Übungsblatt 7, Abgabe Montag 5. Dezember 18 Uhr

Arbeiten Sie die Aufgaben aus Blatt 6 insbesondere die Fragen zum Paper von Greffenstette nach.

Aufgabe 1

(a) Fassen Sie folgende Ausführungen zum Brill Tagger in einigen Sätzen zusammen. Geben Sie insbesondere den Unterschied zwischen Regeltemplates und Regelinstanzen an.
Lesen Sie in das Originalpaper (auf der Webseite verlinkt) hinein und beurteilen Sie ob die Ausführungen das Wesentliche wiedergeben.

(b) Verwenden Sie wiederum Text cg28 aus dem Brownkorpus. Suchen Sie Beispiele, die Ihrer Ansicht nach in der ersten Runde nicht richtig getaggt würden (einfaches Tagging mit dem wahrscheinlichsten Tag). Welche Templates (siehe unten) könnten eingesetzt werden um für diese Beispiele doch den richtigen Tag einzufügen. Welche Regelinstanzen ergeben sich.

Part-of-Speech-Tagging Using Rules

Lexikon-Lookup oder morphologische Analyse führt oft zu Ambiguitäten: *a chair, to chair a session*.
Ziel von Tagging Algorithmen: Ambiguitätsauflösung durch kontextbasierte Methoden.

Textstatistik (alle Angaben nur „Hausnummern“, grobe Schätzer): 50-60% der Wörter haben einen eindeutigen Tag, 15-25% haben nur zwei Alternativen.
Einfaches Tagging mit dem häufigsten POS für ein Wort führt zu 75% Erfolg. Für Englisch berichten manche Papers auch bis zu 90%.
Dies ist unsere **Baseline**.
Zwei lokale Methoden wurden für Tagging erfolgreich zur Disambiguierung benutzt. Regelbasierte Constraints, linker/rechter Wortkontext oder statistische Methoden oft unter Nutzung von Hidden-Markov-Modellen.

Brills Tagger ist regelbasiert.
Er besitzt ein Lexikon und es wird zunächst angenommen, dass das Lexikon perfekt in dem Sinne ist, dass es alle Wörter des zu taggenden Dokuments enthält. Es kommen also keine unbekannten Wörter (unknown words, out of vocabulary words OVW – sind in Wirklichkeit je nach Korpus zwischen 3-5% abhängig vom Lexikon)

Lexikonstruktur: wortform – wahrscheinlichster Tag, [Array anderer möglicher Tags]
can - Vmod,[NS]

siehe oben für 50-60% der Token im Text (Zahl für Brown Corpus) ist das Array leer.

Brill's Algorithmus

Schritt 1: Tagge alle Wörter des Texts mit ihrem wahrscheinlichsten POS-Tag.

Schritt 2: Anwendung von Transformationen = kontextbasierte Regeln, die einen Worttag in einen neuen umschreiben. Constraint: die Transformation wird nur dann durchgeführt, wenn sie für das spezielle Wort gültig ist. Das heißt: neuer POS-Tag ist teil des Alternativenarrays im Lexikoneintrag für das Wort.
Transformationen werden sequentiell im Text von rechts nach links ausgeführt.

Bsp.: Englisch, ändere den Tag von Modalverb zu Nomen, wenn das Vorgängerwort ein Artikel ist.
Deutsch, ändere den Tag von Artikel zu Pronomen wenn das Vorgängertoken ein Nomen oder ein Komma ist.

The can rusted.
Wer ist der Mann, der kommt.

Die Regeln werden gemäß folgenden abstrakten Templates erzeugt.

In Prologschreibweise

alter(A,B,prevtag(C))
alter(A,B,nexttag(C))
alter(A,B,prev2tag(C))
alter(A,B,next2tag(C))

als Originalregeln aus dem Brillpaper

1. The preceding (following) word is tagged c.
2. The word two before (after) is tagged c.
3. One of the two preceding (following) words is tagged c
4. One of the three preceding (following) words is tagged c.
5. The preceding word is tagged c and the following word is tagged d.
6. The preceding (following) word is tagged c and the word two before (after) is tagged d.
7. The current word is (is not) capitalized.
8. The previous word is (is not) capitalized.

