Notizen zum Seminar "Parallele Korpora": Automatische Verfahren zur Wortalinierung

Sebastian Pado

1. Dezember 2010

Notizen zur 3. Sitzung des Seminars "Parallele Korpora". Ohne Gewähr.

1 Klassifikationsverfahren für Wortalinierung

- Wie können wir Worte in parallelen Sätzen alinieren?
- Verwende ein Lexikon
  - Es gibt oft kleine, manuell erstelle Lexika
  - Dies gilt insbesondere für Texte in bestimmten Domänen, die in der Praxis häufig vorkommen.
  - Beispiel: KNOWA (Pianta und Bentivogli 2004)

Herausforderung: rein lexikonbasierte Verfahren machen viele Fehler. Wieso?

- Mehrwortausdrücke: MWEs müssen kollabiert und als ein Token behandelt werden
- Übersetzungen für Funktionswörter stehen oft nicht im Lexikon
- Wenn Wörter zweimal in einem Satz vorkommen, sind beide Übersetzungen "gleich gut"
- KNOWA verwendet eine Reihe von Heuristiken, um die Alinierung zu optimieren.

Es gibt zwei große Erweiterungen der Lexikonidee: assoziationsbasierte (heuristische) Verfahren und probabilistische (kookkurenzbasierte) Verfahren.

2 Assoziationsbasierte Alinierung

Assoziationsbasierte Alinierung verzichtet auf das Lexikon und versucht dafür, alle möglichen Hinweise auf mögliche Alinierungen zu sammeln und zu kombinieren. Die Idee ist, sich zu überlegen, unter welchen Bedingungen es wahrscheinlicher oder unwahrscheinlicher wird, dass zwei Wörter (s, t) aliniert sind:

- Wörter mit der gleichen Wortart werden tendentiell häufiger aliniert
- Wörter mit dem gleichen Phrasentyp werden tendentiell häufiger aliniert
- Wörter mit der gleichen Position im Satz werden tendentiell häufiger aliniert
- Wörter, die häufig gemeinsam in alinierten Sätzen vorkommen, werden tendentiell häufiger aliniert
- Wörter, die einander ähnlich sehen, werden tendentiell häufiger aliniert

Diese Modelle bezeichnet man auch als "clue alignments" (hinweisbasierte Alinierungen), weil man Hinweise auf Alignments kombiniert. Für Details siehe Tiedemann (2003).

Die Modelle bestehen dann aus zwei Komponenten: die erste Komponente berechnet Alignment-Wahrscheinlichkeiten für jedes Paar von Wörter aus den Hinweisen. Das Ergebnis ist eine reellwertige $S \times T$-Matrix. Die zweite Komponente versucht dann, aus dieser Matrix Alignments abzulesen (Optimierungsschritt).

Berechnung von Alignment-Wahrscheinlichkeiten. Einer der wichtigsten Hinweise ist in der Tat das gemeinsame Vorkommen von Wörtern in Sätzen, das mit dem Dice-Koeffizienten gemessen wird:

$$A_{dice}(s,t) = \frac{2|\{S \mid S \text{ enthält } s \text{ und } t\}|}{|\{S \mid S \text{ enthält } s\}| + |\{S \mid S \text{ enthält } t\}|}$$

ausserdem String-Ähnlichkeit, $A_{str}(s,t)$, die zB über die längste gemeinsame Zeichenkette (longest common subsequence) gemessen werden kann.

Die anderen Hinweise (gemeinsame Wortarten, gemeinsamer Phrasentyp etc.) verlangen doch zumindest eine geringe Menge an alinierten Daten. Technisch funktioniert das so, dass man eine Feature-Funktion definiert, die z.B. ein Wort auf seine Wortart abbildet. Dann:

$$A_{feat}(s,t) = \frac{2freq(feat(s), feat(t))}{freq(feat(s)) + freq(feat(t))}$$

wobei $freq$ gezählt wird auf den bereits alinierten Daten.

Als nächstes müssen die A-Scores in Wahrscheinlichkeiten umgerechnet werden; dazu wird einfach normalisiert. Zum Schluss werden die A-Scores kombiniert, und zwar per Disjunktion: es wird angenommen, dass ein starker Hinweis genügt, um ein Alignment wahrscheinlich zu machen. Details siehe Tiedemann (2003).

Extraktion von Alinierungen. Wie gesagt, das Ergebnis des 1. Schrittes ist eine reellwertige $S \times T$-Matrix. Jetzt ist die Frage, wie man aus dieser Matrix ein Alignment extrahiert. Eine einfache Möglichkeit ist, anzunehmen, dass das Alignment eine Funktion ist. Das macht die Extraktion sehr einfach:

$$A(s) = argmax_t P(s,t)$$

Das heisst, man sucht einfach für jedes Eingabewort das wahrscheinlichste Ausgabewort nach den Assoziationswahrscheinlichkeiten. (Dies kann man natürlich auch wieder umgekehrt machen, um ein Alignment von T nach S zu bekommen.)

