

Erweiterungsmodul: Maschinelle Übersetzung

Teil 1: Statistische maschinelle Übersetzung

Helmut Schmid

22. Mai 2019

- Vorlesung: Mittwoch 14:15 – 15:45 in Raum 115
- Übungen: Dienstag 16:15 – 17:45 in Raum 123 (oder einem Rechnerpool)
- Folien und weitere Informationen finden Sie auf der Kursseite, die über meine Homepage erreichbar ist

Thema des Kurses ist die maschinelle Übersetzung:

- vorwiegend aus der Sicht der Sprachverarbeitung
 - Herausforderungen der Modellierung der maschinellen Übersetzung
 - Grundlegendes Verständnis der regelbasierten maschinellen Übersetzung
 - Vertieftes Verständnis der statistischen maschinellen Übersetzung
 - Einführung in Deep Learning und neuronaler MÜ
- aber teilweise auch aus linguistischer Sicht
 - Verständnis der linguistischen Herausforderungen der Übersetzung
 - Besondere Herausforderungen der Übersetzung bei verschiedenen Sprachpaaren

Koehn, Philipp (2009): **Statistical Machine Translation**

Was wird im Kurs von Ihnen verlangt?

- Abgabe der Übungen (gibt Bonuspunkte)
- Klausur am Semesterende

Fragen zum Organisatorischen?

Diese Vorlesung basiert auf der Vorlesung von **Alex Fraser** im SS 2017, die wiederum auf einer Vorlesung von **Chris Callison-Burch** basiert. Es werden auch Folien von **Philipp Koehn** verwendet.

Was ist maschinelle Übersetzung?

- Automatische Übersetzung von Text aus einer Sprache in eine andere
- Beispiele: Systran Babelfish, Moses, Google Übersetzer, Bing Übersetzer, DeepL etc.

Warum ist MÜ schwierig?

- Ambiguitäten bzgl. Wortart und Wortbedeutung
- Wortstellung
- Pronomen
- Zeit
- Idiome
- etc...

Unterschiedliche Wortstellungen

- Englisch: SVO
- Japanisch: SOV
- Englisch: IBM bought Lotus
- Japanisch: IBM Lotus bought
- Englisch: Reporters said IBM bought Lotus
- Japanisch: Reporters IBM Lotus bought said

Pronomen sind oft eine große Herausforderung bei der Übersetzung:

- Bei einigen Sprachen wie Spanisch oder Italienisch sind Subjektpronomen optional (**Pro-Drop**-Sprachen)
- Stattdessen zeigt die Verbflexion die Person an:
 - o ⇒ ich
 - as ⇒ du
 - a ⇒ er / sie / es
 - amos ⇒ wir
 - áis ⇒ ihr
 - an ⇒ sie
- Wann sollte **er/sie/es** verwendet werden?
- Wie sollte das engl. Wort **it** ins Deutsche übersetzt werden?

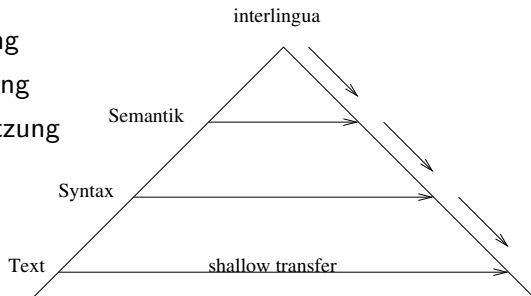
Unterschiede bei den Zeitformen

- Spanisch hat zwei Vergangenheitsformen:
 - eine für eine bestimmte Zeit in der Vergangenheit und
 - eine für eine unbestimmte Zeit
- Bei der Übersetzung vom Deutschen oder Englischen ins Spanische muss eine der beiden Formen ausgewählt werden.

- to kick the bucket bedeutet sterben
- Ein bone of contention hat nichts mit Knochen zu tun
- lame duck, tongue in cheek, to cave in
- etc...

Methoden der maschinellen Übersetzung

- Wort-für-Wort-Übersetzung
- Syntax-basierte Übersetzung
- Semantik-basierte Übersetzung
- Interlingua-Ansätze

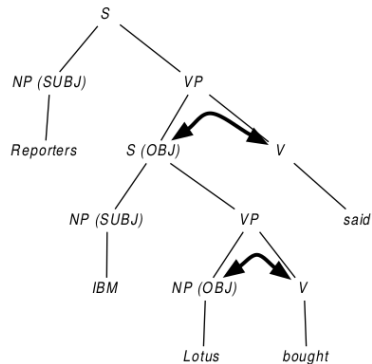
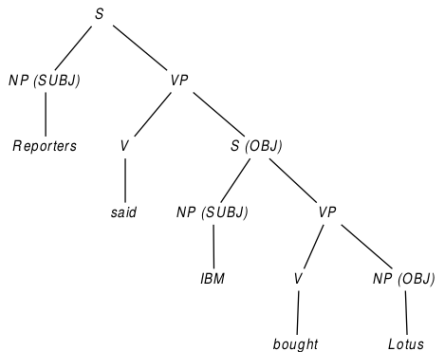


- kontrollierte Sprache

- beispielbasierte Übersetzung
- statistische MÜ
- neuronale MÜ

- Jedes Wort im Text wird mit einem **bilingualen Wörterbuch** übersetzt
- Vorteile
 - einfach zu implementieren
 - liefert eine grobe Idee vom Textinhalt
- Nachteile
 - Probleme mit Wortstellung, Wortambiguitäten, Pronomen etc.
 - schlechte Übersetzungsqualität

Syntax-basierte Übersetzung



Schritte:

- Satz parsen
- Konstituenten umordnen
- Wörter übersetzen

- Vorteile

- löst das Wortstellungsproblem
- Die Komponenten sind wiederverwendbar:

Ein englischer Parser kann in einem EN-DE und in einem EN-FR-System verwendet werden

- Nachteile

- Es muss eine Grammatik/Parser für jede Sprache entwickelt werden
- Manchmal verwenden Sprachen unterschiedliche syntaktische Kategorien

Peter likes to swim

Peter schwimmt gerne

- Der Satz wird in eine logische Formel übersetzt
John must not go \Rightarrow OBLIGATORY(NOT(GO(JOHN)))
- Aus der logischen Formel wird ein Satz der Zielsprache generiert
OBLIGATORY(NOT(GO(JOHN))) \Rightarrow John darf nicht gehen

- Vorteile
 - eine einzige sprachunabhängige Repräsentation
 - Es kann zwischen beliebigen Sprachen übersetzt werden, für die ein Parser/Generator existiert
- Nachteile
 - Eine solche sprachunabhängige Repräsentation zu definieren und zu generieren ist nur in stark eingeschränkten Anwendungsbereichen möglich.

- Der Satz wird in eine (sprachabhängige) logische Formel übersetzt
 $\text{John likes to swim} \Rightarrow \text{LIKE}(\text{SWIM}(\text{JOHN}))$
- Die englische logische Formel wird in eine deutsche übersetzt
 $\text{LIKE}(\text{SWIM}(\text{JOHN})) \Rightarrow \text{GERNE}(\text{SCHWIMMEN}(\text{JOHN}))$
- Aus der deutschen logischen Formel wird ein Satz der Zielsprache generiert
 $\text{GERNE}(\text{SCHWIMMEN}(\text{JOHN})) \Rightarrow \text{John schwimmt gerne}$

Im deutschen **VerbMobil**-Projekt wurden flache Übersetzung (bspw. für Grußformeln), Syntax-basierte Übersetzung und Semantik-basierte Übersetzung kombiniert.

- Definiere eine **Teilmenge** der Sprache, die einfach zu übersetzen ist und bspw. keine Ambiguitäten erlaubt.
- Stelle durch entsprechende **Richtlinien** sicher, dass alle zu übersetzenden Texte in dieser Teilsprache formuliert werden.
- **Übersetze** auf Basis von Syntax/Semantik/Interlingua
- Beispiele: **Wetterberichte, Werkstatt-Handbücher**

- **Vorteil:** Die Übersetzungen in dem eingeschränkten Sprachbereich sind recht zuverlässig und hochwertig
- **Nachteil:** nicht auf beliebige Text anwendbar, nur auf Texte, die den Richtlinien folgen

- **Ziel:** Übersetzer unterstützen (statt ihn zu ersetzen)
- erfordert ein **Parallelkorpus** oder einen Translation Memory
- Wenn für den Satz(teil), der gerade übersetzt wird, im Speicher bereits eine Übersetzung vorliegt, wird diese dem Übersetzer vorgeschlagen.
- Mit geeigneten Regeln und Heuristiken können auch Sätze übersetzt werden, für die nur ein ähnlicher Satz im Speicher gefunden wurde.

Parallelkorpus

Source	Translation
A-t-on acheté les actions ou les biens des entreprises nationalisées?	Have the shares or properties of nationalized companies been purchased?
Quel était le genre de travaux exécutés aux termes de ces contrats?	What was the nature of the work performed under these contracts?
Le recours est rejeté comme manifestement irrecevable	The action is dismissed as manifestly inadmissible
Les propositions ne seront pas mises en application maintenant.	The proposal will not now be implemented.
La République française supportera ses propres dépens	France was ordered to bear its own costs
Production domestique exprimée en pourcentage de l'utilisation domestique	Domestic output as a % of domestic use
La séance est ouverte à 2 heures.	The House met at 2 p.m.
...	...

- Vergleich mit **menschlischer Übersetzung** ohne CAT
 - + schneller und dadurch geringere Kosten
 - + unterstützt die einheitliche Übersetzung bspw. von Fachausdrücken
- Vergleich mit **maschineller Übersetzung**
 - + höhere Qualität
 - höhere Kosten

- berechnet die **wahrscheinlichste Übersetzung** eines Satzes
- verwendet ein **statistisches Modell** der Übersetzung
- Das Modell wird auf einem Parallelkorpus **trainiert**.

- Vorteile:
 - funktioniert für alle Sprachpaare
 - kann mit lexikalischen **Ambiguitäten** und **Idiomen** umgehen
 - geringer Aufwand für die Anpassung an neue Sprachpaare
- Nachteile:
 - erfordert ein **großes Parallelkorpus**
 - erzeugt manchmal **ungrammatische** Sätze
 - Die Übersetzungssysteme sind recht **komplex**
 - schwer zu analysieren, wie eine Übersetzung zustande gekommen ist

- ähnlich der statistischen maschinellen Übersetzung
- verwendet ein neuronales Netzwerk als statistisches Modell
- berechnet ebenfalls die wahrscheinlichste Übersetzung

Vorteile gegenüber SMT

- einfachere Implementierung
- bessere Übersetzungen
- aktuell der Stand der Technik

Nachteile gegenüber SMT

- Das Training ist aufwändig
- noch schwerer nachzuvollziehen, wie eine Übersetzung zustande kam

Wir haben betrachtet

- einige **linguistische Probleme** bei der maschinellen Übersetzung
- verschiedene Ansätze zur **maschinellen Übersetzung** (im Überblick)

In den weiteren Vorlesungen werden wir

- einige **linguistische Probleme** ausführlicher untersuchen
- die Methoden der **maschinellen Übersetzung** genauer betrachten
- **statistische** und **neuronale Übersetzung** detailliert behandeln

- wurde früh als mögliche Computeranwendung erkannt
- **Warren Weaver (1949):** *I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.*
- IBM hat 1954 ein einfaches Wort-für-Wort-Übersetzungssystem entwickelt.

Warum ist MÜ relevant?

Kommerzielles Interesse

- Automatische und computerunterstützte Übersetzung kann **Kosten reduzieren**
- Texte, deren manuelle Übersetzung zu teuer wäre, können damit übersetzt werden.
- Eine automatische Übersetzung geht **schneller** als eine manuelle.
- Die **EU** gibt pro Jahr fast eine halbe Milliarde Euro für Übersetzungen und Dolmetscherdienste aus
- Die **US-Geheimdienste** sind sehr an MÜ interessiert und haben viel Geld in Forschung investiert.
- Die maschinelle Übersetzung hat in der vergangenen 25 Jahren große Fortschritte gemacht.

Warum ist MÜ relevant?

Akademisches Interesse

- eine der größten Herausforderungen in der maschinelle Sprachverarbeitung
- erfordert umfassendes sprachliches Wissen (Morphologie, Syntax, Semantik, Pragmatik, Weltwissen)
- Linguistische **Annotationen** (Wortart-Tags, Parsebäume etc.) können mit MÜ-Hilfe in andere Sprachen **transferiert** werden

Welche Übersetzungseinheiten?

