Learning String Edit Distance

Der zweite Vortrag des Tages wurde von Anton Serjogin gehalten. Zu Beginn stellte er die String-Distanz vor. Die String-Distanz ist eine Art, die Unterschiedlichkeit von zwei Strings, in konkreten Zahlen auszudrücken und zu messen. Die String-Distanz wird zum Beispiel in „Natural language processing“ und in der Bioinformatik verwendet. Dann stellte er die Levenshtein-Distanz vor. Diese ist die am meisten verwendete Metrik, die bei der String-Edit-Distanz Verwendung finden. Es existieren drei wichtige Operationen. Erstens „Deletion“, zweitens „Insertion“ und drittens „Substitution“. Er führte einige Beispiele auf, die die Operationen näher erläuterten. Man kann Wörter minimal verändern, um ein anderes Wort zu erhalten. Bei einer Deletion-Operation wird zum Beispiel ein Buchstabe weggelassen. Aus dem Wort „remote“ entsteht so das Wort „emote“. Die Insertion-Operation macht im Grunde das Gegenteil und fügt Buchstaben ein. Aus dem Wort „lad“ wird zum Beispiel das Wort „glad“. Bei der Substitution verändert man einen Buchstaben und ersetzt ihn durch einen anderen. Aus dem Wort „carrot“ wird zum Beipiel das Wort „parrot“.

Bei dieser Transduktion wandelt man eine String, zum Beispiel ein Wort, von einem Zustand in einen anderen Zustand. Bei der String-Edit-Distanz werden diese Transduktionen ohne Speicherung durchgeführt und jede Transduktion erstellt entweder ein Deletion-Paar, ein Insertion-Paar oder ein Substitution-Paar. Wenn man dies als „stochastische“ Transduktion Interpretiert, erhält man zwei String-Distanzen. Einmal die Viterbi-Edit-Distanz und einmal die Stochastic-Edit-Distanz.

Wenn ein String-Paar viele mögliche Generierungspfade besitzt, dann wird die Viterbi-Edit-Distanz bevorzugt. Wichtig hierbei ist, dass die Levenshtein-Distanz für alle „Identity-Edit“ Operationen die Kosten auf null reduziert. Bei Viterbi und Stochastic sind aber nie null. Bei der Generierung der String-Paare ist es schwierig die Parameter zu bestimmen, da der Stochastische „Transducer“ ohne Speicherung arbeitet. Wichtig ist hier also die „Expectation Maximazation“.

Es gibt hier drei Varianten des speicherlosen stochastischen „Transducer“. Einmal „Parameter-Tying“, „Finit mixtures“ und einen stochastische„Transducer“ mit Speicher.

Weiterhin stellte er die String-Klassifizierung vor. Dafür wird ein Korpus, mit „gelabelten“ Strings benötigt. Die Bedingte Wahrscheinlichkeit wird mit der gemeinsamen Wahrscheinlichkeit, durch Anwendung des Bayes-Theorems erlangt. Diese Bedingte Wahrscheinlichkeit definiert mit einer utilitären Funktion den Klassifizierer.

Danach kam er auf die Application zu sprechen. Dies ist das Problem des Lernens der Aussprache von Wörtern. Dabei hat er das Switchboard Korpus verwendet. Für die Erkennung der Aussprache braucht man ein 6-Tupel. Darin ist ein Set aus Wörtern, ein Alphabet mit phonologischen Segmenten, ein Alphabet mit phonetischen Segmenten, ein Aussprache Lexikon, ein Trainingskorpus mit annotierten phonetischen Strings und ein Korpus mit nicht annotierten Alphabet mit phonologischen Segmenten. Heraus kommt ein Set aus Labels für den letzten Teil des Tupels.

Bei der Application gibt es fünf Experimente die auf die 7 Modelle angewendet werden. Jede Interpretation hat ein „Tied-Modell“, ein „Untied-Modell“ und ein „Mixture-Modell“. Bei der Evaluation fällt auf, dass die Levenshtein-Distanz besonders schlecht. Weiterhin stellt man fest, dass das Aussprache-Lexikon gute Ergebnisse liefert, wenn es direkt von tatsächlicher Aussprache erstellt wird.