einzelnen Klassen und multiplizierten diese mit den Wahrscheinlichkeiten und erhielten so ein gewichtetes Ergebnis (Miura et al., 2014). Wir gewichteten die Wahrscheinlichkeiten allerdings alle gleich und entschieden uns letztendlich dafür, den Durchschnitt aller Wahrscheinlichkeiten der gesamten Systeme für jede Klasse zu berechnen, da dies in unseren Tests die höchste Genauigkeit erzielte.

Zusammengefasst gingen wir wie folgt vor. Als Erstes trainierten wir alle vier nachimplementierten Systeme einzeln für sich mit den Trainingsdaten von 2013. Danach klassifizierten wir die Tweets der Testdaten mit jedem System einzeln. So erhielten wir von jedem Klassifikator die einzelnen Wahrscheinlichkeiten für die drei Klassen, positiv, neutral und negativ. Aus diesen Wahrscheinlichkeiten wurde dann der Durchschnitt aus allen Systemen für jede Klasse berechnet. Die Klasse, die so die größte Wahrscheinlichkeit erhielt, wurde als Endaussage verwendet. Dieses einfache System lässt sich sehr leicht nachimplementieren und erreicht trotz seiner Einfachheit eine hohe Genauigkeit wie man im Kapitel 5 sehen kann.

Kapitel 5

Evaluierung

Zur Evaluierung unseres Systems orientierten wir uns an der Unteraufgabe B der Twitteraufgaben des SemEval 2013 und 2014, genauer gesagt, die Sentimentanalyse eines gesamten Tweets. Dazu verwendeten wir die bereitgestellten Trainings-, Entwicklungs- und Testdaten und verglichen unser System mit denen der Teilnehmer des SemEval 2013 und 2014.

5.1 Aufbau der Evaluation

Zum Trainieren unseres Systems verwendeten wir die Trainingsdaten des SemEval 2013, diese bestehen aus 9.728 Tweets (3.662 positive, 4.600 neutrale und 1.466 negative). Wir verwendeten die Entwicklungsdaten des SemEval 2013 um unsere Klassifikatoren einzustellen und um die bestmögliche Kombination der einzelnen Systeme zu finden, wie in Kapitel 4 beschrieben. Die Entwicklungsdaten bestehen aus 1654 Tweets (575 positive, 739 neutrale und 340 negative). Für die Evaluierung testen wir dann unser System mit den Testdaten des SemEval 2013 und mit den Testdaten des SemEval 2014. Die Testdaten des SemEval 2013 bestehen aus 3.813 Tweets (1.572 positive, 1.640 neutrale und 601 negative) und die Testdaten des SemEval 2014 bestehen aus 1.853 Tweets (982 positive, 669 neutrale und 202 negative). Bei der Evaluierung der 2014er Testdaten verwendeten wir auch die 2013er Trainingsdaten zum Trainieren, da die Organisatoren für die Evaluierung der 2014er Testdaten keine neuen Trainingsdaten zur Verfügung stellten, sondern die Trainingsdaten von 2013 nutzten. Die verwendeten Tweets in den Daten wurden zwischen Januar 2012 und Januar 2013 gesammelt. Dabei wurden hauptsächlich Tweets ausgewählt, die zu dieser Zeit populäre Themen enthielten. Diese Tweets wurden dann per Hand klassifiziert. Um die Systeme der Teilnehmer zu vergleichen und in eine Rangliste einordnen zu können, berechneten die Organisatoren für jedes System einen so genanten F1-Score. Dieser berechnet sich aus der Genauigkeit

und der Trefferquote. Die Genauigkeit berechnet sich aus richtig erkannten Tweets durch alle erkannten Tweets einer Klasse. Für positive Tweets ergibt sich daraus die Formel 5.1.

$$p_{pos} = \frac{\text{wirklich positiv und als positiv erkannte Tweets}}{\text{alle als positiv erkannten Tweets}} \tag{5.1}$$

Die Trefferquote berechnet sich aus richtig erkannten Tweets durch alle tatsächlichen Tweets einer Klasse. Für positive Tweets ergibt sich daraus die Formel 5.2.

$$r_{pos} = \frac{\text{wirklich positiv und als positiv erkannte Tweets}}{\text{alle wirklich positiven Tweets}} \tag{5.2}$$

Für positive Tweets ergibt sich so der F1-Score aus der Formel 5.3.

$$F_{pos} = 2 * \frac{p_{pos} * r_{pos}}{p_{pos} + r_{pos}} \tag{5.3}$$

Zur Berechnung der F1-Score für mehrere Klassen, in unserm Fall drei, berechnet man den Durchschnitt der einzelnen F1-Scores. Im Fall von SemEval wurden für die Berechnung der Endwertung nur der F1-Score von positiven und negativen Tweets genommen, wodurch sich die Formel 5.4 für den F1-Score ergibt.

