Kolloquium, 29. Mai 2017

Dayyan Smith

Corpus based Identifcation of Text Segments

Vortragender: Thomas Ebert

Betreuer: Martin Schmitt

Die meisten NLP Anwendungen verarbeiten Texte als Sequenz von Worten. Da nicht eindeutig definiert ist was ein Wort ist, ist die automatische Tokenisierung sehr fehleranfällig, was lokale Anpassungen notwendig macht. Nun stellt sich die Frage, ob die Segmentierung eines Textes in Worte überhaupt die beste Art für einen Computer ist einen Text zu segmentieren. Das Ziel von Thomas' Bachelorarbeit ist die Entwicklung eines Algorithmus', der einen Eingabesatz in seine "beste" Segmente zerlegt. Dann wird geprüft ob der neue Ansatz besser ist als der klassische wortbasierte. Außerdem muss dann noch analysiert werden, welche Chancen und Risiken der neue Ansatz mit sich bringt.

Zunächst werden N-Gramme der N-Gramme der Längen 1 bis 10 aus dem Englischen Wikipedia Korpus extrahiert. Dafür werden die ersten 10.000 Texte verwendet. Nachdem eine Frequenzliste für die N-Gramme erstellt wurde, werden die N-Gramme mit einem Gütemaß (Gütemaß = n*log(freq), wobei n = N-Gramm-Länge, freq = Anzahl Vorkommen des N-Gramms) bewertet. Zum Testen wird ein Eingabesatz in die N-Gramme mit dem höchsten Gütemaß zerlegt.

Problematisch ist die Laufzeit, welche mit größer werdender Eingabe exponentiell steigt. Ein heuristischer Ansatz, bei dem die Größe des Fensters festgelegt wird, garantiert zwar nicht mehr die höchste Güte, aber die Segmentierung könnte immer noch besser sein als beim symbolischen Ansatz.

Auch die Evaluierung bringt Probleme mit sich. Die Granularität von Segmenten kann nicht immer objektiv bestimmt werden. Die Korrektheit von Segmentgrenzen kann bei Information Retrieval zwar vernachlässigt werden, bei news boundary detection sind Segmentgrenzen aber wichtig. Evaluiert wird die Auswirkung auf die Endanwendung: bei der Sentimentanalyse das Erkennen des richtigen Sentiments auf Satzebene. Das verwendete Korpus enthält Movie Reviews. Verglichen wird die Performance mit der Verwendung von character-n-ngram embeddings, die mit word2vec erstellt wurden mit normalen word embeddings.

Bisher ist aufgefallen, dass auch Buchstaben N-Gramme eine Zipfsche Verteilung aufweisen, dass die häufigsten N-Gramme, welche länger als 3 sind Funktionswörter enthalten und dass die häufigsten N-Gramme, welche länger als 8 sind Inhaltswörter erhalten.