Word-sense Disambiguation – Überblick

Christian Wurm (Düsseldorf)

Düsseldorf

WSD – word sense disambiguation

Einige Fakten zum Anfang (nach Agirre& Edmonds):

- Allgemein gilt: je häufiger ein Wort, desto mehr Lesarten hat es (vielleicht neutraler:

$$\text{Häufigkeit} \sim \text{Anzahl Lesarten}$$

- Die häufigsten 121 Englischen Worte (Types) machen etwa 1/5 der Wortvorkommen aus (token) – Denke an Zipf:

$$f(w) \propto r(w)^{-1}$$

- Die Häufigsten 121 Englischen Worte haben im Schnitt 7.8 Lesarten!
- Daraus folgt, dass ein durchschnittlicher englischer Satz (ca. 20 Worte, geschrieben) im Durchschnitt mehr als 7.8^4 rein lexikalische Lesarten hat, und höchstwahrscheinlich viele mehr (exponentielles Wachstum)!

WSD – word sense disambiguation

- Die allermeisten dieser Lesarten laufen unbemerkt unterm Radar – für uns Menschen!
- Für Maschinen wird lexikalische Disambiguierung oft als **KI-Vollständig** beschrieben – das bedeutet, sie werden es so gut machen wie wir, wenn sie allgemein (mindestens) so intelligent sind wie wir
- Ich glaube aber nicht dass das stimmt...

- WSD hängt auch stark an verschiedenen Aufgabenstellung: es ist überhaupt nicht apriori klar, was eine Wortbedeutung ist! Das kann sein

  1. Ein definierter Knoten in einer Wissenbasis (knowledge graph)
  2. Eine linguistisch motivierte Einheit
  3. Eine mögliche Übersetzung (im Kontext maschinellen Übersetzens)
  4. ...

Je nach Ansatz haben wir unterschiedliche Definitionen von Lesart, und das Problem stellt sich anders...

WSD und lexikalische Semantik

Auch wenn die beiden im Prinzip zusammen gehöhren:
Lexikalische Semantik hat sich immer wenig um WSD gekümmert – und umgekehrt!

- Ernste Modelle lexikalischer Semantik waren immer zu komplex, um sie effektiv für WSD zu nutzen
- Semantiker haben WSD-Methoden immer als theoretisch irrelevant betrachtet
- Angewandte Computerlinguisten sind dagegen normalerweise eher an "schnellen Ergebnissen" interessiert
- Ergebnis: man hält sich voneinander fern!

WSD in der Computerlinguistik

WSD ist ein altes und aktuelles Thema in der Computerlinguistik ($>$ 700 Artikel in ACL Anthology, Stand 2007!)

Aber: es gibt bis heute wenig Evidenz das separate, explizite WSD wirklich wichtig ist für NLP Applikationen!
Das führt uns zu einer wichtigen Unterscheidung:

- Explizites WSD: hier gibt es eine klar definierte Aufgabe, Lesarten zu finden. Systeme werden auch explizit evaluiert
- Integriertes/implizites WSD: Hier wird WSD als Teil einer größeren Aufgabe durchgeführt: Maschinelle Übersetzung, Semantisches Parsing etc. Die Disambiguierung ist hier natürlich implizit – aber sie wird nicht separat trainiert oder evaluiert!

Implizites WSD: Beispiel maschinelle Übersetzung

Tatsächlich gibt es noch wenig Evidenz, dass explizites WSD als separate Komponente wichtig ist für andere Systeme!

Tatsache ist, dass implizite Systeme genau "nach Bedarf" disambiguieren!

Implizites WSD: Beispiel maschinelle Übersetzung

Moderne Übersetzungssysteme haben meisten eine zweigeteilte Architektur:

1. Das Übersetzungsmodell (besagt wie nah ein Satz an der Ausgangssprache ist)
2. Das Sprachmodell (besagt, wie gut ein Satz in der Zielsprache ist)

Natürlich haben beide Modelle oft widerstreitende Interessen:

- Sprachmodelle bevorzugen kurze Sätze mit häufigen Worten (immer und sowieso)
- Übersetzungsmodelle bevozugen spezifische Übersetzungen

Explizites WSD

Das Zusammenspiel der beiden Modelle führt dazu, dass z.B. `plant` übersetzt wird mit `Pflanze` – aber an keiner Stelle wird die Eingabe `plant` wirklich disambiguiert!