Diese Regeltemplates werden dann mit tatsächlichen Regeln gefüllt. Regelinstanzen wie die oben angegebenen Beispiele für Englisch und Deutsch.

Brill berichtet er kommt mit 500 Regelinstanzen auf 97% richtig getaggte laufende Wörter im Text (Token).

Natürlich wollte Brill nicht 500 Regeln aufschreiben. Darum hat er sie automatisch gemäß den acht Templates aus einem getaggten Korpus (= gold standard, z.B. manuell getaggtes Brownkorpus, abgeleitet). Diese Art des Lernens wird als Transformation Based Learning bezeichnet -> Klasse supervised learning. Warum? Weil das Trainingskorpus mit den Zielkategorien annotiert ist.

Regeln Lernen:

(1) Annotiere jedes Wort im gestrippten Korpus (d.h. die manuellen Tags des goldstandard wurden entfernt) mit dem wahrscheinlichsten POS-Tag
(2) Vergleiche paarweise diese erste Version des Taggings mit dem goldstandard. Damit haben wir alle Fehler!
(3) Für jeden Fehler: instantiere die Regeltemplates, die den Fehler korrigieren würden. Also für das Beispiel the/ART can/Vmod rusted/V, können wir template-1 alter(A,B,prevtag(C)) mit A=Vmod, B=NS und C=ART instantieren und bekommen die oben angegebene Regel. Das sind die potentiellen Regeln, aber nicht alle sind gute Kandidaten.
(4) Für jede so erzeugte Regel: Berechne ihre Auswirkung auf den getaggten Korpus im Vergleich zum goldStandard: wie viele Fehler werden beseitigt und wie viele Fehler werden neu eingeführt = Nettoeffekt der Regel.
(5) Wähle diejenige Regel aus die den größten Nettoeffekt hat. Hänge die Regel an die geordnete Liste der Transformationsregeln an (am Anfang ist die Liste leer...dann erster Durchlauf usw.)
(6) Wende die Regel auf das aktuelle Korpus an für die i-te Wiederholung das Korpus der vorhergehenden Runde also Korpus(i-1). Daraus bekommen wir das Korpus das der Input für die nächste Runde wird Korpus(i).
(7) Falls die Fehlerrate Korpus(i) in Vergleich zum goldstandard noch größer als eine bestimmte Grenze ist, gehe zu Schritt 3 und wiederhole das Verfahren – solange bis entweder die Fehlerrate klein genug ist oder aber keine Regeln mehr mit positivem Nettoeffekt gebildet werden können.

Produktivste Regel bei Brill: Nomen nach Verb wenn das Vorgängertoken „to“ ist.

Methode zur Behandlung von OVW bei Brill, Kombination aus häufigstem Tag und der Suffixmethode.

Aufgabe 2

(a) Lesen Sie das Paper „Parsing by Chunks“ von Steven Abney. Fassen Sie zusammen, was Chunk Parsing ist. Geben Sie an wo Chunk Parsing erfolgreich eingesetzt werden könnte. Klären Sie den Begriff „major head“ . Nehmen Sie dazu Stellung: ein klares Problem stellt dar, dass u.a. Verb-Objekt Relationen nicht abgebildet werden. Warum sind diese so wichtig?

(b) Wie sieht der Regelapparat für so einen Chunkparser aus?

(c) Verwenden Sie wiederum Text cg28 aus dem Brownkorpus und geben Sie Beispiele für Chunk Strukturen an.

Aufgabe 3

Schauen Sie sich das auf der Webseite verlinkte Video von Chomskys Vortrag in London an (zumindest einige Minuten). Versuchen Sie unter Benutzung auch der Diskussion auf der Linguist List (Link ist auf der Webseite) herauszufinden, worum es in etwa geht.
Haben Sie eine eigene Meinung zu diesem Thema?