- eigentliches Ziel: hochwertige Übersetzung ganzer **Dokumente**
- Fast alle Systeme arbeiten derzeit auf **Satzebene**.
- Die Übersetzung einzelner Sätze ist ein wichtiges **Teilproblem**.
- Aber manchmal wird satzübergreifender **Kontext** benötigt:
Look at this cat/dog! Isn't it cute? er oder sie
Did you see this car? It is driving too fast! es (Auto) oder er (Wagen)

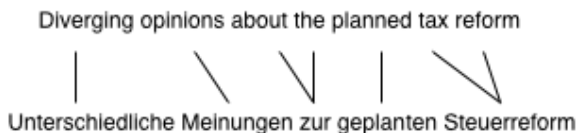
Erstellung eines SMÜ-Systemes

- Ein Korpus von Übersetzungen wird zusammengestellt
⇒ Parallelkorpus
- Satzalignierung: In jedem Dokumentpaar werden diejenigen Sätze bestimmt, die Übersetzungen voneinander sind
⇒ Liste von Satzpaaren (je ein Satz und seine Übersetzung)
- Wortalignierung: In jedem Satzpaar werden die Wörter des einen Satzes mit ihren Entsprechungen in der Übersetzung verbunden.
⇒ Liste von wortalignierten Satzpaaren
- Training des Übersetzungsmodelles auf den wortalignierten Satzpaaren
⇒ Modellparameter
- Anwendung des Modelles auf neuen Text
⇒ Übersetzung

- gegeben: Ein **Quelldokument** und seine **Übersetzung**
- gesucht: die Übersetzung jedes **Satzes** des Quelldokumentes
- Der n-te Satz der Übersetzung ist nicht unbedingt die Übersetzung des n-ten Quellsatzes
- Außer 1:1-Entsprechungen gibt es auch die Fälle 1:0 (Löschung), 0:1 (Einfügung) und n:m ($n, m \geq 1$)
- In den europäischen Parlamentsdebatten sind etwa 90% der Satzentsprechungen 1:1

- **Align** (Gale & Church)
 - aligniert Sätze auf Basis ihrer Länge in Buchstaben
 - Bei kurzen Sätzen ist eine kurze Übersetzung wahrscheinlich
 - Bei langen Sätzen ist eine lange Übersetzung wahrscheinlich
 - 1:1 Übersetzungen sind wahrscheinlicher als 1:0, 0:1, 1:2, 2:1 etc.
 - funktioniert recht gut
 - Probleme bei längeren Einschüben in einem der Dokumente
- **Char-Align** (Church)
 - aligniert anhand von identischen Buchstabenfolgen
 - funktioniert gut bei ähnlichen Sprachen und technischen Texten
- **Cognates** (Melamed)
 - benutzt Cognates (einschließlich Sonderzeichen) zur Alignierung
- **Length & Lexicon** (Moore; Braune & Fraser)
 - Alignierung auf Basis von Buchstabenlängen
 - Extraktion eines bilingualen Lexikons
 - verfeinerte Alignierung mit Hilfe des Lexikons

In jedem Satzpaar alignieren wir die Wörter, die Übersetzungen voneinander sind:



Teilprobleme bei der Erstellung eines SMÜ-Systems

- Definition eines statistischen **Modelles**
- **Schätzung** der Modellparameter
- **effiziente Berechnung** der wahrscheinlichsten Übersetzung eines Satzes (Decoding)
- **Evaluierung** auf Testdaten

Mit einem SMÜ-Modell wollen wir die **wahrscheinlichste Übersetzung** $\hat{\mathbf{e}}$ eines Satzes \mathbf{f} bestimmen:

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg \max_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})}{p(\mathbf{f})} = \arg \max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e})$$

Dieses **Noisy-Channel-Modell** besteht aus zwei Komponenten:

- dem Sprachmodell $p(\mathbf{e})$
- dem Übersetzungsmodell $p(\mathbf{f}|\mathbf{e})$

- SMÜ wurde von Forschern aus dem Bereich der **Spracherkennung** entwickelt
- Dort entspricht $p(\mathbf{e})$ dem Sprachmodell und $p(\mathbf{f}|\mathbf{e})$ dem **Akustikmodell**.
- Schon Warren Weaver hat die Übersetzung als **Dekodierung** eines verrauschten Signales interpretiert.
- Das Modell für $p(f|e)$ in der SMÜ unterscheidet sich aber von dem in der Spracherkennung dadurch, dass die Wörter **umgestellt** werden können.

Folgende **Vorgehensweise** wurde in der Spracherkennung (SE) entwickelt:

- Reduziere das Evaluierungsergebnis auf eine einzige Zahl
 - In der SE wird die Ausgabe des Systems mit einem **Transkript** verglichen
 - und die **Ähnlichkeit** berechnet
 - Dann wird der Erkenner modifiziert, um die Ähnlichkeit zu erhöhen.
- **Shared Tasks**: Alle sollten dieselben Daten verwenden, damit die Ergebnisse vergleichbar sind.

Diese Vorgehensweise wurde in der Sprachverarbeitung übernommen und ist heute Standard.

- SMÜ kann auf der Ebene eines Korpus, Dokumentes oder Satzes evaluiert werden.
- Eine Evaluierung sollte zwei **Aspekte** der Übersetzungsqualität messen:
 - **Adäquatheit**: Wird die Satzbedeutung korrekt übermittelt?
 - **Flüssigkeit**: Ist die generierte Übersetzung grammatikalisch korrekt?

Eingabe: Ich bin müde.

	Adäquatheit	Flüssigkeit
Tired is I.	5	2
Cookies taste good!	1	5
I am tired.	5	5

Grundidee:

- Vergleich der automatischen Übersetzung mit einer manuell erstellten Übersetzung
- Berechnung eines Evaluierungsmaßes

- **Editierabstand** (Levenshtein-Abstand) zur Referenzübersetzung = minimale Zahl der Wortersetzungen, -löschungen, und -einfügungen, um die Ausgabe in die Referenzübersetzung umzuwandeln
- Der Editierabstand wird dann noch durch die Länge der Referenzübersetzung geteilt.
- Die „Flüssigkeit“ wird gut erfasst.
- Die Adäquatheit wird weniger gut erfasst.
- Der Vergleich ist zu streng:
Ausgabe 1: He saw a man and a woman
Ausgabe 2: He saw a cat and a dog
Referenz: He saw a woman and a man

⇒ Beide Ausgabe erhalten dieselbe Bewertung.

Positionsunabhängige Wortfehlerrate (PER)

- Hier wird die **Überlappung** der Wortmengen der beiden Sätze gemessen.
- Dazu wird die WER berechnet, nachdem beide Wortlisten sortiert wurden.
- Die **Adäquatheit** wird auf Wortebene gut gemessen.
- Die **Flüssigkeit** wird überhaupt nicht erfasst.
- Der Vergleich ist nicht streng genug:
Ausgabe 1: he saw a man
Ausgabe 2: saw man a he
Referenz: he saw a man

⇒ Beide Ausgaben erhalten dieselbe Bewertung.

- Geometrisches Mittel der **Precision** der Mengen von 1-, 2-, 3- und 4-Grammen
- zusätzlicher **Brevity Penalty**
- Bei der Berechnung der Precision wird **Clipping** angewendet:
Ausgabe : the the the the the
Referenz: the man ate the cake

⇒ Unigramm-Precision: 2/5 und nicht 5/5
- BLEU **korreliert** auf Korpusebene gut mit menschlichen Bewertungen, nicht aber auf Satzebene

- BLEU ist gut geeignet für den Vergleich von **SMÜ-Systemen** auf denselben Daten
- aber weniger geeignet, um bspw. SMÜ-Systeme mit **regelbasierten Systemen** zu vergleichen.
- **METEOR** ist eine Erweiterung von BLEU, die auch positiv berücksichtigt, wenn zwar das **Lemma** nicht aber die Flexionsform korrekt ist.
- Für die Bewertung **einzelner Sätze** gibt es kein gutes automatisches Maß.
- BLEU ist kein **absolutes** Qualitätsmaß.
Ein System mit BLEU-Score 25 auf Korpus 1 kann besser sein als ein System mit BLEU-Score 30 auf einem Korpus 2.

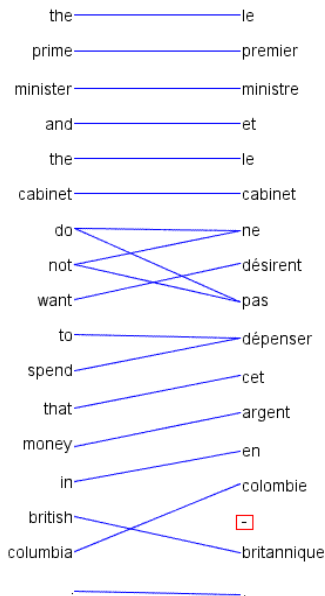
Zuletzt behandelt:

- parallele Korpora
- Satzalignierung
- Prinzip der maschinellen Übersetzung
- Evaluierung und BLEU

Als Nächstes:

- Wortalignierung
- IBM-Modelle
- Phrasen-basierte SMÜ
 - Training
 - Decoding

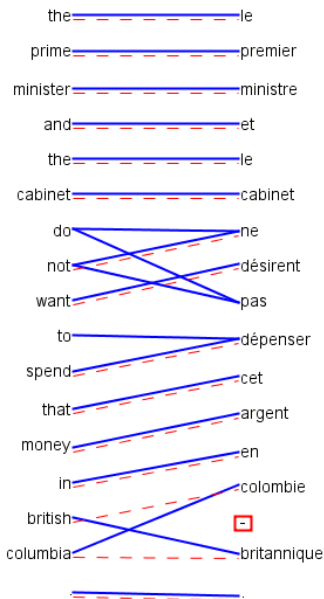
Wortalignierung



- Annotation **minimaler** Übersetzungsentsprechungen
- im konkreten Kontext

Das Bild zeigt manuell erstellte Alignierungen.

Wortalignierung



- Automatische Alignierungen werden oft mit **IBM Modell 4** erzeugt.
- kein **linguistisches** Wissen
- kein Training auf manuell **annotierten** Texten
- **unüberwachtes** Lernen der Wortalignierung

Die roten gestrichelten Linien im Bild zeigen automatische Alignierungen.

- multilingual
 - statistische maschinelle Übersetzung
 - Extraktion von bilingualen Wörterbüchern
 - Cross-Lingual Information Retrieval
 - Projektion linguistischer Annotationen
 - Verbesserung der Satzalignierung
 - Extraktion paralleler Sätze aus ähnlichen Korpora
- monolingual
 - Paraphrasierung
 - automatische Zusammenfassung

Idee 1:

- SMÜ-System mit den alignierten Daten trainieren
- auf Testdaten mit BLEU evaluieren

Vor- und Nachteile:

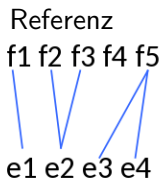
- + Die Evaluierung misst, worauf es wirklich ankommt.
- Training und Evaluierung eines SMÜ-Systems sind aufwändig.
- Das Ergebnis hängt von dem verwendeten SMÜ-System ab.

Idee 2:

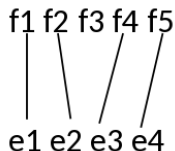
- Berechnung von **Precision** und **Recall** bzgl. manuell annotierter Referenzdaten

Vor- und Nachteile:

- + einfach und schnell
- benötigt manuell annotierte Daten
- Die Evaluierung misst nicht direkt, worauf es wirklich ankommt.



generiert



Precision = 3/4 da (e3,f4) falsch ist

Recall = 3/5 da (e2,f3) und (e3,f5) fehlen

F_α -Score:

$$F_\alpha = \frac{1}{\frac{\alpha}{precision} + \frac{1-\alpha}{recall}}$$

- α erlaubt es, Precision oder Recall höher zu gewichten
- $0.1 < \alpha < 0.4$ sinnvoll für SMÜ
⇒ Recall ist wichtiger

- Übersetzung eines Wortes durch Nachschlagen im **Lexikon**
Haus → house, building, home, household, shell
- mehrere mögliche Übersetzungen
 - einige häufiger als andere bspw. house und building
 - Spezialfälle: shell ist das Haus einer Schnecke
- Anm.: Im Folgenden wird immer ins Englische übersetzt

Zählen, wie oft **Haus** mit welchem Wort übersetzt wurde

Übersetzung	Häufigkeit
house	8000
building	1600
home	200
household	150
shell	50

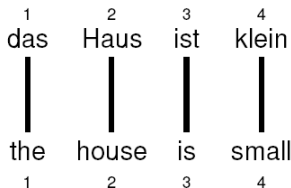
Schätzung der Übersetzungswahrscheinlichkeiten

Maximum-Likelihood-Schätzung

Übersetzung von Haus	Häufigkeit	Wahrscheinlichkeit
house	8000	0.8
building	1600	0.16
home	200	0.02
household	150	0.015
shell	50	0.005

Wort-Alignierung

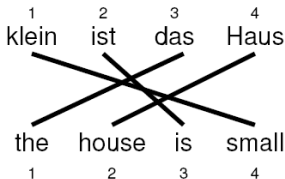
In einem parallelen Text **alignieren** wir Wörter der einen Sprache mit Wörtern der anderen Sprache



Die **Wortpositionen** werden mit 1–4 durchnummeriert.