(In diesem Fall fällt wohl die Hauptlast auf das Sprachmodell, aber was die Übersetzung macht ist immer etwas obskur)

Explizites WSD

Explizites WSD ist also nicht ganz unproblematisch (es gibt kaum Anwendungen hierfür)! Dennoch werden wir hier nur explizites WSD betrachten.

Es gibt einige allgemeine Einsichten aus WSD:

- Lokale Kontexte sind informativer als globale
- Nomen brauchen eher topikale Merkmale,
- Verben brauchen eher syntaktische Merkmale (Verben sind ja syntaktisch reicher)
- Es gibt eine allgemeine Methode der Evaluation von WSD: Senseval

Unsupervised

- Aber: Methoden für WSD sind so verschieden wie es nur geht. Daher unterteilen wir nur grob in:
  1. Unsupervised
  2. Knowledge-based
  3. Supervised

Unsupervised

Bedeutet: wir haben als **Eingabe**

- Die Worte mit ihren Lesarten
- und rohen Text

Die **Ausgabe** ist ein Algorithmus, der für beliebige Eingabetexte die entsprechenden Worte nach ihren Lesarten klassifiziert.

Unsupervised bezieht sich hier also auf die Trainingsdaten, die nicht annotiert sind. Die Lesarten sind normalerweise vorgegeben!

Supervised

Bedeutet: wir haben als **Eingabe**

- Die Worte mit ihren Lesarten
- und Text, der nach Lesarten annotiert ist.

Die **Ausgabe** ist ein Algorithmus, der für beliebige Eingabetexte die entsprechenden Worte nach ihren Lesarten klassifiziert.

Das ist relativ trivial, das Problem ist, dass solche Daten nur für wenige Sprachen existieren!

Knowledge-based

Bedeutet: wir haben als **Eingabe**

- Die Worte mit ihren Lesarten
- und eine Wissensbasis, in der die Lesarten unereinander Relationen haben

Die **Ausgabe** ist ein Algorithmus, der für beliebige Eingabetexte die entsprechenden Worte nach ihren Lesarten klassifiziert.
Das ist soz. semi-supervised: wir haben von Hand erstellte Daten – aber es ist nicht die Art von Daten, die wir für unseren Algorithmus brauchen!
Wir müssen also den Transfer leisten!

Senseval

Entscheidend für WSD war, dass es eine einheitliche Methode der Evaluation gibt:

- Die Ergebnisse verschiedener Methoden bei verschiedenen Aufgaben sind nur sehr bedingt zu vergleichen!
- Daher gibt es auch keinen *state-of-the-art*, keinen Vergleich für Modelle
- Senseval (Kilgarrif 2000) leistet das. Ein großer Schritt für WSD!

Senseval ist dabei einfach ein standardisierter Datensatz, auf dem standardisierte WSD-Aufgaben durchgeführt werden können (auf verschiedenen Sprachen).

Das ist nichts großes, aber es zeigt klar auf, welche Methoden besser und welche schlechter sind!

Senseval

Nach einheitlicher Meinung der Beteiligten hat WSD (supervised!) ein Plateau erreicht:

- Die Akkuratheit von WSD-Systemen unterscheidet sich nicht signifikant von dem sog. Inter-annotator agreement (Annotatoren sind sich oft uneins über Dinge)
- Das liefert starke Evidenz dafür, dass das Maximum erreicht ist:
- Besser als ein menschlicher Annotator kann ein WSD-System per Definition nicht werden!

