Statistische Maschinelle Übersetzung

Teil I - Einführung

Thomas Schoenemann
Heinrich-Heine-Universität Düsseldorf

Sommersemester 2012

Zielsetzung

Grundproblem: gegeben ein Satz in einer Ausgangssprache.
$\Rightarrow$ übersetze (automatisch) in die gewünschte Zielsprache

Es gibt viele "richtige" Übersetzungen.

Fachausdruck: maschinelle Übersetzung (machine translation)

Gegenwärtiger Stand:

- Fehlerfreie maschinelle Übersetzung: nur bei kurzen Sätzen
- Sätze mit Fehlern können akzeptable sein (z.B. Interaktion Computer - menschlicher Posteditor).
- Gegenstand aktiver Forschung

Ansätze

1. Regelbasierte Ansätze
   - historisch gesehen zuerst
   - viele kommerzielle Produkte (z.B. Systran)
   - erfordert viel manuelle Eingaben
   - viel manueller Aufwand bei Wechsel des Sprachpaars
2. **Wahrscheinlichkeiten / Kostenfunktionen**
   - Basis: maschinelles Lernen
   - Manuelle Eingaben optional (Lexika etc.)
   - Sprachpaar leicht wechselbar

Probabilistische Übersetzung

Grundkonzept: Übersetzung durch maschinelles Lernen

1. Entwerfe ein allgemeines Modell (=Kostenfunktion) für die Übersetzung von einer beliebigen Sprache in eine beliebige andere.
   - muss nur einmal ("im Leben") gemacht werden.
   - trotzdem: Gegenstand aktiver Forschung
2. Das Modell hat generell eine Vielzahl von Parametern. Lerne diese aus bilingualen Trainingsdaten (*Corpora*)
   - muss für jedes Sprachpaar wiederholt werden
   - sollte auch bei Wechsel der Domäne wiederholt werden (z.B. Parlamentsdebatten vs. Telefonreservierungsystem)
3. Das Modell ist nun komplett bekannt. Eingabesätze können maschinell übersetzt werden. Dazu wird idealerweise der Ausgabesatz mit den geringsten Kosten ermittelt.

Inhalt der Vorlesung

- Word Alignment
- Sprachmodellierung (Language Modelling)
- Phrasenbasierte Ansätze
- Evaluierung von Übersetzungsprogrammen
- Tuning (Anpassung von Gewichtungsparametern)
- Baum- und Syntaxbasierte Ansätze

Literatur

Die Vorlesung basiert auf dem Buch:

Philipp Koehn, *Statistical Machine Translation*, Cambridge University Press, 2010.

Evtl. auch hilfreich:

- Christopher Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press 1999.
- Online-Tutorial von Kevin Knight: `http://www.isi.edu/natural-language/mt/wkbk.rtf`
- Carstensen et al. (Hrsg) : Computerlinguistik und Sprachtechnologie
- Verwandte Vorlesungen:
  - Einführung in die Computerlinguistik
  - Mathematische Grundlagen der Computerlinguistik

Literatur : Mathematischer Hintergrund

Für Interessierte: Mathematischer Hintergrund für Optimierung mit Nebenbedingungen:

Dimitri Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, 1999

Fallbeispiele (1)

Aus dem Europarl Corpus (Protokolle des Europäischen Parlaments, manuell übersetzt)

- `Frau Präsidentin , zur Geschäftsordnung .`

  ↔ `Madam President , on a point of order .`

gleiche Wortordnung, 1-many Entsprechungen

- `Ich bitte Sie , sich zu einer Schweigeminute zu erheben .`

  ↔ `Please rise , then , for a minute's silence .`

freie Übersetzung, reflexiv vs. irreflexiv, 'Ich', 'Sie' und 'then' ohne Entsprechung.

- `Ich werde dem Vorschlag von Herrn Evans folgen .`

  ↔ `I shall do as Mr Evans has suggested .`

Mehrwortausdruck mit Lücke: "dem Vorschlag folgen" ↔ "do as suggested".