- Wir formalisieren die Alignierung durch eine **Alignierungsfunktion** **a**
- Diese bildet ein englisches Zielwort an Position i auf ein deutsches Quellwort an Position j ab
- Beispiel: **a**: $\{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$

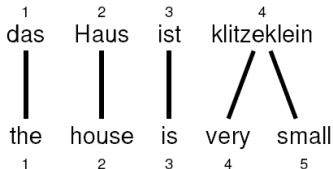
Wörter können bei der Übersetzung **umgeordnet** werden:



a: {1 → 3, 2 → 4, 3 → 2, 4 → 1}

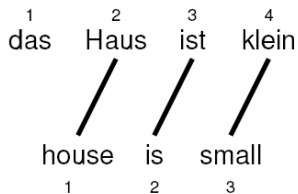
1:n-Übersetzungen

Ein Quellwort kann mit mehreren Zielwörtern übersetzt werden:



$a: \{1 \rightarrow 1, 2 \rightarrow 1, 3 \rightarrow 1, 4 \rightarrow 4, 5 \rightarrow 4\}$

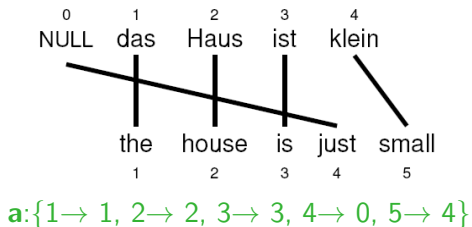
Wörter können bei der Übersetzung **weggelassen** werden:



a: {1 → 2, 2 → 3, 3 → 4}

Worteinfügungen

Wörter können bei der Übersetzung **hinzugefügt** werden:



- Alignmentfunktionen liefern eine **einfache Repräsentation** des Alignmentgraphen
- Aber sie sind **asymmetrisch**
 - Ein Nullsymbol gibt es nur auf der deutschen Seite
 - Deutsche Wörter können mit mehreren englischen Wörtern aligniert sein
 - aber nicht umgekehrt!

Wir werden nun die IBM-Modelle betrachten, die 1993 von Brown et al. bei IBM als **statistische Übersetzungsmethode** entwickelt wurden.

Heute werden diese Modelle nur noch für die **Wortalignierung** eingesetzt.

Es handelt sich um **generative Modelle**. Sie zerlegen den Übersetzungsprozess in viele kleine Schritte.

Wir suchen die wahrscheinlichste Übersetzung $\hat{\mathbf{e}}$ eines gegebenen Satzes \mathbf{f}

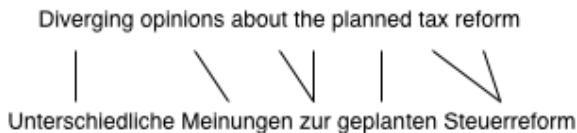
$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg \max_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})}{p(\mathbf{f})} = \arg \max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e})$$

Für $p(\mathbf{e})$ nehmen wir ein NGramm-Sprachmodell

$$p(\mathbf{e}) = p(e_1, \dots, e_n) = \prod_{i=1}^{n+1} p(e_i | e_{i-k}, \dots, e_{i-1})$$

Wie können wir $p(\mathbf{f}|\mathbf{e})$ definieren?

Wir nehmen an, dass jedes Wort in **f** (oberer Satz) die Übersetzung eines bestimmten Wortes in **e** (unterer Satz) ist, mit dem es aligniert ist.



Da es mehrere Alignierungen von **e** und **f** geben kann, definieren wir $p(\mathbf{f}|\mathbf{e})$ als Summe über alle diese Alignierungen **a**:

$$p(\mathbf{f}|\mathbf{e}) = \sum_a p(\mathbf{a}, \mathbf{f}|\mathbf{e})$$

a ist hier eine **versteckte Variable**, weil es nicht bekannt ist.

$p(\mathbf{a}, \mathbf{f} | \mathbf{e})$ können wir allgemein weiter zerlegen in:

$$p(\mathbf{a}, \mathbf{f} | \mathbf{e}) = p(J | \mathbf{e}) \prod_{j=1}^J p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e})$$

Damit modellieren wir einen statistischen Prozess, welcher

- die **Länge** J des Satzes \mathbf{f} mit der Wahrscheinlichkeit $p(J | \mathbf{e})$ wählt
- und dann für jede Position in \mathbf{f} von 1 bis m
 - eine **e-Position** a_j mit Wahrscheinlichkeit $p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e})$ wählt
 - und ein **Wort** f_j mit Wahrscheinlichkeit $p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e})$ wählt

→ Beispiel durchspielen

IBM Modell 1

$p(\mathbf{a}, \mathbf{f} | \mathbf{e})$ können wir allgemein weiter zerlegen in:

$$p(\mathbf{a}, \mathbf{f} | \mathbf{e}) = p(J | \mathbf{e}) \prod_{j=1}^J p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e})$$

IBM Modell 1 macht nun die folgenden **vereinfachenden Annahmen**

- Die Wahrscheinlichkeit der Länge des Zielsatzes J ist uniform verteilt

$$p(J | \mathbf{e}) = \varepsilon$$

- Auch die möglichen Alignierungen a_j sind alle gleich wahrscheinlich

$$p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) = \frac{1}{I+1}$$

- Das Wort f_j hängt nur von dem damit alignierten Wort e_{a_j} ab:

$$p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e}) = p(f_j | e_{a_j})$$

Ergebnis: (I =Länge von \mathbf{e})

$$p(\mathbf{a}, \mathbf{f} | \mathbf{e}) = \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^J p(f_j | e_{a_j})$$

Die einzigen trainierbaren Parameter sind hier die Übersetzungswahrscheinlichkeiten $p(f|e)$

$$p(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^J p(f_j|e_{a_j})$$

Da die einzelnen f_j hier völlig unabhängig voneinander generiert werden, kann die Aposteriori-Wahrscheinlichkeit der Alignierung a_j sehr einfach berechnet werden:

$$p(a_j = i|\mathbf{f}, \mathbf{e}) = \frac{p(f_j|e_i)}{\sum_{i'=0}^n p(f_j|e_{i'})}$$

Das Training von IBM Modell 1 erfolgt iterativ mit dem **EM-Algorithmus**:

- **E-Schritt:** Berechnung der erwarteten Häufigkeit $c(e,f)$ für alle Paare (e,f)
- **M-Schritt:** Neuschätzung der Übersetzungswahrscheinlichkeiten $p(f|e)$ aus den Häufigkeiten

EM-Pseudocode:

Uniforme Initialisierung von $p(f|e)$ mit $\frac{1}{F}$ (F = Vokabulargröße)

Für T Iterationen

// E Schritt

Häufigkeiten $c(e, f)$ mit 0 initialisieren

Für alle Satzpaare \mathbf{f}, \mathbf{e}

 Für alle Positionen j in Satz \mathbf{f}

 Für alle Positionen i in Satz \mathbf{e}

$p(a_j = i | \mathbf{f}, \mathbf{e})$ berechnen (Formel auf vorheriger Folie)

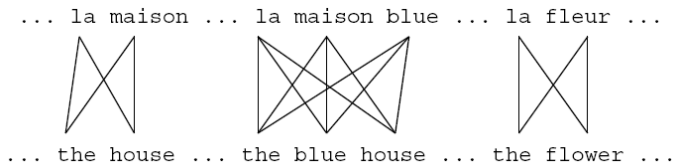
 Häufigkeit $c(e_i, f_j)$ um $p(a_j = i | \mathbf{f}, \mathbf{e})$ erhöhen

// M Schritt

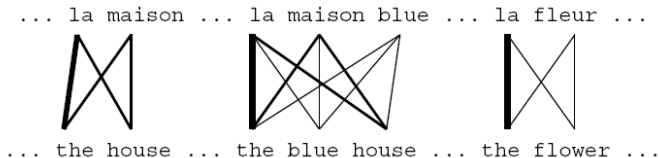
Für alle Wortpaare e, f

 Wahrscheinlichkeit neu schätzen $p(f|e) = \frac{c(e, f)}{\sum_{f'} c(e, f')}$

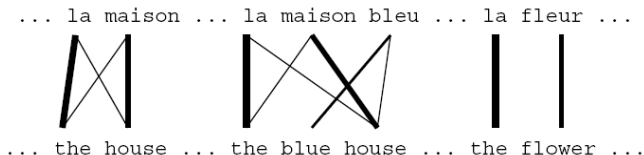
Anm. Der Code kann noch effizienter implementiert werden.



- Am Anfang sind alle Alignierungen **gleich wahrscheinlich**
- Das Modell lernt dann bspw., dass **la** oft mit **the** aligniert ist.



- Nach einer Iteration ist die Alignierung von **la** und **the** wahrscheinlicher geworden.



- Nach einer weiteren Iteration ist die Alignierung von **fleur** und **flower** wahrscheinlich geworden. (Taubenschlagprinzip)

... la maison ... la maison bleu ... la fleur ...
/ | | X | |
... the house ... the blue house ... the flower ...

- Konvergenz
- EM hat die inhärente Struktur entdeckt.

Parameterextraktion aus den alignierten Daten

... la maison ... la maison bleu ... la fleur ...
/ | | X | |
... the house ... the blue house ... the flower ...

$$p(la|the) = 0.453$$

$$p(le|the) = 0.334$$

$$p(maison|house) = 0.876$$

$$p(bleu|blue) = 0.563$$

...

$$p(\mathbf{a}, \mathbf{f} | \mathbf{e}) = p(J | \mathbf{e}) \prod_{j=1}^J p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e})$$

- In Modell 2 sind die Alignierungswahrscheinlichkeiten **nicht mehr uniform**:

$$p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) = p(a_j | j, J, l)$$

- Das System kann daher lernen, welche Quell- und Zielpositionen häufig aligniert sind.
- $p(a_j | j, J, l)$ kann zu $p(a_j | j, l)$ vereinfacht werden, um die Parameterzahl zu reduzieren.
- Modell 2 kann ähnlich einfach wie Modell 1 trainiert werden.

$$p(\mathbf{a}, \mathbf{f} | \mathbf{e}) = p(J | \mathbf{e}) \prod_{j=1}^J p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) p(f_j | a_1^j, f_1^{j-1}, J, \mathbf{e})$$

- Hier hängt die Alignierung von der vorherigen Alignierung ab:

$$p(a_j | a_1^{j-1}, f_1^{j-1}, J, \mathbf{e}) = p(a_j | a_{j-1}, l) \quad \sim a_j - a_{j-1}$$

- **Intuition:** Wenn e_i mit f_j übersetzt wurde, wird e_{i+1} oft mit f_{j+1} übersetzt. Das Umordnen von ganzen Phrasen wird weniger bestraft.
- Die erwarteten Häufigkeiten für den E-Schritt werden hier mit dem Forward-Backward-Algorithmus berechnet.

IBM-Modell 3

verwendet die **Rückwärtsalignierung** $b_i = \{j | a_j = i\}$ und einen anderen Typ von Übersetzungsmodell:

$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = p(\mathbf{f}, \mathbf{b} | \mathbf{e}) = \left(\prod_{i=1}^I p(b_i | b_1^{i-1}, \mathbf{e}) \right) p(b_0 | b_1^I) p(\mathbf{f} | \mathbf{b}, \mathbf{e})$$

IBM Modell 3 vereinfacht die Formel folgendermaßen:

$$p(b_i | b_1^{i-1}, \mathbf{e}) = p(\phi_i | e_i) \phi_i! \prod_{j \in b_i} p(j | i, J) \quad \text{mit } \phi_i = |b_i|$$

- $p(\phi_i | e_i)$ ist ein **Fertility**-Modell
- $\phi_i!$ berücksichtigt, dass es viele Reihenfolgen gibt, in denen dieselbe Menge von Positionen b_i gewählt werden kann.
- $p(\mathbf{f} | \mathbf{b}, \mathbf{e}) = \prod_{i=0}^I \prod_{j \in b_i} p(f_j | e_i)$
- $p(b_0 | b_1^I)$ ist normalverteilt: Für jedes alignierte Wort f_j wird mit Wahrscheinlichkeit q ein null-aligniertes Wort generiert.

$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = p(\mathbf{f}, \mathbf{b} | \mathbf{e}) = \left(\prod_{i=1}^I p(b_i | b_1^{i-1}, \mathbf{e}) \right) p(b_0 | b_1') p(\mathbf{f} | \mathbf{b}, \mathbf{e})$$

IBM Modell 4 vereinfacht die Formel zu:

$$p(b_i | b_1^{i-1}, \mathbf{e}) = p(\phi_i | e_i) p_{=1}(b_{i1} - \overline{b_{\rho(i)}} | \dots) \prod_{k=2}^{\phi_i} p_{>1}(b_{ik} - b_{i,k-1} | \dots)$$

- Die **erste** Position in b_i hängt vom Abstand zur mittleren Position $\overline{b_{\rho(i)}}$ in der letzten nicht-leeren Menge $b_{\rho(i)}$ ab.
- Die **weiteren** Positionen in b_i hängen vom Abstand zur vorhergehenden Position in b_i ab.
- Während das HMM-Modell Positionen in \mathbf{e} vergleicht, vergleicht Modell 4 Positionen in \mathbf{f} .
- Anmerkung: Bei Modell 4 hängt die Alignmentwahrscheinlichkeit auch noch von der **Klasse** (Cluster) des vorhergehenden Wortes ab.