Senseval

Die Aufgaben bestehen in anderen Problemen:

- Unsupervised WSD (ohne annotierte Trainingsdaten). Das ist insbesondere wichtig für Sprachen mit wenigen Ressourcen (also praktisch alle bis auf 4)
- Unsupervised WSI (Induction): können die verschiedenen Lesarten rein auf der Basis von Korpora induzieren?
- ...

Senseval

- Senseval basiert auf WordNet, eine Graph-basierte *meaning bank*
- (Geordnete) Graphen sind Paare (E, V), $E \subseteq V^2$
- Knoten in wordnet Repräsentieren "atomare Bedeutungen", oder "Bedeutungsatome"
  Also: Ein normales Wort bekommt viele Knoten in WordNet zugewiesen
- Kanten reäsentieren **Relationen von Bedeutungen**:
  - Hyponym
  - Hyperonym
  - Antonym
  - ...

WordNet: Was sind atomare Bedeutungen?

Hier sind wir wieder bei der Ausgangsfrage: was ist eine (unambige) Bedeutung? Im Kontext von WSD wird das zu:

Was ist ein Wortsinn?

Das ist eine schwierige Frage. Senseval nutzt WordNet, aber:

WordNet kümmert sich nicht um "unsere" Kriterien für Ambiguität (Konvexität, universelle Distribution, Koprädikation),

WordNet möchte einfach möglichst fein aufdröseln (wie wir sehen werden!)

Ein Beispiel

Nimm die unvermeidliche `Bank`, mit den Bedeutungen:

1. Sitzbank
2. Geldbank

Auf den zweiten Blick sieht man jedoch, dass auch die Geldbank ambig ist:

2a Das Gebäude

2b Die abstrakte Institution

2c Geldschalter

2d Eine Partei im Kasino

2e ...

Sind das noch echte "Wortbedeutungen"?

Was ist also ein Wortsinn?

Man kann sich das von vielen Perspektiven Fragen:

- Theoretische Linguistik
- Praktische Lexikographie
- Psycholinguistik
- oder eben WSD!

Die WSD-community hat (nach eigenen Angaben) die anderen Felder sehr erfolgreich ignoriert.

WSI - word sense inventory

Was man konstruieren muss ist ein **word-sense inventory**, ein Inventar von Wortbedeutungen.
Dann kann man jedem Wort seine Bedeutungen zuweisen.
Die Grundsätze vonn WSI sind:

1. Clarity
2. consistency
3. complete coverage

Es wurden um das Jahr 2000 verschiedene Inventare erstellt:

- LDOCE
- Roget's international thesaurus
- Hector
- WordNet

WordNet ist sicher das größte und bekannteste.

Anwendungen von WSD

Die Anwendungen betreffen folgende Felder (u.a.):

1. Maschinelle Übersetzung
2. Information retrieval
3. Information extraction/text mining
4. (Computationelle Lexikographie)

Maschinelle Übersetzung

Es ist klar, das MÜ WSD benötigt, z.B.

$$\text{change} \mapsto \text{Wechselgeld/Wandel}$$

Irgendwie muss man das machen. Aber:

- WSD ist in allen mir bekannten Fällen *implizit*: es wird irgendwo gemacht, zwischen Sprachmodell und Übersetzungsmodell
- Aber es gibt eben kein WSD-Modul!
- Interessant: wenn man ein explizites WSD-Modul vorschaltet (für die Eingabe), geht die Performance runter!

Information Retrieval

Information retrieval besteht darin, dass wir aus einer großen Basis von Wissen das für uns relevante mit einer Abfrage (Query) herausholen wollen. Wenn queries in natürlicher Sprache gemacht werden (siehe google), dann müssen wir natürlich disambiguieren:

Große Depression

(Wirtschaft, Psychologie, Wetter...)
Benutzt google explizites WSD? Gute Frage!!
Andererseits: Datenbanken sind oft Domänenspezifisch, und das macht WSD meist überflssig...

Information extraction

IE ist etwas komplexer: hier geht es darum, die Information wirklich zu extrahieren (also eine Art Zusammenfassung). Auch hier wichtig:

WSD ist entscheidend für die Relevanz von Worten

Mg – Milligramm, Magnesium

Das steckt aber noch in den Kinderschuhen....