Ergänzung vs. aktiver Ausdruck: "von Herrn Evans" ↔ "Mr. Evans has suggested ".

Keine klaren Segmente: "suggested" kommt in zwei Ausdrücken vor.

Fallbeispiele (2)

- `Das Parlament wird sich am Donnerstag mit dem Bericht [..] befassen .`

  ↔ `The report [..] comes before Parliament on Thursday .`

  Vertauschte Wortordnungen, Artikel für Parlament nur im Deutschen,
  reflexiv vs. irreflexiv, Präsens vs. Futur, Objekt wird Subjekt (≈ Passiv vs. Aktiv)

- `All dies entspricht den Grundsätzen die wir stets verteidigt haben .`

  ↔ `This is all in accordance with the principles that we`
  `have always upheld .`

  Dativausdruck vs. präpositionaler Ausdruck, unterschiedliche Wortstellung im Nebensatz,
  "uphold" ist keine Übersetzung von "verteidigen" (laut `dict.leo.org`).

Wichtig:

- Wir möchten aus diesen Daten Übersetzungsmodelle lernen.
- Daten dienen als Referenz für Evaluierung von Übersetzungsprogrammen.

Fallbeispiele für Satzfragmente

- Mismatch bei Geschlechtern:
  - Franz. `la lune` (fem.) - Deutsch `der Mond` (mask.)
- Mismatch bei Singular/Plural:
  - Deutsch `Es` **`gibt`** `10 Regeln` - Englisch `There` **`are`** `10 rules`
- Mismatch bei Präpositionen:
  - Bei Verben: Deutsch `hindern` **`an`** - Englisch `prevent` **`from`**
  - Bei Nomen: Dt. **`auf`** `der Straße` - Eng. **`in`** `the street`
- Mismatch bei Verben:
  - Dt. `eine Entscheidung` **`treffen`** - Eng. `to` **`take`** `a decision`

⇒ **Wörter sollten in ihrem *Kontext* betrachtet werden.**

Sätze vs. Texte

- Hauptgegenstand der Forschung: Übersetzung einzelner Sätze.
- Grund: Texte sind zu komplex.
- Aber: es geht teilweise wichtiger Kontext verloren.
  - "dangling pronouns": `Jim sah Lukas in den Raum gehen. Dann ging er ...`
  - Thema des Textes kann disambiguieren:
    Engl. `bat` - Dt. `Baseballschläger, Fledermaus`
  - Der Zeitpunkt kann wichtig sein: `2011` vs. `letztes Jahr`
  - Manche Sätze sind für sich genommen doppeldeutig:
    Engl. `he is trying .`
    Dt. `er versucht es .` | `er bemüht sich .` |
    `er ist ein anstrengender Mensch .`

    Partizip Präsens Aktiv vs. Adjektiv

Vorverarbeitung / Tokenization

Typische Eingabe (Engl.):

`My son's friend, however, plays a high-risk game.`

Intern repräsentiert als (Satzzeichen und Wörter sind gleichwertig.):

`my son 's friend , however , plays a high @-@ risk game .`

- Abtrennung von Satzzeichen von "echten" Wörtern (*Tokenization*) (Vorsicht: `500.000` $\rightarrow$?)
- Trennung bei "-","/", evtl. von Verbundwörtern
- alles klein geschrieben
  - weniger Datenknappheit
  - leichter Informationsverlust (mit maschinellem Lernen recht gut korrigierbar)
- pro Sprache nur ein Zeichensatz (z.B. UTF-8, Unicode)
- Evtl. Erkennung von Zahlen (`zwei | 2 | 2.0 | 10,0 | 2.`), Datumsangaben und Eigennamen

Fortgeschrittene Vorverarbeitung

- morphologische Analysen,
  z.B. `macht` → `machen | 3. Sg. Präsens, aktiv, Ind.`
- Part-of-Speech Analyse (*POS tagging*): Annotation von Nomen, Verben, Pronomen etc.
- Syntaktische/Semantische Analyse (Parsing)

Diese Schritte werden weniger universell eingesetzt.

