INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Projektseminar Natural Language Processing (SS 2016)

Philipp Kochanski
Tobias Cabanski

1 Einleitung und Zielsetzung

Die sich jedes Jahr wiederholende Konferenzreihe SemEval stellt verschiedene Aufgaben im Bereich Natural Language Processing und speziell im Teilgebiet Sentiment Analysis zur Verfügung. In dieser Ausarbeitung wird ein Lösungsansatz für den in der SemEval-2016 gestellten Task 4 „Sentiment Analysis in Twitter“ vorgestellt. Die Aufgabe ist in fünf Subtasks unterteilt, von denen zwei bearbeitet wurden.

In Subtask A soll für einen bestimmten Tweet die Polarität bestimmt werden, für die eine dreistufige Skala vorgegeben ist: positiv, neutral und negativ.

Subtask C behandelt die Vergabe der Polarität für ein bestimmtes Topic, das in einem Tweet genannt wird. Hier soll eine fünfstufige Skalierung zum Einsatz kommen: -2, -1, 0, 1, 2, wobei -2 sehr negativ und 2 sehr positiv bedeutet.

2 Datengrundlage

Zur Bestimmung der Polarität der Tweets wurden bereits annotierte Trainingsdaten zur Verfügung gestellt, die sich in dev-, devtest- und test-Daten aufteilen und für Subtask A insgesamt 6295 Tweets liefern. Zusätzlich dürfen auch die Daten aus der SemEval-2013 benutzt werden, wodurch weitere 8833 Tweets dazu kommen, welche zum Trainieren benutzt werden dürfen. Für Subtask C werden insgesamt 6149 Trainingsdaten bereitgestellt.

3 Vorverarbeitung der Daten

Bevor tiefer gehende Verfahren auf den Trainingsdaten angewendet werden, kommt es zu einem Vorverarbeitungsschritt, mit dem die Tweets bereinigt werden. Komplett ignoriert werden Tweets, die keine ASCII-Zeichen enthalten, was ein deutlicher Hinweis darauf ist, dass der komplette Tweet aus asiatischen Schriftzeichen besteht.

In nahezu jedem Tweet kommt eine URL vor, weshalb das Vorhandensein dieser keine Aussagekraft über die Polarität hat und somit entfernt wird. Weiterhin werden bestimmte Symbole wie Klammern und Semikolons, Punkte an Wortenden und mehrfache Whitespaces entfernt. Insbesondere wurde auch das Raute-Symbol entfernt, welches in Twitter als Hashtag bekannt ist und in vielen Tweets vorkommt. Hinter dem Hashtag stehen für gewöhnlich Wörter, die sich nach dessen Entfernung ebenfalls sehr gut für die Bestimmung der Polarität nutzen lassen. Ziffern und Wörter mit Ziffern wurden ebenfalls herausgenommen, z.B. 22nd.

Einzelne Wörter wurden so modifiziert, dass sie von weiterführenden Verfahren und Lexika, z.B. dem SentiWordNet, erkannt und benutzt werden können. Ein häufiges Phänomen, welches bei der Sentiment Analyse von Social Media Daten auftritt, ist die Elongation von Wörtern, z.B. Awesomeeeeeeeee. Hier soll das Wort so modifiziert werden, dass die zu viel vorkommenden Buchstaben in einem Wort erkannt und entfernt werden.

Eine weitere Modifikation erfolgt an Wörtern, die zusammen geschrieben wurden, aber eigentlich mehrere Einzelwörter darstellen, z.B. BringTheCupHome. Das Wort soll dann so modifiziert werden, dass die Wörter einzeln aufgeführt werden, also aus dem Beispiel dann Bring The Cup Home entsteht. Dabei wird aber darauf geachtet, dass bestimmte Wörter wie iPad oder iPhone so erhalten bleiben.

Abschließend werden alle Großbuchstaben durch Kleinbuchstaben ersetzt.