Lexikographie

Das ist natürlich ein Klassiker! Aber:

- moderne Lexikographie ist korpus-basiert, und natürlich wäre es schön, wenn das Korpus (für arbeitszwecke) bereits disambiguiert wäre!

Natürlich geht das aber in beide Richtungen:

- Lexikographie gibt auch wichtige Eingaben für WSD!

Gemeinsamkeiten

Alle WSD Systeme disambiguieren ein Wort auf Basis von Kontext.
Das ist ein sehr weiter Begriff, er umfasst:

- Globaler Kontext (Thema des Textes)
- Lokaler Kontext (umgebende Worte)
- syntaktischer Kontext (als Ergebnis von POS-tagging)
- semantischer Kontext (basieren auf Lemmas)

Allgemein ist der Merkmalsraum sehr *hochdimensional*, man muss also auf viele Arten von Information achten (anders als beim POS-tagging)

Ansätze für WSD

Alle relevanten Systeme sind statistisch und basieren auf maschinellem Lernen.
Man unterscheidet daher 3 Arten von Ansätzen:

1. Supervised: basierend auf Sinn-annotierten Daten.
2. Unsupervised/semi-supervised: trainiert auf (fast) rohen Daten
3. Knowledge-based systems: basierend auf rohem Text *und* Wörterbüchern oder Wissensbasen
4. Kombinationen

Supervised WSD

Die Systeme funktionieren am besten und haben eine sehr gute Performance:

- 77% Akkuratheit für allgemeine Disambiguierung (eines Textes)
- Gegenüber 80% Inter-Annotator Agreement!
- (WordNet hat sehr fein granulierte Bedeutungen)

Das Problem ist: ausreichend annotierte Daten stehen nur in sehr wenigen Sprachen zur Verfügung!
Aber für diese Sprachen kann man sicher sagen: WSD ist gelöst!

Semi-supervised

Das umfasst solche Methoden wie *bootstrapping*:

Bootstrapping

Aus einem sehr kleinen vorgegebenen Datensatz wird sukzessive ein größerer erzeugt. Das Modell generiert sich also selber neue Daten, auf denen es in jeder Runde besser wird!

Bootstrapping-basierte Methoden erreichten bereits 1995 accuracy von 95% – evaluiert nur auf 12 Worten (vor Senseval!).

Unsupervised

Das ist sicherlich am schwächsten:

- wir haben 66% accuracy,
- gegenüber 46% accuracy für den Klassifikator, der immer den häufigsten Wortsinn wählt (majority baseline)
- (und 16% random baseline)

Das ist sicher ein *hot-topic*: hier kann man noch viel machen!

Knowledge-based

Wissensbasierte Systeme haben eine gute Performance.
Sie sind das Mittel der Wahl, falls keine annotierten Daten zur Verfügung stehen.
Wissensbasen gibt es Multilingual: BabelNet!

Kombinationen

Ein interessanter kombinatorischer Ansatz ist folgender:

- Wir nehmen ein bilinguales, wortaliniertes Korpus (das generiert man bei maschineller Übersetzung)
- Die Alinierung der Worte sollte eine Idee der verschiedenen Lesarten geben.
- Hierauf kann man nun bootstrapping oder supervised Methoden applizieren!

WSD – also?

Zusammenfassend kann man sagen:

- WSD ist eines der ältesten Probleme der CL
- Inzwischen ist das Problem gut verstanden und kann teilweise als gelöst betrachtet werden.
- Dennoch ist noch nicht ganz klar, wo die Rolle von WSD im natural language processing liegt!

Evtl. Lösung

WSD wird schon immer für seine separate Performance evaluiert.
Vielleicht ist das der Fehler:

$\Rightarrow$ Vielleicht muss man WSD-Systeme daraufhin evaluieren, wie gut sie anderen Systemen zuarbeiten!

Literatur

Edmonds, Agirre: Introduction, in: Eneko Agirre & Philip Edmonds: Word Sense Dismabiguation. 2007