Faustregel: was sich gut monolingual verarbeiten lässt sollte auch dort gemacht werden.
Aber: kein genereller Konsens. Nicht-triviales Problem, Methode kann Auswirkungen auf das ganze System haben.

Hauptgegenstand der Vorlesung: Kernübersetzung (nach Tokenization, z.B. lowercase → lowercase)

Übersetzung als Noisy Channel Model

Warren Weaver, 1947: *When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'*

Das Problem, einen gegebenen Satz in eine andere Sprache zu übersetzen, wird daher oft als **Decoding** bezeichnet (→ Kryptographie).

Oft auch: *noisy channel model*. Jemand schickt einen Satz in Englisch über einen Kanal (z.B. ein Glasfaserkabel), aber durch starkes Rauschen kommt er in einer anderen Sprache beim Empfänger an (→ Signalverarbeitung, Fehlerkorrektur).

Notation

Fettgedruckte Symbole/Variablen sind Vektoren, gelegentlich auch Mengen: $\mathbf{f}, \mathbf{e}, \mathbf{a}$

normalgedruckte Symbole sind Skalare: f, e, a.
Nicht-fette Großbuchstaben sind konstante Skalare: $I, J, K, \ldots$

Häufig für ein Vektor der Länge J: $\mathbf{f} = f_1^J = (f_j)_{j=1,\ldots,J}$

Kronecker-δ:

$$\delta(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{else} \end{cases}$$

$\log(\cdot)$ bezeichnet den Logarithmus *naturalis*, also zur Basis $e = 2.78\ldots$.

Häufig werden die Klammern beim Logarithmus weggelassen: $\log x$ anstatt $\log(x)$.

Notation (2)

- Das Zeichen $\sum$ bezeichnet eine Summe, z.B.

$$\sum_{k=1}^{K} = x_1 + x_2 + \ldots + x_K$$

  leere Summen sind 0, z.B. $\sum_{k=1}^{0} = 0$

  Manchmal auch Summen über Mengen: $\sum_{k \in \{1,3,7\}} r_k = r_1 + r_3 + r_7$

- Das Zeichen $\prod$ steht für ein Produkt, z.B.

$$\prod_{k=1}^{K} = x_1 \cdot x_2 \cdot \ldots \cdot x_K$$

  leere Produkte sind 1, z.B. $\prod_{k=1}^{0} = 1$

  Manchmal auch Produkte über Mengen: $\prod_{k \in \{1,3,7\}} r_k = r_1 \cdot r_3 \cdot r_7$

Notation (3)

- Das Zeichen "$\cdot$", z.B. in $(\cdot, i')$ steht für "don't care", also für etwas beliebiges, das im Folgenden nicht verwendet wird.
- Die Notation $\{p_k\}$ steht für eine Menge von Variablen, die sich jeweils aus dem Kontext ergibt. Wenn zum Beispiel ein K bekannt ist, steht sie für $\{p_1, \ldots, p_K\}$.
- Wenn von **dem** maximierenden Argument gesprochen wird, so ist dies generell als **ein** maximierendes Argument zu verstehen (es kann mehrere geben). Analog für Minimierung.

Notation (4)

- e_1^I steht für eine Sequenz von I Wörtern/Tokens
- $e_{k_1}^{k_2}$ steht für eine Untersequenz einer Sequenz e_1^I $(k_1, k_2 \leq I)$, die sich aus dem Kontext ergibt. Ist $k_2 < k_1$, so ist die Untersequenz leer. Ansonsten besteht sie aus den $k_2 - k_1 + 1$ Wörtern $e_{k_1}, \ldots, e_{k_2}$.
- Beispiel:

  $e_1^8 =$ `er macht die große rote tür auf .`

  $e_2^5 =$ `macht die große rote`

Grundlegende Begriffe

- Unter einem **Satz** verstehen wir eine Folge von Tokens, die nicht notwendig syntaktisch oder semantisch wohlgeformt ist.