4 Subtask A

Im Folgenden wird beschrieben, wie die weitere Verarbeitung der Daten für Subtask A erfolgt und mit welchen Methoden die Polarität für einen Tweet in einer dreistufigen Skalierung bestimmt wird.

4.1 Standardverfahren

Als Standardverfahren werden in diesem Fall die drei Machine Learning Verfahren Support Vector Machines, K-Nearest-Neighbours und Naive Bayes bezeichnet, die mit relativ wenig Aufwand mit Daten trainiert und dann auf nicht annotierte Daten angewendet werden können, um eine Vorhersage zu erhalten. Als Features wurden hier Wort-N-Gramme verwendet.

4.2 Okapi BM25 Verfahren

Das Verfahren Okapi BM25 bedient sich der in [2] definierten Formel

$$score(D,Q) = \sum_{i=1}^{n} idf(q_i) \frac{tf_{D_{q_i}} \cdot (k_1 + 1)}{tf_{D_{q_i}} + k_1 \cdot (1 - b + b \cdot \frac{|D|}{ADL})}$$

welche eine Variante der tf-idf Formel darstellt. Als Eingabeparameter wird ein Dokument D und eine Anfrage Q übergeben und berechnet wird ein Score, der die Ähnlichkeit der Eingaben darstellt.

Das besondere an der Formel im Vergleich zu anderen Ähnlichkeitsformeln wie tf-idf ist, dass die Dokumentenlänge $|D|$ in Form der Anzahl der Wörter eines Tweets in die Berechnung mit einfließt, genauso wie die durchschnittliche Länge aller Dokumente ADL. Eine weitere Besonderheit ist, dass die zwei Parameter b und k_1 zur Optimierung der Ergebnisse benutzt werden können.

Die Implementierung des Verfahrens erfolgt so, dass der Score für ein Testdokument gegen alle vorhandenen Trainingsdokumente berechnet wird. Die Trainingsdokumente werden dann absteigend nach dem Score sortiert und es werden die Top n dieser Dokumente ausgewählt, um dann durch eine Mehrheitsentscheidung die Polarität des Testdokumentes zu bestimmen. Durch diese Implementierung entsteht neben den Optimierungsparametern b und k_1 noch ein weiterer Parameter n, mit dem die Ergebnisse optimiert werden können.

4.3 SentiWordNet-Vader-Verfahren

Dieses Verfahren nutzt zum einen das SentiWordNet [1], welches aus dem WordNet [3] entstanden ist und zu jedem Synset zwei Polaritätswerte zuordnet. Diese werden als PosScore und NegScore bezeichnet und können Werte zwischen 0 und 1 annehmen. Aus diesen beiden Werten lässt sich noch ein dritter Score für die Objektivität berechnen, der ObjScore. Dieser wird durch die Formel ObjScore = 1 - PosScore + NegScore berechnet. Die Wörter eines Tweets werden mittels des Wortes selbst und des POS-Tags zum SentiWordNet zugewiesen.

Zum anderen kommt das Lexikon zum Einsatz, welches in dem Analysetool Vader Sentiment [4] benutzt wird und von C.J. Hutto und Eric Gilbert entwickelt wurde. Dieses Lexikon besitzt 7525 Einträge und deckt vor allem Begriffe ab, die häufig auf Plattformen der sozialen Medien benutzt werden. Die Polarität eines Wortes kann hier einen Wert zwischen -4 für sehr negativ und 4 für sehr positiv annehmen.

In der Implementierung wird ermittelt, welcher Score für ein Wort jeweils im SentiWordNet und im Vader Lexikon gefunden wird. Die ermittelten Werte werden dann zum Klassifikationsscore zusammen genommen, der sich durch PosScore - NegScore + VaderScore berechnet. Bei diesem Score entscheidet dann das Vorzeichen über die Polarität des zu testenden Tweets.