Anmerkungen

- Die Modelle 3 und 4 sind **defizient**, da sie auch unsinnigen Alignierungen eine positive Wahrscheinlichkeit geben.
- IBM **Modell 5** ist eine **nicht defiziente** Erweiterung von Modell 4 die aber in der Praxis nicht eingesetzt wird
- Bei **Modell 1 und 2** und HMM-Modell können die erwarteten Häufigkeiten des E-Schrittes **exakt** berechnet werden.
- Bei den **Modellen 3 und 4** ist das wegen der Abhängigkeiten zwischen den Alignierungen nicht mehr möglich.

Training der IBM-Modelle 3 und 4

- Hier können die erwarteten Häufigkeiten nicht exakt berechnet werden.
- Stattdessen wird zunächst die beste Alignierung gemäß Modell 1, Modell 2 oder HMM-Modell als **Startalignierung** berechnet.
- Dann wird versucht, eine bessere Alignierung gemäß Modell 3/4 zu ermitteln, indem
 - verschiedene kleine **Modifikationen** angewendet werden
Alignierungen löschen, hinzufügen, vertauschen
 - Die erhaltenen Alignierungen werden mit Modell 3/4 bewertet und die beste wird übernommen.
 - Dann wird rekursiv wieder versucht, durch kleine Änderungen noch bessere Alignierungen zu finden.
- Aus der besten Alignierung (oder den n besten Alignierungen) werden dann die Häufigkeiten extrahiert.

Wortalignierung mit den IBM-Modellen

- Training von Modell 1 auf dem satzalignierten Parallelkorpus
- Training von Modell 2 (Initialisierung mit Parametern von Modell 1)
- Training von Modell 3 (Initialisierung mit Modell 2)
- Training von Modell 4 (Initialisierung mit Modell 3)
- Ausgabe der wahrscheinlichsten Alignierung gemäß Modell 4

Alternative: Modell 1 \rightarrow HMM-Modell \rightarrow Modell 4

- Symmetrisierung von bidirektionalen Alignierungen
- Extraktion von Übersetzungsphrasen
- Phrasenbasierte Übersetzung

Wortalignierung

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

Hier zeigen schwarze Felder in der Matrix an, welche Wörter aligniert sind.

	john	biss	ins	grass
john				
kicked				
the				
bucket				

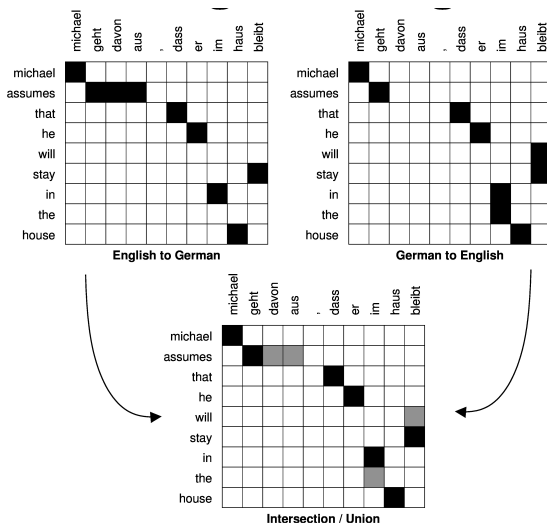
Wie sollten die beiden Idiome **kicked the bucket** und **biss ins gras** aligniert werden.

gras sollte normal nicht mit **bucket** übersetzt werden

- Die IBM-Modelle erlauben **n:1-Übersetzungen**, weil mehrere Quellwörter mit demselben Zielwort aligniert sein können.
- **1:n-Übersetzungen** sind nicht möglich, weil die Alignierung dann keine Funktion mehr ist.
- Tatsächlich braucht man sogar **n:m-Übersetzungen**
kicked the bucket – biss ins Gras

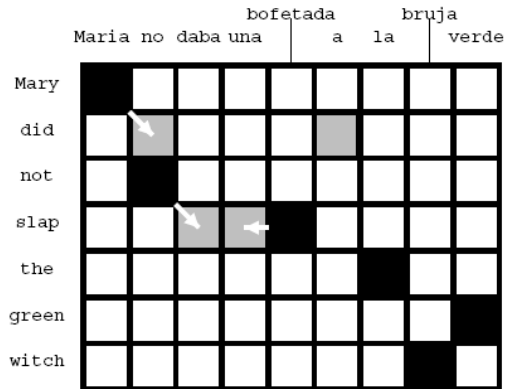
- Das Parallelkorpus wird zunächst mit den IBM-Modellen in beiden Richtungen aligniert.
- Dann werden die beiden Alignierungen zu einer neuen Alignierung “symmetrisiert”.

Symmetrisierung von Wortalignierungen



Berechnung der Schnittmenge und der Vereinigung der beiden Alignierungen

Symmetrisierung von Wortalignierungen



Hinzufügen weiterer Alignierungen zur Schnittmenge

Symmetrisierungs-Heuristik

grow_diag_final(e2f, f2e)

neighbouring = $\{(-1,0), (0,-1), (1,0), (0,1), (-1,-1), (-1,1), (1,-1), (1,1)\}$

alignment $A = \text{intersect}(e2f, f2e)$

// grow diag

while new points added **do**

for all English words $e \in [1...e_n]$, foreign words $f \in [1...f_n]$, $(e, f) \in A$ **do**

for all neighbouring alignment points $(e_{new}, f_{new}) \in \text{union}(e2f, f2e)$ **do**

if (e_{new} unaligned **or** f_{new} unaligned) **then**

 add (e_{new}, f_{new}) to A

// final

for all English words $e \in [1...e_n]$, foreign words $f \in [1...f_n]$, $(e, f) \in \text{union}(e2f, f2e)$ **do**

if (e_{new} unaligned **or** f_{new} unaligned)

 add (e_{new}, f_{new}) to A

GIZA++

- implementiert von Och & Ney
- trainiert nacheinander Modell 1, HMM und Modell 4 für jede Richtung
- dann Symmetrisierung
- verwendet von Moses, dem Standard-Toolkit zur Implementierung von SMÜ-Systemen

Alternative: fast_align

- Verbesserung von IBM Modell 2
- viel schneller und ähnlich gut (bei MÜ-Einsatz)

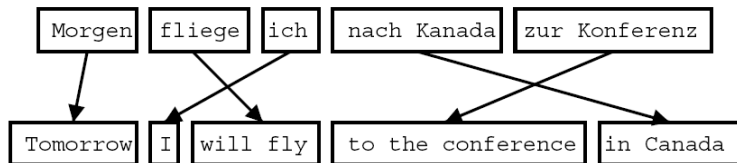
Wir haben zuletzt behandelt:

- Wortalignierung
- IBM-Modelle und HMM-Modell
- Symmetrisierung von Alignierungen

Als Nächstes kommt:

- phrasenbasierte SMÜ
 - Modellierung
 - Parameterschätzung
 - Dekodierung (Anwendung zur Übersetzung)

Phrasenbasierte SMÜ



- Der Quellsatz wird in Phrasen segmentiert
Eine “Phrase” ist hier eine beliebige Wortfolge, nicht eine linguistische Phrase
- Jede Phrase wird übersetzt.
- Dann werden die Phrasen umgeordnet.

Hauptkomponenten

- Phrasen-Übersetzungsmodell $\phi(f|e)$
- Umordnungsmodell
- Sprachmodell $p_{LM}(\mathbf{e})$ (mindestens ein Trigramm-Modell)
 $p(\text{Peter, lacht}) =$
 $p(\text{Peter}|\text{START}, \text{START}) p(\text{lacht}|\text{START}, \text{Peter}) p(\text{ENDE}|\text{Peter}, \text{lacht})$
- Bayes' Regel

$$\arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg \max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e}) = \arg \max_{\mathbf{e}} p_{LM}(\mathbf{e})\phi(\mathbf{f}|\mathbf{e})\omega^{|\mathbf{e}|}$$

- Der Satz \mathbf{f} wird in I Phrasen $F_1^I = F_1, \dots, F_I$ zerlegt (mit Wk. $\omega^{|\mathbf{e}|}$).
- Zerlegung von $\phi(\mathbf{f}|\mathbf{e})$:

$$\phi(F_1^I|E_1^I) = \prod_{i=1}^I \phi(F_i|E_i)d(a_i - b_{i-1})$$

a_i, b_i sind die Start- und Endposition der i-ten Phrase

Vorteile der phrasenbasierten Übersetzung

- Mit n:m-Übersetzungen können **Idiome** übersetzt werden
- Durch die größeren Übersetzungseinheiten kann **lokaler Kontext** berücksichtigt werden
- Je mehr Daten zur Verfügung stehen, desto **längere Phrasen** können gelernt werden

Phrasen-Übersetzungstabelle

Phrasenübersetzungen für: **den Vorschlag**

Englisch	$\phi(\mathbf{e} \mathbf{f})$	Englisch	$\phi(\mathbf{e} \mathbf{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.159	it	
the proposals	0.159

Erstellung der Phrasen-Übersetzungstabelle

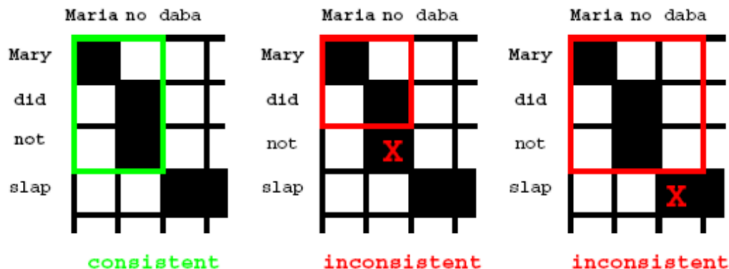
- symmetrisierte bidirektionale Wortalignierung

					bofetada			bruja	
	Maria	no	daba	una		a	la		verde
Mary	■	□	□	□	□	□	□	□	□
did	□	■	□	□	□	□	□	□	□
not	□	■	□	□	□	□	□	□	□
slap	□	□	■	■	■	□	□	□	□
the	□	□	□	□	□	■	■	□	□
green	□	□	□	□	□	□	□	□	■
witch	□	□	□	□	□	□	□	■	□

- Extraktion aller Phrasenpaare, die mit der Alignierung konsistent sind

Erstellung der Phrasen-Übersetzungstabelle

Die Phrasenpaare müssen zur Alignierung **konsistent** sein:



Konsistent bedeutet, dass kein Wort innerhalb der Phrase mit einem Wort außerhalb der Phrase aligniert ist.

Erstellung der Phrasen-Übersetzungstabelle

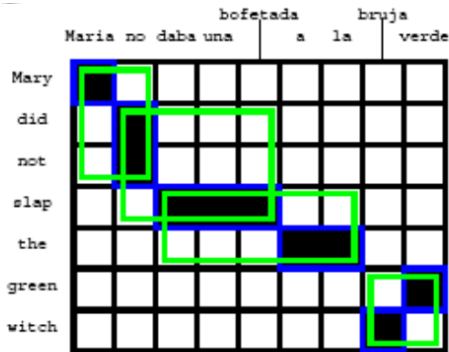
Die kleinsten mit der Alignierung konsistenten Phrasen

					bofetada		bruja	
	Maria	no	daba	una	a	la	verde	
Mary	■							
did		■						
not		■						
slap			■	■	■			
the						■	■	
green								■
witch							■	

Maria, Mary | no, did not | daba una bofetada, slap | a la, the | bruja, witch | verde, green

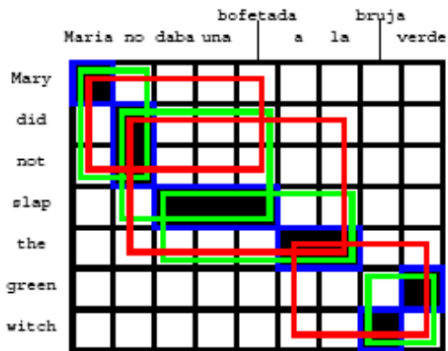
Erstellung der Phrasen-Übersetzungstabelle

Kombinationen von 2 minimalen Phrasen:



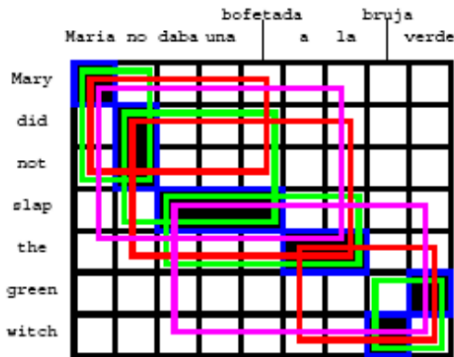
Maria, Mary | no, did not | daba una botefada, slap | a la, the | bruja, witch | verde, green
Maria no, Mary did not | no daba una botefada, did not slap | daba una botefada a la, slap the |
bruja verde, green witch