$$F_1 = (F_{pos} + F_{neg})/2 \tag{5.4}$$

5.2 Ergebnis

In der Tabelle 5.1 und Tabelle 5.2 kann man sehen, wie unser System unter den anderen Teilnehmern einzuordnen wäre. Man kann deutlich erkennen, das unser System mit den Testdaten von 2013 und unter den Teilnehmern von 2013 den ersten Platz von 38 Teilnehmern belegte und so ein besseres Ergebnis ablieferte als die einzelnen Systeme, die wir verwendeten. Unter den Teilnehmern des SemEval 2014 und den Testdaten von 2013 erzielten wir den zweiten Platz von 50 Teilnehmern und mit den Testdaten von 2014 belegten wir den fünften Platz.

5.3 Einfluss der Einzelsysteme

In der Tabelle 5.3 kann man sehen, wie groß der Einfluss der einzelnen Systeme auf das gesamte System ist. Dazu haben wir jeweils nur drei Systeme miteinander kombiniert und ein System weggelassen, so kann man feststellen wie das Ergebnis des Gesamtsystems ohne dieses System ist. Man kann sehen, dass

der Einfluss der einzelnen Systeme eher gering ist, da sich das Ergebnis kaum verändert, wenn man nur ein System weglässt. Auch kann man erkennen, dass alle Systeme wichtig für das Gesamtsystem sind, da die Genauigkeit immer fällt, wenn ein System weggelassen wird, außer bei dem System von NRC mit den Testdaten von 2014, da kann man erkennen, dass die Genauigkeit steigen würde, wenn man es weglassen würde.

5.4 Fehleranalyse

Wie in den Konfusionsmatrizen in Tabelle 5.4 zu sehen, passieren die meisten Fehler bei der Klassifizierung von neutralen Tweets, die als positiv erkannt werden und bei der Klassifizierung von positiven Tweets die als neutral erkannt werden. Der Fall, dass Tweets die positiv sind als negativ erkannt werden, und umgekehrt, tritt nicht so häufig ein. Diese Fehlklassifizierung ist allerdings auch viel schwerwiegender als die Fehlklassifizierung im neutralen Bereich. Die Fehlklassifizierung von positiven Tweets, die als neutral erkannt werden, kann leicht damit zusammenhängen, dass die eigentlich positiven Tweets nur wenig Anhaltspunkte für eine positive Aussage liefern und daher eher als neutral erkannt werden. Des Weiteren muss man beachten das die Zahlen im negativen Bereich viel kleiner sind, da in den Testdaten viel weniger negative Tweets vorkommen als positive oder neutrale. Weitere Möglichkeiten für eine Fehlklassifizierung können Tweets sein, die eher sarkastisch geschrieben wurden. Diese lassen sich von Menschen möglicherweise leichter erkennen, doch der Klassifikator kann mit solchen Tweets Probleme haben. Dies bestätigen auch weitere Testdaten, denn zur SemEval 2014 wurden auch Testdaten zur Verfügung gestellt, die nur sarkastische Tweets enthielten.

Mit diesen Testdaten hatten alle Teilnehmer des SemEval 2014 Probleme. Der beste Teilnehmer für sarkastische Tweets erreichte nur eine F1-Score von 58,16. Unser System belegte mit 51,86 den 13. Platz wie der Tabelle 5.2 zu entnehmen ist.

Ein weiteres Problem ist, dass nicht allen Teilnehmern die kompletten Tweets zur Verfügung standen, da diese von jedem Teilnehmer selbst mit Hilfe der Twitter API heruntergeladen werden mussten, da die Twitter Nutzungsbedingungen nur diese Art des Downloads zulassen. Somit kann es allerdings vorkommen, dass einige Tweets nicht mehr zur Verfügung standen, als sich die Teilnehmer die Tweets heruntergeladen haben, wodurch einige Teilnehmer ihr System nur mit weniger Tweets trainieren konnten.

Team	F1-Score
Unser System	**71,56**
NRC-Canada	69,02
GU-MLT-LT	65,27
teragram	64,86
BOUNCE	63,53
KLUE	63,06
AMI&ERIC	62,55
FBM	61,17
AVAYA	60,84
SAIL	60,14
...	...
Durchschnitt	53,70

Tabelle 5.1: Ergebnisse SemEval 2013, Top 10 von 39 (Teilnehmer SemEval 2013 + Unser System).