2.1 Probleme

- Die Auswahl der Hinweise ist arbiträr
- Das Verfahren ist halbüberwacht: je grösser das existierende alinierte Korpus ist, desto besser sind die Feature-Assoziationen A_{feat} abzuschätzen.
- Die Kombination nimmt an, dass ein starkes Feature reicht: schlechte Features können also das ganze Modell zerstören
- Die Assoziationsscores haben keine klare Interpretation: was optimieren wir hier eigentlich?

3 Estimationsbasierte Alinierung

Insbesondere das Problem "was wir hier eigentlich genau optimieren" wird durch estimationsbasierte Alinierung (EA) gelöst. EA-Modelle können als Erweiterung von assoziationsbasierten Modellen verstanden werden, die eine klare probabilistische Interpretation hat.

Die am weitesten verbreiteten EA-Modelle sind die sogenannten "IBM-Modelle" aus der maschinellen Übersetzung. Hier steht das Alignment eigentlich gar nicht selbst im Mittelpunkt, sondern dient nur dazu, zwischen den zwei Sätzen zu mediieren.

Die IBM-Modelle sind "generative Modelle" – das heisst probabilistische Modelle, bei denen wir eine Theorie darüber haben, in welcher Richtung die Kausalität verläuft. Die IBM-Modelle nehmen alle an, dass wir von einer fremden Sprache F ins Englische E übersetzen. Die Theorie ist nun, dass:

- der fremdsprachliche Satz F gegeben ist
- F zuerst eine Alinierungsfunktion $A : F \rightarrow E$ erzeugt
- Dann F und A zusammen einen englischen Satz E erzeugen

Diese Generierungsprozesse haben Parameter, und mithilfe dieser Parameter lässt sich die Wahrscheinlichkeit von (F, A, E)-Paaren angeben. Wenn wir die Parameter so optimieren, dass den Daten, die wir vor uns sehen, eine möglichst hohe Wahrscheinlichkeit

zugewiesen wird, dann haben wir vermutlich "vernünftige" Parameter gefunden. Damit ist das Ziel klar.

Es gibt eine ganze Sequenz von IBM-Modellen: Modell 1 bis 6 (5 und 6 werden selten verwendet). Die einzelnen Modelle unterscheiden sich dadurch, welche spezifischen Annahmen sie über den generativen Prozess machen (von wenigen zu vielen).

Modell 1. Modell 1 ist ein "lexikalisches Modell":

$$P(E, A|F) \propto \prod_{i=1}^{|F|} tr(E_{A(i)}|F(i))$$

($\propto$ steht für "proportional zu": wir lassen hier einfach eine Normalisierungskonstante verschwinden.)

Was passiert hier? Für jedes fremdsprachliche Wort wird das englische alinierte englische Wort gesucht, und die lexikalischen Übersetzungswahrscheinlichkeiten werden aufmultipliziert. Das beste Alignment für ein gegebenes Satzpaar kann dann gefunden werden als:

$$\hat{A}(E, F) = argmax_A P(E, A|F)$$

Es ist aufschlussreich, sich klarzumachen, was dieses Modell *nicht* kann:

- es kann keine Alinierungen präferieren, die "lokal" sind (also würde es denselben Fehler für den dog/cane-Satz aus Abschnitt 3 machen wie KNOWA)
- es kann nur n-zu-1-Alinierungen erzeugen.
- es berücksichtigt auch keine englische Wortstellung (dafür ist in den Übersetzungsmodellen ja das Sprachmodell da).

Modell 2 Modell 2 erweitert Modell 1 (das lexikalische Modell L) um eine Komponente, die die "Abweichung" eines Wortes $E_A i$ von der Position des Originalwortes F_i zu bewerten versucht (reordering component R):

$$P(E, A|F) \propto \prod_{i=1}^{|F|} tr(E_{A(i)}|F(i))P(A(j)|j) = L \cdot R$$

Dieses Modell lernt (hoffentlich), dass $P(5|5) > P(1|5)$, d.h. dass das 5. Wort des Eingabesatzes wahrscheinlicher das 5. Wort des Zielsatzes wird als das 1. Wort des Zielsatzes. Leider ist dies nur eine sehr grobe Annäherung, weil diese Wahrscheinlichkeiten für jede Position im Satz neu gelernt werden müssen.

Modell 3 Modell 3 erweitert Modell 2 um eine "fertility component" F: d.h., die Möglichkeit, dass Quellwörter mehr als ein Zielwort erzeugen. Damit haben wir hier das erste Modell, das nicht mehr annimmt, dass A eine Funktion ist. Das macht das Modell aber auch deutlich komplexer, ich zeige hier nur die Struktur:

$$P(E, A|F) \propto L \cdot R \cdot F$$

F sagt für jede lexikalische Einheit separat (!) voraus, in wieviele Worte sie übersetzt werden soll.

Modell 4 Modell 4 schliesslich führt lokale Umordnung ein, also Wahrscheinlichkeiten vom Typ "wie wahrscheinlich ist es, dass Wörter um n Positionen wandern".

3.1 Estimierung

Woher bekommen wir nun die Parameter dieser Modelle? Es zeigt sind, dass die Parameter durch unüberwachte Verfahren induziert werden können mithilfe des EM (Expectation Maximization)-Algorithmus. Die Idee ist dabei, dass man immer zwischen (gewichteten) Alinierungen und Modellparametern hin- und herspringen kann.