Erstellung der Phrasen-Übersetzungstabelle



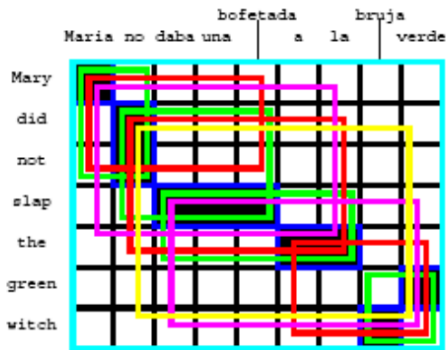
Maria, Mary | no, did not | daba una bofetada, slap | a la, the | bruja, witch | verde, green
Maria no, Mary did not | no daba una bofetada, did not slap | daba una bofetada a la, slap the |
bruja verde, green witch Maria no daba una bofetada, Mary did not slap | no daba una bofetada
a la, did not slap the | a la bruja verde, the green witch

Erstellung der Phrasen-Übersetzungstabelle



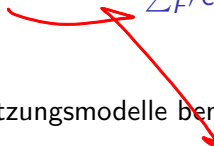
Maria, Mary | no, did not | daba una bofetada, slap | a la, the | bruja, witch | verde, green
Maria no, Mary did not | no daba una bofetada, did not slap | daba una bofetada a la, slap the |
bruja verde, green witch Maria no daba una bofetada, Mary did not slap | no daba una bofetada
a la, did not slap the | a la bruja verde, the green witch Maria no daba una bofetada, Mary did
not slap | no daba una bofetada a la, did not slap the | a la bruja verde, the green witch

Erstellung der Phrasen-Übersetzungstabelle



Maria, Mary | no, did not | daba una bofetada, slap | a la, the | bruja, witch | verde, green
Maria no, Mary did not | no daba una bofetada, did not slap | daba una bofetada a la, slap the |
bruja verde, green witch Maria no daba una bofetada, Mary did not slap | no daba una bofetada
a la, did not slap the | a la bruja verde, the green witch Maria no daba una bofetada, Mary did
not slap | no daba una bofetada a la, did not slap the | a la bruja verde, the green witch
no daba una bofetada a la bruja verde, did not slap the green witch Maria no daba una bofetada
a la bruja verde, Mary did not slap the green witch

Wahrscheinlichkeitsverteilung über Phrasenpaare

$$\phi(F|E) = \frac{\text{count}(F, E)}{\sum_{F'} \text{count}(F', E)}$$


Diskriminative Übersetzungsmodelle benutzen eventuell zusätzlich

- umgekehrte Wahrscheinlichkeit: $\phi(E|F) = \frac{\text{count}(F, E)}{\sum_{E'} \text{count}(F, E')}$
- lexikalisierte Übersetzungswahrscheinlichkeiten
berechnet mit IBM Modell 1

Umordnungsmodell

- Die ^{Distanz}
Kosten für das Umordnen von zwei Phrasen betragen

$$\underline{d(l) = z^l}$$

falls der Start der aktuellen Phrase um l Wortpositionen gegenüber dem Ende der vorhergehenden Phrase (absolut) verschoben ist.

- Dieses Umordnungsmodell ist sehr einfach und modelliert Umordnungen über weite Distanzen nicht sehr gut. Daher hilft oft ein Umordnungslimit, welches bspw. Umordnungen über mehr als 6 Wortpositionen verbietet.
- Wenn sich die Wortstellung in der Quell- und Zielsprache nicht unterscheidet, kann eine monotone Übersetzung sinnvoll sein.
bspw. bei Hindi ↔ Urdu
Hier hat das Umordnungslimit den Wert 0.

Lexikalisiertes Umordnungsmodell

- Bei der Übersetzung vom Spanischen oder Französischen ins Englische muss die Wortstellung von Adjektiven und Nomen oft vertauscht werden: **green witch – bruja verde**
- Die Wortstellung hängt hier also von den Wörtern/Wortarten ab.
- **Lexikalisierte Umordnungsmodelle** sagen für eine Phrase voraus,
 - ob Sie direkt auf die vorherige Phrase folgt (**monotone**)
 - ob Sie mit der vorherigen Phrase vertauscht wird (**swap**)
 - oder ob ein anderer Fall vorliegt (**discontinuous**)
- Die Wahrscheinlichkeiten der drei Möglichkeiten werden aus den Trainingsdaten geschätzt.

$$p_o(\text{orientation}|\mathbf{f}, \mathbf{e}) = \frac{\text{count}(\text{orientation}, \mathbf{f}, \mathbf{e})}{\sum_o \text{count}(o, \mathbf{f}, \mathbf{e})}$$

- Es muss also gezählt werden, wie oft jede Phrase mit welcher Reihenfolge in den Trainingsdaten aufgetreten ist.
- **Problem:** Wie kann ein solches Teilmodell in ein PBMT-Modell integriert werden?

Generative vs. Diskriminative Modelle

Bisher haben wir nur **generative Modelle** betrachtet:

- Ein aligniertes Satzpaar wird in vielen **Einzelschritten** generiert.
- Jeder Schritt hat eine **Wahrscheinlichkeit**.
- Die Wahrscheinlichkeiten werden multipliziert.
- Die Wahrscheinlichkeiten aller Satzpaare summieren zu 1.

Diskriminative Modelle

- definieren **Merkmalsfunktionen**, welche die Satzpaare charakterisieren
Diese Merkmale können auch Wahrscheinlichkeiten sein.
- Die Merkmalswerte werden mit **Gewichten** multipliziert.
- Die Summe der gewichteten Merkmale wird in eine Wahrscheinlichkeit transformiert.
- Es gibt keine Zerlegung in **Einzelschritte**.
- Beliebige Merkmalsfunktionen können einfach hinzugefügt werden.

- Wiederholung zum generativen phrasenbasierten Modell
- Parameteroptimierung
- Übergang zum diskriminativen Modell
- Optimierung der Merkmalsgewichte
- Hinzufügen weiterer Merkmalsfunktionen

Eigentlich möchten wir berechnen

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg \max_{\mathbf{e}} \sum_{\mathbf{a}} p(\mathbf{e})p(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

Diese Berechnung ist jedoch sehr schwierig. Daher berechnen wir stattdessen oft

$$\hat{\mathbf{e}}, \hat{\mathbf{a}} = \arg \max_{\mathbf{e}, \mathbf{a}} p(\mathbf{e})p(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

Wir berechnen also die wahrscheinlichste alignierte Übersetzung.

Generatives Phrasenbasiertes Übersetzungsmodell

$$p_{TM}(f, a|e) \quad p_D(a) \quad p_{LM}(e) \quad c^{|e|}$$

Handwritten annotations in red:

- Unordnung* (Unordering) above $p_D(a)$
- Alignment* (Alignment) below $p_{TM}(f, a|e)$
- Sprachmodell* (Language model) below $p_{LM}(e)$
- Ersetzung* (Replacement) below $p_D(a)$

Beispiel

Quellsatz: |Morgen| |fliege| |ich| |nach Kanada|

Übers. 1: |Tomorrow| |I| |will fly| |to Canada|

Übers. 2: |Tomorrow| |fly| |I| |to Canada|

Wir erwarten hier folgende Bewertungen:

	Phrasenübers.	Umordnung	Sprachm.	Längenbonus
Übers. 1	gut	$z^4 < 1$	gut	c^6
Übers. 2	gut	$z^0 = 1$	schlecht	$c^5 < c^6$

Welche Übersetzung wird besser bewertet?

- Das Sprachmodell und der Längenbonus präferieren Übersetzung 1.
- Das Umordnungsmodell präferiert Übersetzung 2.

Wir können versuchen, c und z so zu optimieren, dass Übersetzung 1 präferiert wird.

	Phrasenübers.	Umordnung	Sprachm.	Längenbonus
Übers. 1	gut	$z^4 < 1$	gut	c^6
Übers. 2	gut	$z^0 = 1$	schlecht	$c^5 < c^6$

Optimierung von z und c

- Nimm ein neues Übersetzungskorpus (dev-Daten).
- Probiere verschiedene Werte für z und c .
- Berechne jeweils den BLEU-Score und gib die beste Kombination zurück.

```
Best = 0
for z in {0.0, 0.1, 1.2, ..., 1.0}
    for c in {1.0, 1.1, 1.2, ..., 3.0}
        hyp = decode(z,c,dev)
        if BLEU(hyp) > Best
            Best = BLEU(hyp)
            BestParams = (z,c)
return BestParams
```

Hinzufügen von Gewichten

- Was können wir tun, wenn wir wissen, dass das Sprachmodell sehr gut oder schlecht ist?
- Wir können die Sprachmodell-Wk. zum Exponenten nehmen.

$$p_{LM}(\mathbf{e})^{\lambda_{LM}}$$

- $\lambda_{LM} > 1$: Das Sprachmodell ist gut und wichtig
- $\lambda_{LM} < 1$: Das Sprachmodell ist schlecht und unwichtig
- $\lambda_{LM} = 0$: Das Sprachmodell wird ignoriert.
- Dem Übersetzungsmodell können wir ebenfalls ein Gewicht geben:

$$p_{TM}(\mathbf{f}, \mathbf{a} | \mathbf{e})^{\lambda_{TM}} p_D(\mathbf{a}) p_{LM}(\mathbf{e})^{\lambda_{LM}} c^{|\mathbf{e}|}$$

Umformungen

$$\begin{array}{r} 0.5 \\ 0.3 \\ \hline 0.2 \\ 1 \end{array} \quad \begin{array}{r} 0.52 \\ 0.3 \\ \hline 0.25 \end{array} \quad \begin{array}{r} 0.5^2 / 0.75 \\ 0.3 / 0.25 \\ \hline 0.2 / 0.25 \\ 1 \end{array}$$

Im **Decoding** müssen wir folgenden Ausdruck berechnen:

$$\begin{aligned} \hat{\mathbf{e}}, \hat{\mathbf{a}} &= \arg \max_{\mathbf{e}, \mathbf{a}} p(\mathbf{e}, \mathbf{a} | f) \\ &= \arg \max_{\mathbf{e}, \mathbf{a}} \frac{p_{TM}(\mathbf{f}, \mathbf{a} | \mathbf{e})^{\lambda_{TM}} p_D(\mathbf{a}) p_{LM}(\mathbf{e})^{\lambda_{LM}} c^{|\mathbf{e}|}}{\sum_{\mathbf{e}', \mathbf{a}'} p_{TM}(\mathbf{f}, \mathbf{a}' | \mathbf{e}')^{\lambda_{TM}} p_D(\mathbf{a}') p_{LM}(\mathbf{e}')^{\lambda_{LM}} c^{|\mathbf{e}'|}} \end{aligned}$$

Die Konstante im Nenner hat keinen Einfluss auf das Ergebnis der Maximierung:

$$\hat{\mathbf{e}}, \hat{\mathbf{a}} = \arg \max_{\mathbf{e}, \mathbf{a}} p_{TM}(\mathbf{f}, \mathbf{a} | \mathbf{e})^{\lambda_{TM}} p_D(\mathbf{a}) p_{LM}(\mathbf{e})^{\lambda_{LM}} c^{|\mathbf{e}|}$$

Wir können statt der Funktion selbst auch ihren Logarithmus maximieren, da der Logarithmus eine monoton steigende Funktion ist:

$$\begin{aligned} \hat{\mathbf{e}}, \hat{\mathbf{a}} &= \arg \max_{\mathbf{e}, \mathbf{a}} \log(p_{TM}(\mathbf{f}, \mathbf{a} | \mathbf{e})^{\lambda_{TM}} p_D(\mathbf{a}) p_{LM}(\mathbf{e})^{\lambda_{LM}} c^{|\mathbf{e}|}) \\ &= \arg \max_{\mathbf{e}, \mathbf{a}} \log p_{TM}(\mathbf{f}, \mathbf{a} | \mathbf{e})^{\lambda_{TM}} + \log p_D(\mathbf{a}) + \log p_{LM}(\mathbf{e})^{\lambda_{LM}} + \log c^{|\mathbf{e}|} \end{aligned}$$

Umformungen

Mit $P_D(\mathbf{a}) = \prod_i z^{d_i}$ erhalten wir:

$$\begin{aligned}\hat{\mathbf{e}}, \hat{\mathbf{a}} &= \arg \max_{\mathbf{e}, \mathbf{a}} \log p_{TM}(\mathbf{f}, \mathbf{a}|\mathbf{e})^{\lambda_{TM}} + \log \prod_i z^{d_i} + \log p_{LM}(\mathbf{e})^{\lambda_{LM}} + \log c^{|e|} \\ &= \arg \max_{\mathbf{e}, \mathbf{a}} \lambda_{TM} \log p_{TM}(\mathbf{f}, \mathbf{a}|\mathbf{e}) + \sum_i d_i \log z + \lambda_{LM} \log p_{LM}(\mathbf{e}) + |e| \log c\end{aligned}$$

Reparametrisierung: $\lambda_D := \log z$ $\lambda_{LB} := \log c$

$$\hat{\mathbf{e}}, \hat{\mathbf{a}} = \arg \max_{\mathbf{e}, \mathbf{a}} \lambda_{TM} \log p_{TM}(\mathbf{f}, \mathbf{a}|\mathbf{e}) + \lambda_D \sum_i d_i + \lambda_{LM} \log p_{LM}(\mathbf{e}) + \lambda_{LB} |e|$$

Statt eines Längenbonus $\lambda_{LB}|e|$ wird oft ein Length Penalty verwendet: $\lambda_{LP}(-|e|)$.

