Quantitative Methoden in der Computerlinguistik

Übung 6: Plan of attack

Alena Zwahlen:

Selena Calleri:

Petra Wittwer: 11-914-488

## Feature Engineering

In erster Linie werden wir versuchen uns beim Feature Engineering auf zeichenbasierte n-Gramme fokussieren, da dieser Ansatz allgemein bei Spracherkennung sehr erfolgreich ist. Zusätzlich zu den n-Grammen werden wir versuchen bestimmte „Kennworte“ zu identifizieren, deren Vorkommen ein besonders starker Indikator für eine bestimmte Klasse ist. Beispielsweise kann es sein, dass in den Daten das Wort „nei“ fast nur in Samples für die Klasse BE vorkommt. Um diese Auszuwerten fügen wir pro Kennwort eine Dimension hinzu, deren Wert 1 ist falls das Wort vorkommt und 0 sonst. Um zu vermeiden, dass wir solche Worte von Hand identifizieren müssen, werden wir versuchen diese mit Hilfe des CALGARI Algorithmus[[1]](#footnote-1) zu berechnen. Weitere Verbesserungen erhoffen wir uns durch Bestimmung der Häufigkeit von Vokalen sowie der Einschränkung von n-Grammen auf solche, welche Vokale enthalten.

## Methodik

Wir werden möglichst viele, für die Problemstellung geeignete Klassifikatoren des scikit-learn Frameworks ausprobieren und durch Kreuzvalidierung vergleichen. Um für einen gegebenen Classifier geeignete Parameter und Features zu finden benutzen wir die im Framework enthaltene GridSearchCV Methode. Wir erhoffen uns dadurch das Risiko zu verkleinern, dass wir einen allfälligen guten Klassifikator verwerfen nur weil wir nicht durch Zufall die richtigen Parameter gefunden haben. Falls sich herausstellt, dass einige unserer Klassifikatoren gleich gut sind, aber keiner wirklich herausragt haben wir noch die Möglichkeit diese zu einem Voting-Classifier zusammenzufassen. Der entsprechende Code ist grösstenteils ebenfalls schon in scikit-learn enthalten.

Wahrscheinlich wird die Menge der der Features bei uns eher gross sein. Um allfälligen unnötigen Features entgegenzuwirken, werden wir versuchen nur eine Auswahl der zur Verfügung stehenden Features zu benutzen. Wir werden sowohl von Hand versuchen sinnvolle Subsets von Features zu finden, als auch mit Hilfe von scikit-learn’s „select\_k\_best“-Methode.

Falls wir genügend Zeit haben, werden wir als zweiten Hauptansatz noch versuchen, basierend auf den Features die sich als besonders vielversprechend Herausgestellt haben ein Neuronales Netz von Hand zu bauen. Als Framework können wir hierfür beispielsweise das auf Theano basierende Lasagne[[2]](#footnote-2) verwenden.

1. HECHT, Brent, et al. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, 2011. S. 237-246. [↑](#footnote-ref-1)
2. <https://lasagne.readthedocs.io/en/latest/> [↑](#footnote-ref-2)