Besonders einfach ist das für das Modell 1, wo die Modellparameter nur aus den Übersetzungswahrscheinlichkeiten tr bestehen.

- Initialisierung: aliniere alle Wörter in F in allen Wörtern in E. Annahme: alle diese Alinierungen sind gleich wahrscheinlich.
- **Expectation**-Schritt (vereinfacht): zähle die Alinierungs-Links aus und estimiere $tr(E_i|F_j)$ als die Anzahl der Links von F_j zu E_i geteilt durch die Anzahl aller Links, die von F_j ausgehen.
- **Maximization**-Schritt: verwende die neuen Parameter tr, um an jeden Alinierungs-Link ein Gewicht (nämlich die entsprechende Wahrscheinlichkeit laut tr) zu schreiben
- **Expectation**-Schritt: zähle die Alinierungs-Links aus und re-estimiere $tr(E_i|F_j)$. Da die Links jetzt mit einer Wahrscheinlichkeit assoziiert sind, gilt:

  $$tr(E_i|F_j) = \frac{\sum_{E_i,F_j \in (E,F)} tr(E_i|F_j)}{\sum_{F_j \in (E,F)} \sum_{E_k \in (E,F)} tr(E_k|F_j)}$$

  In Zähler betrachten wir alle Satzpaare, in denen E_i und F_j vorkommen und summieren die Wahrscheinlichkeiten, die wir von diesen Links bekommen. Im Nenner berechnen wir die Summe *aller* ausgehenden Alinierungskanten aller Vorkommen von von F_j.

  Die Formel ist so zu verstehen, dass wir den neuen Wert von tr (vor dem Gleichheitszeichen) definieren durch die Werte von tr aus der letzten Iteration (nach dem Gleichheitszeichen).

- **Maximization**-Schritt: wie oben
- Wiederhole bis zur Konvergenz.

Details dazu in Folien von Alexander Fraser, Philip Köhn, oder in Och und Ney (2003).

3.2 Würdigung und Probleme

Die IBM-Modelle sind die am meisten verwendeten Modelle für Wortalinierung - also ein de facto-Standard. Dadurch, dass sie unüberwacht induziert werden können, sind sie auch für neue Sprachpaare einfach zu bauen. Die komplexeren IBM-Modelle beinhalten mehrere Komponenten, durch die sie verschiedene Eigenschaften potentieller Alinierungen gegeneinander "abwägen" können. Im Gegensatz zu den assoziationsbasierte Alinierungsverfahren passiert dies aber nicht "von Hand", oder wird nicht hart codiert wie in lexikonbasierten Verfahren, sondern die Daten bestimmen selbst, welche Eigenschaften der Alinierungen die beste Vorhersage für die aktuellen Daten erlauben. Für Sprachpaaren, in denen die optimale Alinierung fast immer 1-1 ist und die Wortstellung gleich bleibt, werden zB die Verteilungen F und R viel Gewicht bekommen; für freie Übersetzungen in Sprachen mit Scrambling wird vermutlich das Gewicht von L überwiegen etc.

Der zentrale Punkt bei den IBM-Modellen ist, dass sämtliche Parameter aus Kookkurenzstatistiken abgeleitet werden. Das führt zu folgenden allgemeinen Problemen.

- Parameterschätzungen für seltene Kookkurenzen (und seltene Ereignisse allgemein) sind unzuverlässig. Beispiele:
  - Lexikalische Wahrscheinlichkeiten für seltene Wörter
  - Fertitity-Wahrscheinlichkeiten für MWEs (notorisch unzuverlässig!)
  - Reordering für lange Sätze, deren hohe Positionen selten gesehen wurden
- Korrelierte Ereignisse (mehrteilige Eigennamen, Idiome, Phrasen etc.) können nicht auseinandergehalten werden.
  - Beispiel: "British Columbia/Colombie britannique" ist in den Canadian Hansards aliniert als "British/Colombie" und "Columbia/Britannique". Wieso?
  - In den Hansards ist sehr viel öfter von BC die Rede als entweder von Grossbritannien oder Kolumbien. Daher kommen die zwei MWEs häufig gemeinsam in Satzpaaren vor. Daher:
    - Auf der Ebene der lexikalischen Wahrscheinlichkeiten gilt daher: $P(British|Colombie) \approx P(British|Britannique)$ und $P(Columbia|Colombie) \approx P(Columbia|Britannique)$.
    - Auf der Ebene der Reordering-Wahrscheinlichkeiten gilt vermutlich, dass P(paralleles Alignment)>P(kreuzendes Alignment), weil die Wortstellungen von Französisch und Englisch oft parallel sind.

– Modell 1 kümmert sich nicht um R; daher ist die Wahl nur von L abhängig und in diesem Fall weitgehend dem Zufall überlassen
– Modelle 2ff. wägen L und R gegeneinander ab, und da L keine klare Präferenz hat, R aber schon, wird das parallele Alignment gewählt.
– Ähnliche Probleme ergeben sich im Deutschen häufig mit Funktionsverbgefügen.