Vorteil: Alle Merkmalsfunktionen haben einheitlich negative Werte.

Der Wert von λ_{LP} wird negativ sein, während λ_{LB} positiv war.

Analog: $\lambda_{D'} \sum_i -d_i$ ersetzt $\lambda_D \sum_i d_i$

$$\text{score}(\mathbf{e}, \mathbf{a}, \mathbf{f}) = \lambda_{TM} \log p_{TM}(\mathbf{f}, \mathbf{a}|\mathbf{e}) + \lambda_D \sum_i d_i + \lambda_{LM} \log p_{LM}(\mathbf{e}) + \lambda_{LB} |\mathbf{e}|$$

Wegen der Gewichte bekommen wir keine Wahrscheinlichkeitsverteilung, wenn wir die obigen Werte zum Exponenten nehmen.

Stattdessen müssen wir die Softmax-Funktion anwenden, um aus den Werten (bedingte) Wahrscheinlichkeiten zu machen:

$$\begin{aligned} \text{score}(\mathbf{e}, \mathbf{a}|\mathbf{f}) &= \frac{1}{Z} e^{\text{score}(\mathbf{f}, \mathbf{a}, \mathbf{e})} \\ Z &= \sum_{\mathbf{e}, \mathbf{a}} e^{\text{score}(\mathbf{f}, \mathbf{a}, \mathbf{e})} \end{aligned}$$

- Das erhaltene diskriminative Modell besitzt 4 Merkmalsfunktionen und 4 λ -Gewichte.
- Die Gewichte müssen **trainiert** werden.
- Dazu dürfen nicht die Daten verwendet werden, aus denen die Phrasentabelle extrahiert wurde, da die Gewichte für **neue Daten** optimal sein sollen.
- Stattdessen nehmen wir separate **Development**-Daten, die klein sein können (bspw. 1000 Satzpaare).

Wir können für die Optimierung ein Grid-Search verwenden:

~~10⁴~~

- Wähle für jeden λ -Parameter bspw. 10 verschiedene sinnvolle Werte
- Probiere alle 10^4 möglichen Kombinationen aus:
 - Übersetze die Development-Daten mit jeder möglichen Kombination von λ -Werten.
 - Berechne den BLEU-Score der Übersetzungen.
 - Wähle die Kombination mit dem höchsten BLEU-Score

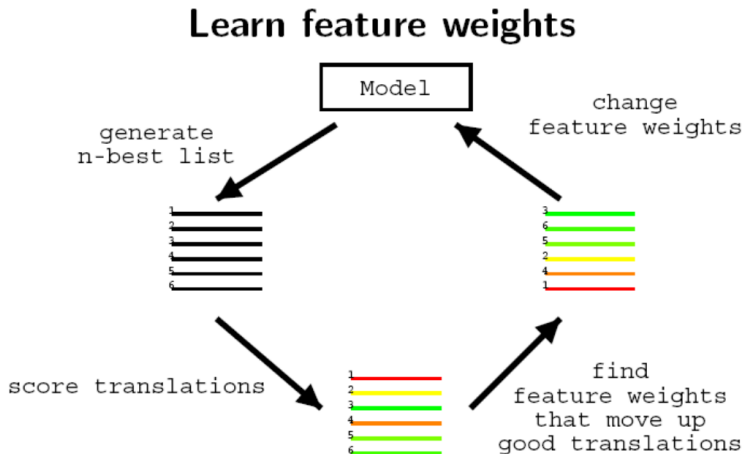
Alternative: **zufällige Wahl** der λ -Werte

- Für n Iterationen
 - Wähle für jeden λ -Parameter einen zufälligen Wert aus dem erlaubten Bereich.
 - Übersetze die Development-Daten mit den erhaltenen λ -Werten.
 - Berechne den BLEU-Score.
 - Speichere den λ -Vektor, falls der BLEU-Score der bisher beste war.
- Vorteile:
 - Es werden mehr unterschiedliche Werte für jeden Parameter probiert.
 - Das Training kann jederzeit beendet werden, ohne dass ein Teil des Suchraumes gar nicht untersucht wurde.

noch bessere Methode: MERT

Idee 1: N-Best-Übersetzungen

- Der aufwändigste Teil des Trainings ist das Übersetzen.
- Mit folgender Methode können wir die Zahl der Übersetzungen reduzieren:
 1. Wähle einen λ -Vektor bspw. (1,1,1,1)
 2. $Hyp = \{\}$
 3. Generiere die z.B. 100 besten Übersetzungen mit diesen λ -Werten und füge Sie zu Hyp hinzu.
 4. Optimierte den λ -Vektor auf den Übersetzungen in Hyp.
 5. Ersetze den ursprünglichen λ -Vektor durch den optimierten.
 6. Weiter mit 3., falls sich der neue λ -Vektor vom alten unterschied.



Quellsatz: |Morgen| |fliege| |ich| |nach Kanada|

Übers. 1: |Tomorrow| |I| |will fly| |to Canada|

Übers. 2: |Tomorrow| |fly| |I| |to Canada|

Angenommen Übers. 1 hat den besseren BLEU-Score.

	Phrasenübers.	Umordnung	Sprachm.	Längenbonus
Übers. 1	-1	-4	-3	-6
Übers. 2	-1	0	-5	-5

Angenommen wir starten mit dem Vektor $(1,1,1,-1)$

Score von Übers. 1 $= 1 * (-1) + 1 * (-4) + 1 * (-3) - 1 * (-6) = -2$

Score von Übers. 2 $= 1 * (-1) + 1 * 0 + 1 * (-5) - 1 * (-5) = -1$

Die schlechte Übers. 2 wird besser bewertet!

	Phrasenübers.	Umordnung	Sprachm.	Längenbonus
Übers. 1	-1	-4	-3	-6
Übers. 2	-1	0	-5	-5

Wir halbieren den Reordering Penalty und verdoppeln das Gewicht des Sprachmodelles: (1,0.5,2,-1)

Score von Übers. 1 = $1 * (-1) + 0.5 * (-4) + 2 * (-3) - 1 * (-6) = -3$

Score von Übers. 2 = $1 * (-1) + 0.5 * 0 + 2 * (-5) - 1 * (-5) = -6$

Nun wird die gute Übers. 1 besser bewertet!

	Phrasenübers.	Umordnung	Sprachm.	Längenbonus
Übers. 1	-1	-4	-3	-6
Übers. 2	-1	0	-5	-5

N-best-Listen enthalten mehrere Sätze und mehrere Übersetzungen pro Satz.

Der λ -Vektor (1, 0.5, 2, -1) wählt Übers. 1 für den ersten Satz und Übers. 2 für den zweiten Satz.

Angenommen Übers. 1 von Satz 2 ist besser.

Wir modifizieren den λ -Vektor zu (3, 0.5, 2, -1)

Nun wird auch bei Satz 2 die gute Übers. 1 besser bewertet!

Satz	Übersetzung	Phrasenübers.	Umordnung	Sprachm.	Längenb.
Satz 1	Übers. 1	-1	-4	-3	-6
Satz 1	Übers. 2	-1	0	-5	-5
Satz 2	Übers. 1	-1	-4	-3	-6
Satz 2	Übers. 2	-1	0	-5	-5

- Wir haben besprochen, wie man die λ -Werte trainiert
 - Je nach Korpus wird bspw. das Umordnen mehr oder weniger bestraft.
 - Dies wird automatisch aus den Developmentdaten gelernt.
- Wie fügen wir nun weitere Merkmalsfunktionen hinzu?

- Neue Merkmalsfunktionen werden einfach mit einem weiteren λ -Wert multipliziert und zum Score der Übersetzung hinzuaddiert.
- Die Merkmalsfunktion kann einen beliebigen numerischen Wert berechnen.
- Die Merkmalsfunktion kann beliebig komplex sein
 - einfach wie der Längenbonus oder
 - komplex wie die Phrasentabelle
- Mit passenden λ -Gewichten werden die neuen Merkmale optimal integriert.

- Die Merkmalsfunktionen dürfen **überlappen**.
- Wir können beispielsweise vier Übersetzungswahrscheinlichkeiten gleichzeitig verwenden: $\phi(\mathbf{e}|\mathbf{f})$, $\phi(\mathbf{f}|\mathbf{e})$, $\phi_{lex}(\mathbf{e}|\mathbf{f})$, $\phi_{lex}(\mathbf{f}|\mathbf{e})$
- In generativen Modellen ist das nicht möglich, weil jedem Schritt genau eine Wahrscheinlichkeit entspricht.
Wir könnten $\phi(\mathbf{f}|\mathbf{e})$ und $\phi_{lex}(\mathbf{f}|\mathbf{e})$ kombinieren, wenn wir einen weiteren Schritt einbauen, der (zufällig) zwischen den beiden Teilmodellen auswählt, aber wir können auf keinen Fall $\phi(\mathbf{e}|\mathbf{f})$ integrieren.

Relevante Informationsquellen:

- Sprachmodell
- Phrasenübersetzungstabellen
- Umordnungsmodelle
- Zahl der Wörter
- Wortübersetzungstabellen
- Zahl der nicht übersetzten Wörter
- Zahl der Phrasen
- Phrasenpaarhäufigkeit
- zusätzliche Sprachmodelle
- ...

- Je mehr Merkmalsfunktionen hinzugefügt werden, desto länger wird der λ -Vektor.
- Der Aufwand für die Parameteroptimierung mit GridSearch steigt exponentiell mit der Zahl der Merkmale.
- Mit **MERT**-Training kann der Aufwand reduziert werden.

Minimum Error Rate Training (MERT)

- versucht den Übersetzungen mit hohem BLEU-Score möglichst hohe Modell-Scores zuzuweisen
- optimiert iterativ einzelne λ -Werte unabhängig voneinander
- entwickelt von Franz Och
- implementiert in Moses
- funktioniert gut mit bis zu etwa 20 Merkmalsfunktionen
- sehr schnell

MERT-Algorithmus

gegeben: Sätze mit n-Best-Übersetzungen

Algorithmus:

Iteriere T -mal

Wähle zufällig einen Gewichtsvektor

Wiederhole bis zur Konvergenz

Für jedes Merkmal

Bestimme sein bestes Gewicht (\rightarrow nächste Folie)

Aktualisiere das Gewicht

Gib den Gewichtsvektor aus dem besten Trainingslauf zurück.

Je nachdem, mit welchem Vektor gestartet wird, ergibt sich ein anderes Ergebnis.
Daher werden mehrere Trainingsläufe durchgeführt.

Optimierung eines Gewichtes λ_k

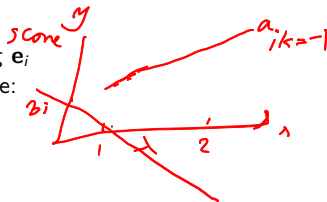
- Bei der Optimierung von λ_k bleiben die übrigen Gewichte fix.
- Den Score der Übersetzung \mathbf{e}_i für Satz \mathbf{f} können wir schreiben mit:

$$p(\mathbf{e}_i|\mathbf{f}) = \lambda_k a_{ik} + b_i$$

a_{ik} ist der Wert des k-ten Merkmal in Übersetzung \mathbf{e}_i

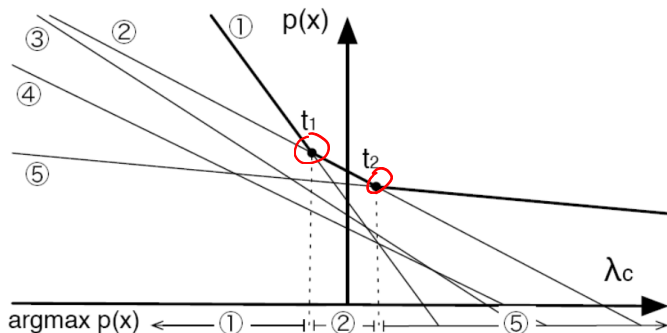
b_i ist die Summe der gewichteten übrigen Merkmale:

$$b_i = \sum_{j \neq k} \lambda_j a_{ij}$$



- Wir trainieren auf je über 100 Übersetzungen von 1000 Sätzen.
- Wir suchen den Wert von λ_k , bei dem der BLUE-Score für die am höchsten bewerteten Übersetzungen aller Sätze maximal wird.

