Schriftliche Wiederholungsprüfung zur Vorlesung
Statistische Methoden in der maschinellen Sprachverarbeitung
SS 2020
Dozent: Helmut Schmid

Sie haben 60 Minuten Zeit plus 5 Minuten zum Absenden.

Allgemeiner Hinweis: Wenn Sie einen Fehler oder ein anderes Problem in einer der Aufgaben entdecken sollten, dann schicken Sie mir bitte eine Nachricht an `schmid@cis.lmu.de`.

Hinweis zu den Aufgaben 1 bis 5: Sie sollen hier keine allgemeine Formel angeben, sondern sagen, wie die Wahrscheinlichkeit in dem konkreten Fall berechnet wird.

Aufgabe 1) Wie berechnen Sie bei einem **Naive-Bayes-Modell** die gemeinsame Wahrscheinlichkeit der Klasse *Sport* und der Wörter *Bayern, schlägt, Manchester*? (2 Punkte)

Aufgabe 2) Wie berechnen Sie bei einem **Markowmodell** 2. Ordnung (Trigramm-Modell) über Buchstaben die Wahrscheinlichkeit der Buchstabenfolge *CIS*? (2 Punkte)

Aufgabe 3) Wie berechnen Sie bei einem **Hidden-Markowmodell** 1. Ordnung die gemeinsame Wahrscheinlichkeit der Wortfolge *Bayern, schlägt, Manchester* und der Tagfolge *NE, VVFIN, NE*? (2 Punkte)

Aufgabe 4) Wie berechnen Sie bei einem **Linear Chain CRF** die bedingte Wahrscheinlichkeit der Tagfolge *NE, VVFIN, NE* gegeben die Wortfolge *Bayern, schlägt, Manchester*, wenn das CRF-Modell nur Tag-Tag-Paare und Wort-Tag-Paare als Merkmale verwendet?

Sie können w(NE,VVFIN) für das Gewicht des Merkmales des Tagpaares (NE, VVFIN) schreiben und w(Bayern,NE) für das Gewicht des Merkmales des Wort-Tag-Paares (Bayern,NE).

Erklären Sie außerdem, wie hier die Normalisierungskonstante Z berechnet wird. (Sie müssen hier keinen genauen Ausdruck angeben.) (4 Punkte)

Aufgabe 5) Wie berechnen Sie mit **interpoliertem Backoff** die geglättete Wahrscheinlichkeit p(Manchester | Bayern, schlägt) aus den Häufigkeiten der Wort-Unigramme (bspw. *f(Manchester)*), -Bigramme (bspw. *f(schlägt, Manchester)*) und -Trigramme (bspw. *f(Bayern, schlägt, Manchester)*), sowie den Discount-Werten (δ_1 etc.) und den Backoff-Faktoren (α(Bayern, schlägt) etc.)? Bei den Unigrammen sollen die Wahrscheinlichkeiten nicht mehr geglättet werden. (Sie können statt α auch alpha schreiben, falls Sie eine Textdatei erstellen.) (3 Punkte)

Aufgabe 6) Erklären Sie in Worten, wie die **Discount**-Werte δ_i und die **Backoff-Faktoren** α(...) bei der interpolierten Backoff-Glättung berechnet werden. (3 Punkte)

Aufgabe 7) Welches Problem gibt es bei der normalen Backoff-Glättung? Wie vermeidet das **Kneser-Ney**-Verfahren dieses Problem? Worin unterscheidet sich die Berechnung der Backoff-Verteilungen bei den beiden Verfahren? (2 Punkte)

Aufgabe 8) Wie **schätzen** Sie die ungeglättete Wahrscheinlichkeit $p(Haus|NN)$ aus den gemeinsamen Wort-Tag-Häufigkeiten $f(w,t)$ (wobei w ein Wort und t ein Tag ist)? (1 Punkt)

Aufgabe 9) Angenommen Sie vergleichen Ihren neu entwickelten Spam-Klassifizierer *Spammy* mit einem Baseline-Spam-Klassifizierer und erhalten folgende Ergebnisse:

Goldstandard	Baseline	Spammy	Häufigkeit
Spam	Spam	Spam	57
Spam	Spam	NoSpam	5
Spam	NoSpam	Spam	7
Spam	NoSpam	NoSpam	3
NoSpam	Spam	Spam	2
NoSpam	Spam	NoSpam	6
NoSpam	NoSpam	Spam	2
NoSpam	NoSpam	NoSpam	154

Hier gab es bspw. 57 Emails, die sowohl vom Baseline-Tagger als auch von Spammy korrekt als Spam klassifiziert wurden.

Sagen Sie so genau wie möglich, wie Sie hier mit dem **Vorzeichentest** berechnen, ob Spammy signifikant besser als der Baseline-Klassifikator ist. (3 Punkte)

Aufgabe 10) Mit dem **Forward-/Backward-Algorithmus** können Sie die Wahrscheinlichkeit jedes möglichen Tags bei jedem Eingabewort berechnen. Sie wissen dann beispielsweise, dass das Tag *NN* beim 3. Wort eines gegebenen Satzes die Wahrscheinlichkeit 0,47 hat.

Wie könnten Sie auf Basis dieser Wahrscheinlichkeiten (sinnvoll) eine **beste Tagfolge** für den Satz definieren? (Die Lösung kennen Sie nicht aus der Vorlesung.)

Bekommen Sie mit dieser Methode dieselbe Tagfolge wie mit dem Viterbi-Algorithmus? Versuchen Sie, Ihre Antwort zu begründen. (3 Punkte)

Aufgabe 11) Bei der Parser-Evaluierung verwendet man üblicherweise sowohl **Precision** als auch **Recall** der ausgegebenen Konstituenten, da jede dieser Metriken alleine ausgetrickst werden kann.

Angenommen das Startsymbol der Grammatik lautet S. Wie könnten Sie bei beliebigen Sätzen eine Precision von 1 erzielen, ohne die Sätze wirklich zu parsen? Wie müssen die ausgegebenen Parsebäume aussehen?

Wie könnten Sie bei beliebigen Sätzen einen Recall von 1 erzielen, ohne die Sätze wirklich zu parsen? Wie müssen die ausgegebenen Parsebäume aussehen? (3 Punkte)

Aufgabe 12) Welche Unterschiede gibt es zwischen **generativen** und **diskriminativen** Verfahren?

Welche generativen und welche diskriminativen Modelle kennen Sie aus der Vorlesung? (2 Punkte)

(30 Punkte insgesamt)

Viel Erfolg!