4.4 Ensemble

Aus den vorher genannten Einzelverfahren ist das Ensemble-Verfahren entstanden, indem die Ergebnisse der einzelnen Verfahren zusammen genommen wurden und eine Mehrheitsentscheidung getroffen wurde. Um die Ergebnisse weiter zu optimieren, werden die einzelnen Verfahren gewichtet. Tabelle 1 zeigt eine Übersicht der Ergebnisse der Einzelverfahren sowie des Ensemble-Verfahrens.

Verfahren	**Test F1**
SVM	0.495
KNN	0.194
MNB	0.519
OKAPI	0.416
SWNV	0.496
ENSEMBLE	**0.549**

Tabelle 1: Ergebnisse Subtask A

Bei der Evaluation der Verfahren wurden die dev-, devtest- und test-Datensätze sowie die Tweets aus der SemEval-2013 zum Trainieren benutzt. Zum Testen wurden die offiziellen SemEval-2016 Testdaten benutzt, welche 32009 Tweets enthalten. Als Evaluationsmaß wurde der F1-Score verwendet, welcher ein gewichtetes Mittel über den Precision- und Recall-Wert bildet. Die Ergebnisse zeigen, dass das Naive Bayes Verfahren mit

einem F1-Score von 0.519 das beste Einzelverfahren ist. Durch den Einsatz des Ensemble-Verfahrens konnte der Score auf 0.549 angehoben werden.

5 Subtask C

In diesem Abschnitt wird erklärt, wie das weitere Vorgehen für Subtask C aussieht, in dem die Polarität zu einem gegebenen Topic in einem Tweet berechnet werden soll. In diesem Fall soll eine fünfstufige Skalierung der Polarität erreicht werden.

5.1 Dependency Parsing

Ein erster Ansatz für das Auffinden relevanter Wörter, die zu einem Topic gehören können, ist das Dependency Parsing. Dieses kommt aber bei Tweets schnell an seine Grenzen, da sich diese häufig nicht an die Regeln der Grammatik halten und manchmal auch keine natürliche Sprache verwendet wird. Oft zu Problemen führen auch Terminalsymbole, die willkürlich gesetzt wurden. Dadurch kann es vorkommen, dass der Parser einen Tweet in mehrere Sätze spaltet und somit die Verbindung von Topic und Informationen verloren geht.

Eine naive Lösung für dieses Problem kann dadurch erreicht werden, dass einer bestimmten Anzahl von Verbindungen gefolgt wird, die auf das Topic zeigen oder von ihm weggehen. Diese Wörter werden dann an das aus Subtask A bekannte SentiWordNet-Vader-Verfahren übergeben, welches die Wörter klassifiziert und so die Polarität berechnet werden kann. Wenn zu wenig Wörter gefunden werden, werden alle Wörter aus dem Tweet zur Klassifikation genommen.

5.2 Weitere Verfahren

Ein weiteres Verfahren ist das Betrachten der benachbarten Wörter des Topics in einem Tweet. Dieses Verfahren beruht auf der Annahme, dass die Umgebung des Topics möglicherweise relevante Informationen enthält und somit Wörter, die eine bestimmte Entfernung zum Topic aufweisen, ignoriert werden können.

Das Verfahren aus Subtask A kann ebenfalls auf Subtask C angewendet. Dies bedeutet, dass das Topic ignoriert wird und nur eine Klassifikation des Tweets stattfindet. Durch geringe Anpassungen wegen der fünfstufigen Skalierung können die Verfahren aus Subtask A mit den Daten aus Subtask C arbeiten.

5.3 Ensemble

Wie auch schon in Subtask A, wurden die einzelnen Verfahren in Subtask C zu einem Ensemble-Verfahren zusammengefasst, welches einem Tweet mit einem gegebenen Topic eine Polarität zuweist. Die Ergebnisse lassen sich in Tabelle 2 einsehen.