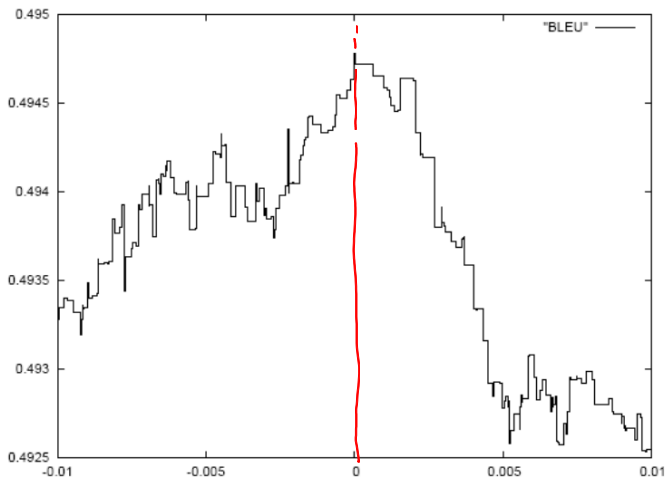
Übersetzungen eines Satzes



- Jede Übersetzung entspricht eine Linie $\lambda_k a_{ik} + b_i$
- Die Übersetzung mit dem höchsten Modell-Score ist die oberste Linie an jedem Punkt der x-Achse.
- Die beste Übersetzung ändert sich an den Grenzpunkten t_j .

BLEU-Score in Abhängigkeit von λ_k

BLEU-Score der vom Modell am besten bewerteten Übersetzungen in Abhängigkeit von λ_k :



Bestimmung des optimalen Wertes für λ_k

- Zwischen zwei Grenzpunkten bleibt die beste Übersetzung dieselbe.
- Wir müssen also das beste Intervall finden.

Algorithmus

Suche die Grenzpunkte (\rightarrow nächste Folie)

Für jedes Intervall zwischen Grenzpunkten

Suche die beste Übersetzung

Berechne ihren BLEU-Score

Wähle das Intervall mit dem höchsten BLEU-Score



Bestimmung der Grenzpunkte

- Jeder Grenzpunkt ist der Schnittpunkt von zwei Geraden.
- Wir berechnen die Schnittpunkte aller Geraden:

$$\begin{aligned}\lambda a_1 + b_1 &= \lambda a_2 + b_2 \\ \lambda a_1 - \lambda a_2 &= b_2 - b_1 \\ \lambda(a_1 - a_2) &= b_2 - b_1 \\ \lambda &= \frac{b_2 - b_1}{a_1 - a_2}\end{aligned}$$

- Für jedes Satzpaar berechnen wir auf diese Weise den Schnittpunkt der Geraden.

Andere Trainingsmethoden

Bei einer großen Zahl von Merkmalen (> 20) sind andere Trainingsverfahren wie **Gradientenanstieg** besser geeignet.

Ziel: Maximierung der Likelihood $L(D)$ der Trainingsdaten D

$$L(D) = \sum_{(\mathbf{e}, \mathbf{f}) \in D} p(\mathbf{e}, \mathbf{f})$$

Gradientenanstieg:

Initialisiere λ

Für T Iterationen

$$\lambda = \lambda + \eta \nabla_{\lambda} L(D)$$

- $\nabla_{\lambda} L(D)$ ist die Menge der partiellen Ableitungen von $L(D)$ nach den λ_i
- η ist die Lernrate.

Der Gradient $\nabla L(D)$ ergibt sich aus der Differenz zwischen den beobachteten und erwarteten Werten der Merkmalsfunktionen f_k :

$$\sum_{(\mathbf{e}, \mathbf{f}) \in D} f_k(\mathbf{e}, \mathbf{f}) - \sum_{\mathbf{f}} \sum_{\mathbf{e}'} p(\mathbf{e}' | \mathbf{f}) f_k(\mathbf{e}', \mathbf{f})$$

Da wir nicht über alle möglichen Übersetzungen e' iterieren können, approximieren wir die Menge aller Übersetzungen mit der n-best-Liste.

Um Overfitting zu vermeiden, können wir **Regularisierung** anwenden:

- L_2 Regularisierung: $L(D) - \lambda^2$
- L_1 Regularisierung: $L(D) - |\lambda|$

Die Regularisierung bestraft große Gewichte.

Wenn die Referenzübersetzung aus einem Trainingsbeispiel mit der gegebenen Phrasentabelle nicht generiert werden kann, ist Ihre Wahrscheinlichkeit unabhängig von den Gewichten immer 0.

In diesem Fall ersetzt man die Referenzübersetzung durch die Übersetzung aus der n-best-Liste mit dem höchsten BLEU-Score relativ zur Referenzübersetzung, der **Oracle**-Übersetzung.

Margin Infused Relaxed Algorithm (MIRA)

funktioniert meist noch besser als Gradientenabstieg.

Ziel: Der nicht normalisierte Modell-Score der gewünschten Übersetzung soll mindestens um 1 größer sein als die Scores aller anderen Übersetzungen.

Wenn das nicht der Fall ist, werden die Gewichte so modifiziert, dass die gewünschte Übersetzung höher bewertet wird.

⇒ ähnlich dem Perzeptron-Training

- Wir haben log-lineare Übersetzungsmodelle betrachtet und
- die Optimierung dieser Modelle (MERT, Gradientenanstieg, MIRA)
- In log-lineare Modelle können beliebige Merkmale integriert werden.
- Es sollte aber möglich, die Merkmale auch für Teilübersetzungen zu berechnen (für die Suche im Decoding).
- nächstes Thema: Decoding

Decoding

- Welche Merkmale werden bei der phrasenbasierten Übersetzung (PBMT) benutzt?
- Wie wird die Bewertung einer Übersetzung berechnet?
- Wie wird die beste Übersetzung berechnet? (Decoding)
 - Überblick über den Übersetzungsprozess
 - effiziente Übersetzung mit der Beam Search (Strahlsuche)
- andere Übersetzungsalgorithmen

Die Bewertung einer Übersetzung ist die gewichtete Summe der Merkmalsfunktionen

$$p(\mathbf{e}, \mathbf{a} | \mathbf{f}) \propto e^{\sum_i \lambda_i f_i(\mathbf{e}, \mathbf{a}, \mathbf{f})}$$

proportional zu

f_i Merkmalsfunktionen

λ_i Merkmalsgewichte

Typische in PBMT Merkmalsfunktionen

- Phrasentabellen-Wahrscheinlichkeiten
 $p_{TM}(\mathbf{e}|\mathbf{f})$ und $p_{TM}(\mathbf{f}|\mathbf{e})$
- lexikalische Übersetzungswahrscheinlichkeiten
 $p_{lex}(\mathbf{e}|\mathbf{f})$ und $p_{lex}(\mathbf{f}|\mathbf{e})$
- Sprachmodell-Wahrscheinlichkeiten
 $p_{LM}(\mathbf{e})$
- Wortbonus
- Phrase Penalty
- Distortion Penalty

Problem: Viele der extrahierten Phrasen sind selten:

$$p(\text{"ein blauer Bus landet auf dem Mars"} | \text{"a blue bus lands on Mars"}) = 1$$

$$p(\text{"a blue bus lands on Mars"} | \text{"ein blauer Bus landet auf dem Mars"}) = 1$$

Ist diese Schätzung zuverlässig?

Automatisch extrahierte Phrasenpaare sind aufgrund von Alignierungsfehlern oft fehlerhaft:

$$p(\text{"; distortion carried - over"} | \text{"; Verzerrung"}) = 1$$

$$p(\text{"; Verzerrung"} | \text{"; distortion carried - over"}) = 1$$

⇒ Wir sollten uns nicht allein auf diese Wahrscheinlichkeiten.

⇒ Gefahr des Overfitting wenn die Häufigkeiten sehr klein sind

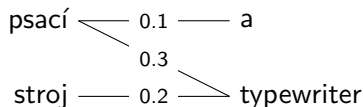
Ziel: Wahrscheinlichkeitsverteilung definiert auf Basis der einzelnen Wörter

- Da wir die einzelnen Wörter meist häufiger gesehen haben als die Phrasen, ist die Gefahr von Overfitting geringer.
- Die Phrasenpaare wurden aus wortalignierten Daten extrahiert.
- Wir merken uns für jedes Phrasenpaar die häufigste Alignment.
- Dann berechnen wir eine Übersetzungswahrscheinlichkeit:

$$p_{lex}(f_1^N | a_1^N, e_1^M) = \prod_{i=1}^N \frac{1}{|B_i|} \sum_{j \in B_i} p(f_i | e_j)$$

B_i ist die Menge der Wörter, die mit f_i aligniert sind.

$$p_{lex}(f_1^N | a_1^N, e_1^M) = \prod_{i=1}^N \frac{1}{|B_i|} \sum_{j \in B_i} p(f_i | e_j)$$



$$p_{lex}(\text{"a typewriter"} | \text{"psací stroj"}) = \frac{1}{1} \cdot 0.1 \cdot \frac{1}{2}(0.3 + 0.2) = 0.025$$

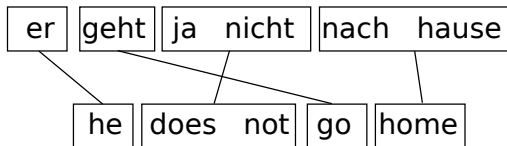
- Maschinelle Übersetzungen tendieren dazu, zu kurz zu sein.
- Wir brauchen einen Mechanismus, um dem entgegenzuwirken.
- Der **Wortbonus** addiert für jedes Wort den Betrag λ_{LB} zum Übersetzungs-Score.
- Je nach Wahl von λ_{LB} werden kürzere oder längere Sätze präferiert.

- Hier wird für jede Phrase der Betrag λ_{PB} zum Score addiert.
- Je nach Wahl von λ_{PB} werden entweder
 - wörtlichere Übersetzungen (mit vielen, kurzen Phrasen) oder
 - idiomatischere Übersetzungen (mit wenigen, langen Phrasen)präferiert.

Distortion Penalty (Umordnungsstrafe)

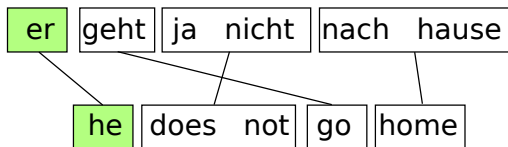
- einfachstes Umordnungsmodell
- Kann für manche Sprachpaare ausreichend sein
bspw. Englisch → Tschechisch
- Differenz zwischen der Endposition der vorherigen Phrase plus 1 und der Startposition der aktuellen Phrase (multipliziert mit dem Gewicht)

Bewertung einer Übersetzung



$$\text{score}(e|f) = 0$$

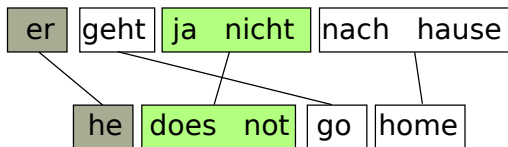
Bewertung einer Übersetzung



$$\begin{aligned} \text{score}(e|f) + &= \lambda_{TM} \cdot \log P_{TM}(\text{"he"}|\text{"er"}) \\ &+ \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}(\text{"er"}|\text{"he"}) \\ &+ \lambda_{lex} \cdot \log P_{lex}(\text{"he"}|\text{"er"}) \\ &+ \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}(\text{"er"}|\text{"he"}) \\ &+ \lambda_D \cdot 0 \\ &+ \lambda_{WP} \cdot 1 \\ &+ \lambda_{PP} \cdot 1 \\ &+ \lambda_{LM} \cdot \log P_{LM}(\text{"he"}|\text{"<S>"}) \end{aligned}$$

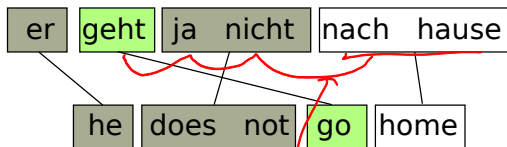
g: De
z: en

Bewertung einer Übersetzung



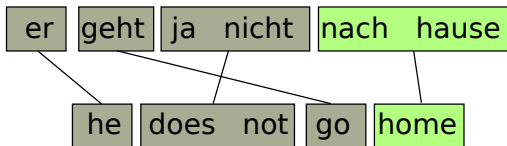
$$\begin{aligned} \text{score}(e|f) + &= \lambda_{TM} \cdot \log P_{TM}(\text{"does not"} | \text{"ja nicht"}) \\ &+ \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}(\text{"ja nicht"} | \text{"does not"}) \\ &+ \lambda_{lex} \cdot \log P_{lex}(\text{"does not"} | \text{"ja nicht"}) \\ &+ \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}(\text{"ja nicht"} | \text{"does not"}) \\ &+ \lambda_D \cdot 1 \\ &+ \lambda_{WP} \cdot 2 \\ &+ \lambda_{PP} \cdot 1 \\ &+ \lambda_{LM} \cdot \log P_{LM}(\text{"does not"} | \text{"<S>he"}) \end{aligned}$$