  Somit ist die Folge `the the the the` für uns ein Satz.

  Bedenken Sie: auch in wohlgeformten Sätzen kommen häufig Wörter doppelt vor. Dies sind oft Artikel oder Pronomen.

- **Basiswahrscheinlichkeiten** bezeichnen Wahrscheinlichkeiten, die nicht weiter zerlegt werden, sondern direkt eingegeben oder aus Daten geschätzt werden.

  Beispiele: $p(\texttt{house}|\texttt{rot}), p(1|3)$

Wahrscheinlichkeiten: Grundlagen

Bedingte Wahrscheinlichkeiten:

$$p(A|B) = \frac{p(A,B)}{p(B)}$$

Gilt auch für mehrere Ereignisse:

$$p(A|B,C) = \frac{p(A,B|C)}{p(B|C)}$$

Nützliche Umformungen:

$$p(A,B) = p(A|B) \cdot p(B) \quad , \quad p(B) = \frac{p(A,B)}{p(A|B)}$$

Kettenregel für Wahrscheinlichkeiten:

$$p(x_1, \ldots x_n) = \prod_i p(x_i|x_{i-1}, \ldots x_1)$$

Wahrscheinlichkeiten für Wörter

Es gibt eine kleine Anzahl von Wörtern, die recht häufig sind, z.B.
– Pronomen
– Artikel
– Hilfsverben
– sehr gängige Nomen, Adjektive, Verben

Andererseits gibt es sehr viele Wörter, die nur sehr selten auftreten.

Man sagt, dass die Verteilung der Wörter *schief* oder *unbalanciert* (engl. *skewed*) ist.
Anmerkung: dies gilt auch für die Verteilung der Buchstaben.

Zipfs Gesetz: die Wk. eines Wortes in Zeile r in der Liste der häufigsten Wörter ist proportional zu $1/r$ (stimmt nur approximativ).

Wahrscheinlichkeiten für Sätze

- Notation: $e_1^I = (e_i)_{i=1,\ldots I}$: Satz der Länge I.
- Grundlegend für die statistische Übersetzung: Wahrscheinlichkeit eines Satzes $p(e_1^I)$.
- Es gibt immens viele Sätze. $\Rightarrow$ Reduktion auf kleinere Einheiten. (Kettenregel, n-gram Zerlegung, Unabhängigkeitsannahme)

$$\text{mit Bigram:} \qquad p(e_1^I) = \prod_i p(e_i|e_{i-1})$$

$$\text{mit Trigram:} \qquad p(e_1^I) = \prod_i p(e_i|e_{i-1}, e_{i-2})$$

$$\text{mit } n\text{-gram:} \qquad p(e_1^I) = \prod_i p(e_i|e_{i-1}, \ldots, e_{i-(n-1)})$$

Wahrscheinlichkeiten: Bayes' Theorem

Es gilt:

$$p(A|B) = \frac{p(A,B)}{p(B)} = \frac{p(B|A)\,p(A)}{p(B)}$$

In unserem Kontext:

– B ist schon bekannt, wir suchen nach einem A

$\Rightarrow$ der Nenner kann weggelassen werden

Probabilistische Übersetzung

Historisch bedingt (Brown et al. '93):
f_1^J = Quellsatz (für Französisch, Eselsbrücke: f="from").
e_1^I = Zielsatz (für Englisch).