Team	Twitter 2014	Twitter 2013	Twitter2014Sarcasm
TeamX	70,96	72,12	56,50
coooolll	70,14	70,40	46,66
RTRGO	69,95	69.10	47,09
NRC-Canada	69,85	70,75	58,16
Unser System	**69,72**	**71,56**	**51,86**
TUGAS	69,00	65,64	52,87
CISUC KIS	67,95	67,56	55,49
SAIL	67,77	66,80	57,26
SWISS-CHOCOLATE	67,54	64,81	49,46
Synalp-Empathic	67,43	63,65	51,06
...	...	...	...
Durchschnitt	60,41	59,78	45,44

Tabelle 5.2: Ergebnisse SemEval 2014, Top 10 von 51 (Teilnehmer SemEval 2014 + Unser System), auf den 2013-er, 2014-er und Sarkasmus Testdaten. Ergebnis auf den Sarkasmus Testdaten dient zur Fehleranalyse.

System	F1-Score 2013	F1-Score 2014
Alle	71,56	69,72
Alle - KLUE	71,02 (-0,54)	69,57 (-0,15)
Alle - GU-MLT-LT	71,14 (-0,42)	68,62 (-1,10)
Alle - NRC	71,14 (-0,42)	70,71 (+0,99)
Alle - TeamX	71,09 (-0,47)	69,11 (-0,61)

Tabelle 5.3: F1-Scores des Gesamtsystems jeweils mit einem System entfernt auf den 2013-er und 2014-er Testdaten.

		Testergebnis					
		SemEval 2013			SemEval 2014		
		postiv	**neutral**	**negativ**	**postiv**	**neutral**	**negativ**
Referenz	**positiv**	1257	223	89	788	162	32
	neutral	413	1097	130	191	427	51
	negativ	82	102	417	47	30	125

Tabelle 5.4: Konfusionsmatrix von Referenz und Testergebnis auf den 2013-er und 2014-er Testdaten.

Kapitel 6

Zusammenfassung und Ausblick

Wir haben uns in dieser Arbeit mit der Sentimentanalyse von Tweets beschäftigt. Hauptsächlich beschäftigten wir uns dabei mit der Unteraufgabe B der Aufgabe 2 des SemEval 2013 und Aufgabe 9 des SemEval 2014. Diese Aufgabe beschäftigte sich mit der Sentimentanalyse eines gesamten Tweets, es sollte bestimmt werden ob die Stimmung eines Tweets positiv, neutral oder negativ ist. Wir implementierten dazu vier Sentimentanalysesysteme der Teilnehmer nach und kombinierten die Ergebnisse dieser zu einer Endaussage.

In dieser Arbeit zeigten wir dazu im Kapitel 2, wie wichtig die Sentimentanalyse für bestimmte Personen, Unternehmen oder Organisationen sein kann. Des Weiteren stellten wir im Kapitel 2 einige Verfahren zur Sentimentanalyse vor und erklärten die Aufgabe des SemEval genauer. Im Kapitel 3 beschäftigten wir uns ausführlicher mit den Teilnehmern des SemEval, wählten vier der Teilnehmer aus und beschrieben deren Sentimentanalysesysteme genauer. Die verwendeten Vorverarbeitungsschritte, Features und Sentimentlexika wurden außerdem detaillierter beschrieben. Die Systeme der vier Teilnehmer wurden dann von uns nachimplementiert und kombiniert, was im Kapitel 4 genauer beschrieben wurde. Das System, das wir so erhielten, wurde anschließend im Kapitel 5 evaluiert. Wir verglichen unser System mit denen der Teilnehmer und führten eine Fehleranalyse durch.

Wie sich in dieser Arbeit zeigt, kann durch die einfache Kombination von mehreren Sentimentanalysesystemen ein Ensemblesystem entstehen das eine höhere Genauigkeit erreicht als die einzelnen Systeme, aus denen es besteht. Unser System erreichte so eine Platzierung unter den besten fünf Systemen des SemEval 2014 und belegte den ersten Platz unter allen Teilnehmern des SemEval 2013. Durch diese Arbeit wurde ein Sentimentanalysesystem geschaffen, das auf Augenhöhe mit den derzeit aktuellen Systemen ist und sich leicht implementieren lässt.

Für die Weiterentwicklung unseres Systems wäre es interessant, weitere Sentimentanalysesysteme in unser Ensemble aufzunehmen. Sinnvoll wären Systeme die komplett neue Features verwenden oder auch Systeme, die ein komplett anderes Vorgehen verwenden, wie z. B. ein System das unüberwachtes Lernen verwendet. Wichtig wäre vor allem ein System zu finden, das die Anzahl an Fehlklassifizierungen verringert. Für die Weiterentwicklung wäre auch es auch sinnvoll, zu sehen, ob man noch ein besseres Kombinationsverfahren findet, das die Genauigkeit des Systems noch weiter erhöhen würde.