Verfahren	**Test MAE**
SVM	1.058
KNN	1.418
MNB	**1.017**
OKAPI	1.226
SWNV	1.099
ENSEMBLE 1	1.021
DEPEN	1.128
AREA	1.244
ENSEMBLE 2	1.040

Tabelle 2: Ergebnisse Subtask C

Zum Trainieren der Verfahren wurden die dev-, devtest- und train-Datensätze aus der SemEval-2016 verwendet. Wie schon bei Subtask A wurde bei Subtask C mittels des offiziellen Testdatensatzes evaluiert, welcher insgesamt 20632 Tweets umfasst. Als Evaluationsmaß wird hier der macroaveraged mean absolute error benutzt, bei dem ein niedriger Wert ein gutes Ergebnis anzeigt. Die Evaluation hat ergeben, dass Multinominal Bayes mit einem MAE-Score von 1.017 nicht nur das beste Einzelergebnis, sondern auch das beste Gesamtergebnis liefert und somit die Ensemble-Verfahren schlägt. Das Ensemble-1-Verfahren ist ein gewichtetes Verfahren über SVM, KNN,

MNB, OKAPI und SWNV. Beim Ensemble-2-Verfahren kommen noch die Einzelverfahren DEPEN und AREA dazu.

6 Fazit

In dieser Ausarbeitung wurde gezeigt, wie sich die Sentiment Analyse von Tweets, wie in der Challenge SemEval 2016 Task 4 als Aufgabe gestellt, lösen lässt. Es wurden Standardverfahren aus dem Bereich Machine Learning angewandt sowie individuelle Verfahren, die den Okapi BM25-Score oder Wörterbücher benutzen. Für Subtask C, wo das Topic eines Tweets ebenfalls eine Rolle spielt, wurde zusätzlich noch ein Depedency-Parsing-Verfahren und ein Verfahren, welches die benachbarten Wörter betrachtet, angewandt. Diese Verfahren wurden dann zu einem Ensemble-Verfahren verknüpft, welches die Ergebnisse der Einzelverfahren gewichtet und über eine Mehrheitsentscheidung die Klassifikation eines Tweets vornimmt.

Die Klassifikationsergebnisse zeigen, dass bei Subtask A das Wörterbuchverfahren SentiWordNet-Vader und bei Subtask C das Machine Learning Verfahren Multinominal Bayes die besten Ergebnisse liefern. Bei beiden Subtasks konnten die Ergebnisse durch anwenden des Ensemble-Verfahrens noch weiter verbessert werden.

7 Ausblick

Während der Bearbeitung der Challenge hat sich gezeigt, dass das Auffinden von Features in kurzen Texten ein Problem darstellt. Bei den Standardverfahren, die N-Gramme als Features benutzten, könnte eine Verbesserung des Ergebnisses erreicht werden, wenn diese verschiedene Features benutzen würden. Als Feature in Frage kommen würde z.B. die word2vec Methode [5], welche durch den Einsatz neuronaler Netze die Wörter von Texten in Vektorform bringt und dabei automatisch Wortzusammenhänge erkennt.

Als weiteres Feature wäre es möglich, sich neben den zwei verwendeten Wörterbüchern SentiWordNet und Vader ein eigenes Wörterbuch aus den Trainingsdaten zusammenzustellen. Der Vorteil an einem solche Wörterbuch ist, dass es speziell für diese Domäne entwickelt wird und so bestimmte Wörter eventuell besser als positiv oder negativ erkennen kann, als die vorgefertigten Wörterbücher. Auch wäre es möglich, ein solches Wörterbuch nicht nur auf Unigramm-Basis aufzubauen, sondern zusätzlich auf Bi- und Trigramm-Basis, was einen weiteren Informationsgewinn bringen könnte.

Literatur

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. European Language Resources Association (ELRA), 2010.

[2] Stefan Conrad. Information Retrieval und Natural Language Processing. Vorlesungsskript, 2015.

[3] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[4] C.J. Hutto and E.E. Gilbert. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *ICWSM-14*. The AAAI Press, 2014.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. 2013.

[6] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18. Association for Computational Linguistics, 2016.