Abschlussbericht: Sentimentanalyse auf Amazon-Reviews

Designing Experiments for Machine Learning Tasks

bei Éva Mújdricza-Maydt

Institut für Computerlinguistik

Ruprecht-Karls-Universität Heidelberg

Caroline Berg Simon Will

27. Februar 2016

Inhaltsverzeichnis

1 Einführung **3**

2 ähnliche Projekte **3**

3 Vorverarbeitung der Daten **3**
3.1 Amazon-Scraper 3
3.2 TreeTagger 3
3.3 Chunks 4

4 SentiWS und Attribute **4**
4.1 Sentimentannotation 4
4.2 Attribute 4

5 Experimente und Ergebnisse **4**
5.1 Klassifizierer 4
5.2 erzielte Resultate 5
5.3 Test auf fremder Domäne 5

6 Fazit **5**

7 Literaturverzeichnis **5**

1 Einführung

Sentiment analysis als Teilgebiet des maschinellen Lernens verfolgt die Aufgabe, einem gegebenen Text, oder Teilen des Textes, einen entsprechenden *sentiment*-Wert, d.h. eine Zahl auf einer definierten Skala, die etwas über den Grad der Positivtät, bzw. Negativität des Textes aussagt, zuzuordnen.
Als Datengrundlage haben wir Review-Texte von Amazon gewählt. Diese schienen sich gut für eine Verarbeitung mithilfe des maschinellen Lernens zu eignen, da ein Kunde, der ein Review verfasst außerdem eine Anzahl von Sternen angeben muss, welche sich als Klassenattribut zum trainieren eines Algorithmus eignet.

2 ähnliche Projekte

Obwohl wir uns nicht direkt an einem bereits vorhandenen Projekt aus diesem Bereich orientiert haben, kann man als vergleichbare Arbeit und zur kritischen Bewertung des Aufbaus unserer Sentiment-Analyse das Projekt *Analyses in Amazon Reviews Using Probablistic Machine Learning* von Callen Rain heranziehen. Es wurden gezielt Texte zu Produkten als Datengrundlage verwendet, die entweder besonders häufig bewertet wurden (z.B. Bücher, CDs, Filme, AmazonKindle) oder kaum Reviews aufwiesen (z.B. LeviJeans, MacBook). Das Problem der unterschiedlichen Verteilung von Reviews mit den jeweiligen Sternen wurde insofern gelöst, als dass nur Reviews mit 5 Sternen (score:1) und Reviews mit 1-2 Sternen (score:0) berücksichtigt wurden. Alle dazwischenliegenden Bewertungen wurden nicht in die Datengrundlage aufgenommen. Zu erwähnende Attribute in diesem Projekt sind zum einen der *bag of words* Ansatz bezüglich der 2000 häufigsten Wörter und der 500 häufigsten Bigramme, die Beachtung von Negation, sowie die Satzlänge der Reviewtexte, wobei besonders lange und besonders kurze Sätze in die Analyse miteinbezogen wurden. Die besten Ergebnisse wurden mit dem *Naïve Bayes Classifier* auf Reviews von AmazonKindle (84 % accuracy) und dem MacBook (88.2 % accuracy) erzielt. Zu beachten ist hierbei, dass die baseline bei 50 % liegt, da lediglich binär klassifiziert wurde.

3 Vorverarbeitung der Daten

3.1 Amazon-Scraper

Der Amazon-Scraper dient dem Herunterladen der für das Projekt erforderlichen Daten (z.B. bewertetes Produkt, Reviewtext und Anzahl der vergebenen Sterne). Hierbei wird automatisch die jeweils folgende Seite aufgerufen (es werden immer 10 Reviews pro Seite angezeigt) und die gesammelte Datenmenge wird im angegebenen Verzeichnis abgespeichert.

3.2 TreeTagger

Um die Daten zu formalisieren haben wir uns für den TreeTagger entschieden, welcher die Texte tokenisiert, lemmatisiert und mit POS-Tags versieht. Es wurden überraschend gute Resultate erzielt, beispielsweise wurden klein geschriebene Nomen richtig erkannt.