Probabilistisches Übersetzungsproblem für f_1^J:

$$\begin{aligned} \text{finde} \quad & \arg\max_{I, e_1^I} \; p(e_1^I, I | f_1^J) \\ = \quad & \arg\max_{I, e_1^I} \; p(f_1^J | e_1^I) \cdot p(e_1^I, I) \end{aligned}$$

- $p(f_1^J|e_1^I)$ generisches Symbol für das Übersetzungsmodell
  - verschiedene Modelle im Lauf der Vorlesung
  - Hauptaufgabe: Verbindung zwischen Quell- und Zielsatz herstellen
- $p(e_1^I, I)$ Sprachmodell (engl. *language model*)
  - Hauptaufgabe: Wohlgeformtheit des Zielsatzes fördern

Freie Software

- Word Alignment
  - GIZA++
  - Berkeley Aligner
  - RegAligner
  - ...
- Übersetzungsprogramme
  - MOSES
  - Joshua
  - CDec
  - Jane
  - ...

Bilinguale Daten

Frei verfügbar:

- Debatten des europäischen Parlaments (`Europarl`)
  - 21 Sprachen, bis zu 2 Millionen Sätze pro Sprache
- Debatten des kanadischen Parlaments, ISI release (`Hansards`)
  - Französisch - Englisch, ca. 200000 Satzpaare (Senate Debates)
  - mehr Daten verfügbar über das LDC (s.u.)

Nur mit Bezahlung:

- Unzählige Korpora, besonders beliebt: Englisch gepaart mit Arabisch, Chinesisch, Japanisch
- Verfügbar meist über das LDC (Linguistic Data Consortium, UPenn)
- Generell: fast jede Sprache abgedeckt (soweit muttersprachliche Wissenschaftler verfügbar).

Lagrang'sche Optimierung (Grundlage für später)

Wir werden (im Training) Probleme der folgenden Art betrachten:

$$\min_{\{p(x)\,|\,x\in X\}} g(\{p(x)\})$$
$$\text{s.t.} \sum_{x\in X} p(x) = 1, \quad p(x) \geq 0 \quad \forall x \in X$$

Zunächst: Addieren von $\lambda(\sum_{x\in X} p(x) - 1)$ für ein (belieb.) $\lambda \in \mathbb{R}$

$$\min_{\{p(x)\,|\,x\in X\}} g(\{p(x)\}) + \lambda(\sum_{x\in X} p(x) - 1)$$
$$\text{s.t.} \sum_{x\in X} p(x) = 1, \quad p(x) \geq 0 \quad \forall x \in X$$

Problem äquivalent: neuer Term ist 0 für alle zulässigen $\{p(x)\}$.

Die neue Kostenfunktion heißt Lagrange-Funktion.

Lagrang'sche Optimierung (2)

Nächster Schritt: Weglassen der Bedingung $\sum_{x \in X} p(x) = 1$:

$$\min_{\{p(x) \,|\, x \in X\}} \left[g(\{p(x)\}) + \lambda \left(\sum_{x \in X} p(x) - 1 \right) \right]$$
$$\text{s.t. } p(x) \geq 0 \quad \forall x \in X$$

Jetzt:

- i.A. nicht mehr äquivalent, wir minimieren über eine größere Menge an Hypothesen.
- Wegen größerer Menge: neues Minimum $\leq$ altes.
- Minimaler Wert und minimierendes Argument sind abhängig von λ.

Konsequenz: maximiere über λ.

Lagrang'sche Optimierung (3)

Resultierendes Endproblem:

$$\max_{\lambda} \min_{\{p(x) \mid x \in X\}} \left[g(\{p(x)\}) + \lambda \left(\sum_{x \in X} p(x) - 1 \right) \right]$$

$$\text{s.t. } p(x) \geq 0 \quad \forall x \in X$$

Gute Nachricht: (insbesondere) für lineare Gleichheitsbedingungen sind das ursprüngliche und das modifizierte Problem äquivalent.
($\rightarrow$ Bertsekas, Nonlinear Programming).

Dieses Prinzip heißt (starke) *Lagrange Dualität*, λ heißt *Lagrange Multiplikator*.

Bei mehreren Gleichheitsbedingungen gibt es einen separaten Multiplikator für jede Bedingung.