Interessant wäre auch, wie unser derzeitiges System zur SemEval 2015 abschneiden würde. Da unser System sehr gute Ergebnisse mit Tweets liefert, wäre es auch gut zu wissen wie das Ergebnis unseres Systemes ausfallen würde, wenn man andere Daten verwendet, wie z. B. Filmkritiken oder Produktreviews.

Literaturverzeichnis

Lars Asker and Richard Maclin. Ensembles as a sequence of classifiers. In *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*, IJCAI'97, pages 860–865, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-555860-480-4. 2.5

Andrea Esuli Stefano Baccianella and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7. 3.3

Alexandra Balahur and Marco Turchi. Improving sentiment analysis in twitter using multilingual machine translated data. Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, pages 49–55. INCOMA Ltd. Shoumen, BULGARIA, 2013. 2.3

Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 2.3

Johan Bollen, Alberto Pepe, and Huina Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, abs/0911.1583, 2009. 2.3

Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996. ISSN 0885-6125. 2.5

Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllll!!!!!!!!!!!!!!: Using word lengthening to detect sentiment in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 562–570, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. 2.3

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, dec 1992. ISSN 0891-2017. 3.1

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, jun 1998. ISSN 1384-5810. 2.2

Jorge Carrillo de Albornoz, Laura Plaza, Pablo Gervás, and Alberto Díaz. A joint model of feature mining and sentiment analysis for product review rating. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 55–66, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. 2.3

Gianluca Demartini. Ares: A retrieval engine based on sentiments sentiment-based search result annotation and diversification. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 772–775, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. 2.3

Nicholas A. Diakopoulos and David A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1195–1198, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. 2.3

Sergei Ermakov and Liana Ermakova. Sentiment classification based on phonetic characteristics. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 706–709, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-36972-8. 2.3

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 4.1

Ronen Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, apr 2013. ISSN 0001-0782. 2

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996. 2.5

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Haixun Wang, David W. Cheung, and Huan Liu. A balanced ensemble approach to weighting classifiers for text classification. *2013 IEEE 13th International Conference on Data Mining*, 0: 869–873, 2006. ISSN 1550-4786. 2.5

M. Ghiassi, J. Skinner, and D. Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Syst. Appl.*, 40(16):6266–6282, November 2013. ISSN 0957-4174. 2.3

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009. 2.3

Tobias Günther and Lenz Furrer. Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 328–332, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 3.4.2

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. 4.1

Yulan He. Latent sentiment model for weakly-supervised cross-lingual sentiment classification. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 214–225, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. 2.3

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. 3.3

Jussi Karlgren, Magnus Sahlgren, Fredrik Olsson, Fredrik Espinoza, and Ola Hamfors. Usefulness of sentiment analysis. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 426–435, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-28996-5. 2.3

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *ICWSM*. The AAAI Press, 2011. 2.3

Kar Wai Lim and Wray Buntine. Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In *ACM International Conference on Information and Knowledge Management (CIKM)*. 2.3

Richard Maclin and David W. Opitz. Popular ensemble methods: An empirical study. *CoRR*, 2011. 2.5

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. 3.4.4, 4.2

Saif M. Mohammad and Peter D. Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 3.1, 3.3

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242, 2013. 3.3, 3.4.1

A. Moniz and F. M. G. de Jong. Sentiment analysis and the impact of employee satisfaction on firm earnings. In *36th European Conference on IR Research, ECIR 2014, Amsterdam, the Netherlands*, volume 8416 of *Lecture Notes in Computer Science*, pages 519–527, London, April 2014. Springer Verlag. 2.3

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 2.4

Finn Årup Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903, 2011. 3.3

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of NAACL*, 2013. 3.1

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 2, 2.1, 2.2, 3.1

R. Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, Third 2006. 2.5

Thomas Proisl, Paul Greiner, Stefan Evert, and Besim Kabashi. Klue: Simple and robust methods for polarity classification. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 395–401, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 3.4.3

Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2): 1–39, 2010. ISSN 0269-2821. 2.5

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. 2.4

Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2): 197–227, July 1990. ISSN 0885-6125. 2.5

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA, 1966. 3.3

Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 2, 2.1, 2.1

Twitter. About twitter, 2014a. URL `https://about.twitter.com/company`. Accessed: 23.11.2014. 1

Twitter. Twitter reports third quarter 2014 results, Oct 2014b. URL `https://investor.twitterinc.com/releasedetail.cfm?ReleaseID=878170`. Accessed: 23.11.2014. 1

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. 3.3

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1980–1984, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. 2.3