3.3 Chunks

Mit Blick auf die Verteilung der Sterne auf den heruntergeladenen Daten, fällt sofort auf, dass Produkte meist sehr gute Bewertungen (4-5 Sterne), eher selten sehr schlechte (1 Stern) und kaum mangelhafte Bewertungen (2-3 Sterne) verliehen bekommen.
Um zu gewährleisten, dass die Algorithmen jeweils auf der gleichen Menge Daten pro Anzahl von Sternen trainieren können, müssen die Daten balanciert werden. Dafür haben wir zunächst randomisierte Symlinks erstellt, die auf ein Review zeigen, und die Links anschließend in balancierten Chunks gespeichert. Durch das Verfahren, welches uns eine ausgewogene Datenmenge zum trainieren bereitstellt, mussten wir allerdings im Schnitt den Verlust von etwa einem Viertel unserer Anfangs heruntergeladenen Daten in Kauf nehmen.

4 SentiWS und Attribute

4.1 Sentimentannotation

Als deutschsprachige Datenbank, um das Sentiment eines Reviews zu bestimmen, bot sich SentiWS an. Es umfasst in etwa 33000 Wortformen und 3500 Lemmata. Dabei wird jedem Wort ein numerischer Wert von -1, für ein negativ konnotiertes Wort, bis 1 zugeordnet. In unserem Fall wird für einen Reviewtext geprüft, welche Wortformen ebenfalls in SentiWS vorkommen. Die aufsummierten Werte werden dann für die jeweilige Wortform (siehe Feature-Extraktion) separat gespeichert.

4.2 Attribute

Zur Extraktion der Features dient ein Python-Programm, das alternativ auch mit einem Shell-Skript aufgerufen werden kann.
Das Attribut token_number gibt die Länge des Reviewtextes an. Desweiteren werden die Sentimentwerte zum einen in Wortklassen eingeteilt. Wir haben jeweils ein Attribut für adjective_sentiment, noun_sentiment und verb_sentiment erstellt. Außerdem enthält das Attribut overall_sentiment die Summe aller Sentimentwerte, die gefunden wurden.
Alle Sentiment-Attribute können der `arff`-Datei auch in einer normierten Version (Vermerk auf Formel) übergeben werden. Als Klassenattribut kann entweder die Anzahl der Sterne für das jeweilige Review oder in einer weiteren Version das Attribut binary_judgement, welches allen Reviews mit 1-2 Sternen den Wert 0 und allen Reviews mit 4-5 Sternen den Wert 1 zuteilt, verwendet werden.

5 Experimente und Ergebnisse

5.1 Klassifizierer

Die besten Resultate haben wir mit folgenden Klassifizierern erzielt:

- J48
- RandomForest

- Naïve Bayes

5.2 erzielte Resultate

Tabelle einfügen

5.3 Test auf fremder Domäne

Für das Testen auf einer fremden Domäne haben wir zunächst auf Smartphone-Reviews trainiert, wobei die `arff`-Datei mit normierten Sentimentattributen erstellt wurde. Hierfür haben wir ein ernüchterndes Ergebnis von 27.7 % erzielt, was wir darauf zurückführen, dass Armbanduhren-Reviews im Schnitt nur circa ein Drittel der Länge von Smartphone-Reviews aufweisen. Bei genauerer Betrachtung der Texte fällt auch auf, dass Armbanduhren-Reviews zu einem großen Teil weniger emotionsgeladen formuliert werden.

6 Fazit

Trotz unseres naiven Ansatzes, was die Attributwahl betrifft, haben wir überraschend gute Ergebnisse erzielt.
Allgemein sollte man bei der Bewertung der Ergebnisse beachten, dass es zwischen den Reviewtexten erhebliche Unterschiede bezüglich der Formulierung, dem Umfang und der Komplexität des Textinhaltes gibt, da den Verfassern keine Normen oder Formalia bezüglich des Inhaltes oder Schreibstils vorgeschrieben werden. Durch diese Variation in Qualität und Quantität der Texte ergeben sich daher Probleme bei der Klassifizierung mittels Lernalgorithmen. Um die Sentimentanalyse zu verfeinern müsste man die Attributwahl etwas komplexer gestalten. Negierende Ausdrücke sollten vermerkt und gegebenenfalls aufgelöst werden. Ein weiteres Problem, das beim Sichten der Texte deutlich wird, ist, dass viele Nutzer in ihren Reviewtexten nicht über das eigentliche Produkt schreiben, sondern beispielsweise von ihren Erfahrungen mit ähnlichen Produkten oder den Lieferumständen berichten.
Um diesem Umstand gerecht zu werden, wäre es nötig, ein Themenattribut einführen, das beispielsweise nur Textabschnitte wertet, welche auch tatsächlich auf das zu bewertende Produkt referieren. Auch die allgemeine Textstruktur könnte für das Sentiment eines Textes ausschlaggebend sein.

7 Literaturverzeichnis