Bewertung einer Übersetzung



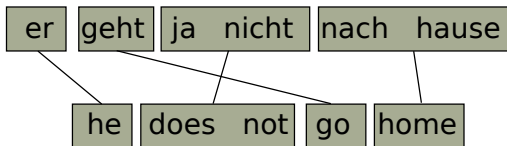
$$\begin{aligned} \text{score}(e|f) + &= \lambda_{TM} \cdot \log P_{TM}(\text{"go"}|\text{"geht"}) \\ &+ \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}(\text{"geht"}|\text{"go"}) \\ &+ \lambda_{lex} \cdot \log P_{lex}(\text{"go"}|\text{"geht"}) \\ &+ \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}(\text{"geht"}|\text{"go"}) \\ &+ \lambda_D \cdot 3 \\ &+ \lambda_{WP} \cdot 1 \\ &+ \lambda_{PP} \cdot 1 \\ &+ \lambda_{LM} \cdot \log P_{LM}(\text{"go"}|\text{"does not"}) \end{aligned}$$

Bewertung einer Übersetzung



$$\text{score}(e|f)_+ = \dots$$

Bewertung einer Übersetzung



$$\text{score}(e|f)_+ = \dots$$

- Wir haben ein mathematisches Modell für die Übersetzung

$$p(\mathbf{e}|\mathbf{f})$$

- Decoding-Aufgabe: Finde die wahrscheinlichste Übersetzung $\hat{\mathbf{e}}$

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f})$$

- Zwei Arten von Fehlern
 - Die wahrscheinlichste Übersetzung ist schlecht \Rightarrow Modell verbessern
 - Die wahrscheinlichste Übersetzung nicht gefunden \Rightarrow Suche verbessern
- Beim Decoding geht es darum, die Suchfehler zu minimieren, nicht die Qualität der Übersetzungen zu maximieren
wobei die beiden meistens (aber nicht immer) korrelieren

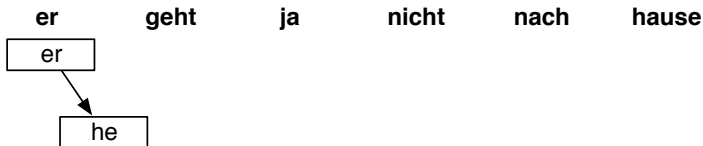
Translation Process

- Task: translate this sentence from German into English

er geht ja nicht nach hause

Translation Process

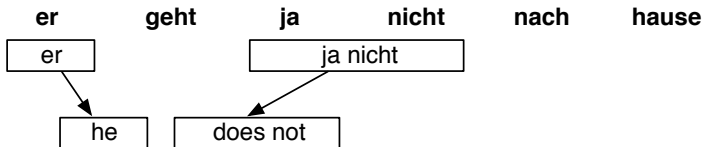
- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Process

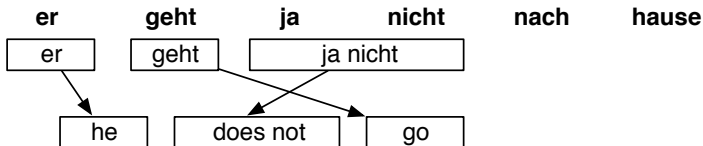
- Task: translate this sentence from German into English



- Pick phrase in input, translate
 - it is allowed to pick words out of sequence reordering
 - phrases may have multiple words: many-to-many translation

Translation Process

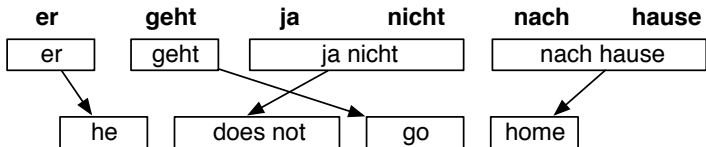
- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Process

- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Options

er	geht	ja	nicht	nach	hause
he	is	yes	not	after	house
it	are	is	do not	to	home
, it	goes	, of course	does not	according to	chamber
, he	go	,	is not	in	at home
it is		not		home	
he will be		is not		under house	
it goes		does not		return home	
he goes		do not		do not	
	is		to		
	are		following		
	is after all		not after		
	does		not to		
	not				
	is not				
	are not				
	is not a				

- Many translation options to choose from
 - in Europarl phrase table: 2727 matching phrase pairs for this sentence
 - by pruning to the top 20 per phrase, 202 translation options remain

Translation Options

er	geht	ja	nicht	nach	hause
he	is	yes	not	after	house
it	are	is	do not	to	home
, it	goes	, of course	does not	according to	chamber
, he	go		is not	in	at home
it is		not		home	
he will be		is not		under house	
it goes		does not		return home	
he goes		do not		do not	
	is		to		
	are		following		
	is after all		not after		
	does		not to		
	not				
	is not				
	are not				
	is not a				

- The machine translation decoder does not know the right answer
 - picking the right translation options
 - arranging them in the right order

→ Search problem solved by heuristic beam search

Decoding: Precompute Translation Options



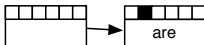
consult phrase translation table for all input phrases

Decoding: Start with Initial Hypothesis



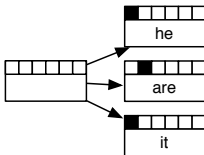
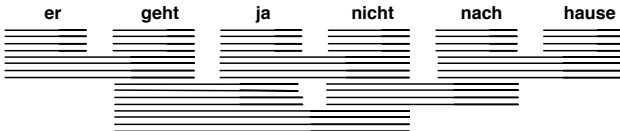
initial hypothesis: no input words covered, no output produced

Decoding: Hypothesis Expansion



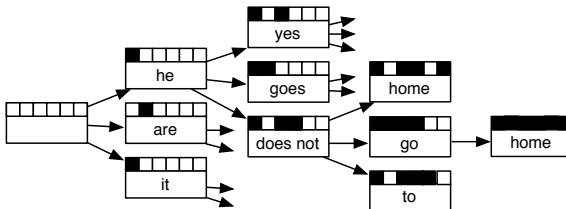
pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



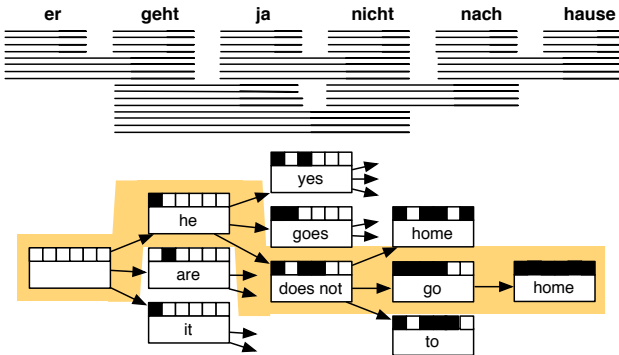
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



backtrack from highest scoring complete hypothesis

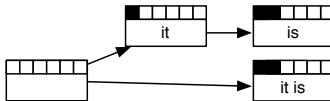
Computational Complexity

- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
 - recombination (risk-free)
 - pruning (risky)

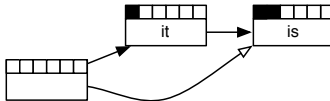
剪枝

Recombination

- Two hypothesis paths lead to two matching hypotheses
 - same number of foreign words translated
 - same English words in the output
 - different scores

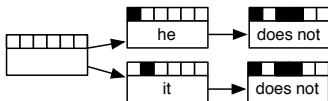


- Worse hypothesis is dropped

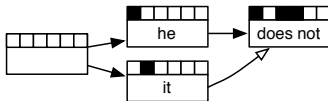


Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
 - same number of foreign words translated
 - same last two English words in output (assuming trigram language model)
 - same last foreign word translated
 - different scores



- Worse hypothesis is dropped



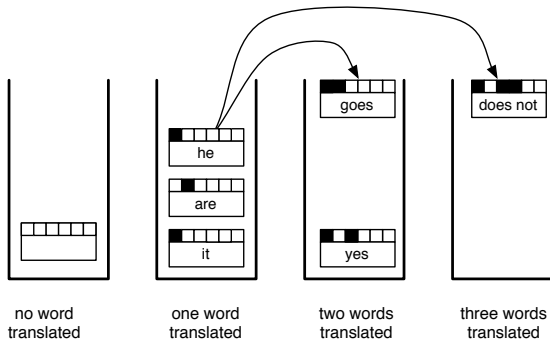
Restrictions on Recombination

- **Translation model:** Phrase translation independent from each other
→ no restriction to hypothesis recombination
- **Language model:** Last $n - 1$ words used as history in n -gram language model
→ recombined hypotheses must match in their last $n - 1$ words
- **Reordering model:** Distance-based reordering model based on distance to end position of previous input phrase
→ recombined hypotheses must have that same end position
- Other feature function may introduce additional restrictions

Pruning

- Recombination reduces search space, but not enough
(we still have a NP complete problem on our hands)
- Pruning: remove bad hypotheses early
 - put comparable hypothesis into stacks
(hypotheses that have translated same number of input words)
 - limit number of hypotheses in each stack

Stacks



- Hypothesis expansion in a stack decoder
 - translation option is applied to hypothesis
 - new hypothesis is dropped into a stack further down

Stack Decoding Algorithm

```
1: place empty hypothesis into stack 0
2: for all stacks  $0 \dots n - 1$  do
3:   for all hypotheses in stack do
4:     for all translation options do
5:       if applicable then
6:         create new hypothesis
7:         place in stack
8:         recombine with existing hypothesis if possible
9:         prune stack if too big
10:      end if
11:    end for
12:  end for
13: end for
```


Pruning

- Pruning strategies
 - histogram pruning: keep at most k hypotheses in each stack
 - stack pruning: keep hypothesis with score $\alpha \times$ best score ($\alpha < 1$)
- Computational time complexity of decoding with histogram pruning

$$O(\text{max stack size} \times \text{translation options} \times \text{sentence length})$$

- Number of translation options is linear with sentence length, hence:

$$O(\text{max stack size} \times \text{sentence length}^2)$$

- Quadratic complexity

Reordering Limits

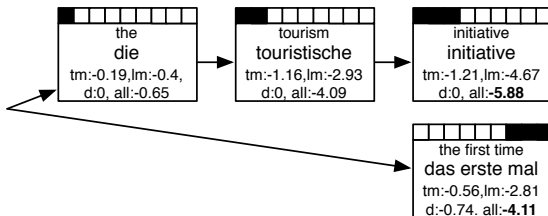
- Limiting reordering to maximum reordering distance
- Typical reordering distance 5–8 words
 - depending on language pair
 - larger reordering limit hurts translation quality
- Reduces complexity to linear

$$O(\text{max stack size} \times \text{sentence length})$$

- Speed / quality trade-off by setting maximum stack size

Translating the Easy Part First?

the tourism initiative addresses this for the first time

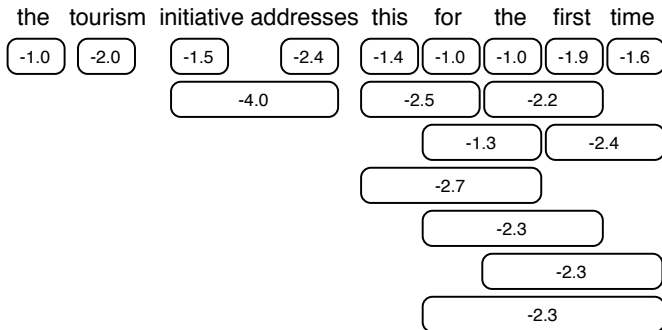


both hypotheses translate 3 words
worse hypothesis has better score

Estimating Future Cost

- Future cost estimate: how expensive is translation of rest of sentence?
- Optimistic: choose cheapest translation options
- Cost for each translation option
 - **translation model**: cost known
 - **language model**: output words known, but not context
→ estimate without context
 - **reordering model**: unknown, ignored for future cost estimation

Cost Estimates from Translation Options



cost of cheapest translation options for each input span (log-probabilities)

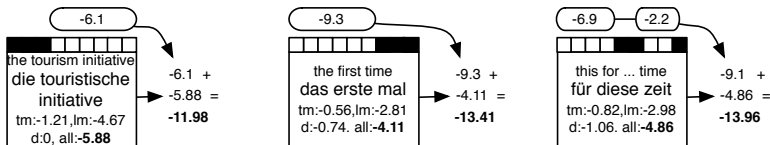
Cost Estimates for all Spans

- Compute cost estimate for all contiguous spans by combining cheapest options

first word	future cost estimate for n words (from first)								
	1	2	3	4	5	6	7	8	9
the	-1.0	-3.0	-4.5	-6.9	-8.3	-9.3	-9.6	-10.6	-10.6
tourism	-2.0	-3.5	-5.9	-7.3	-8.3	-8.6	-9.6	-9.6	
initiative	-1.5	-3.9	-5.3	-6.3	-6.6	-7.6	-7.6		
addresses	-2.4	-3.8	-4.8	-5.1	-6.1	-6.1			
this	-1.4	-2.4	-2.7	-3.7	-3.7				
for	-1.0	-1.3	-2.3	-2.3					
the	-1.0	-2.2	-2.3						
first	-1.9	-2.4							
time	-1.6								

- Function words cheaper (the: -1.0) than content words (tourism -2.0)
- Common phrases cheaper (for the first time: -2.3) than unusual ones (tourism initiative addresses: -5.9)

Combining Score and Future Cost

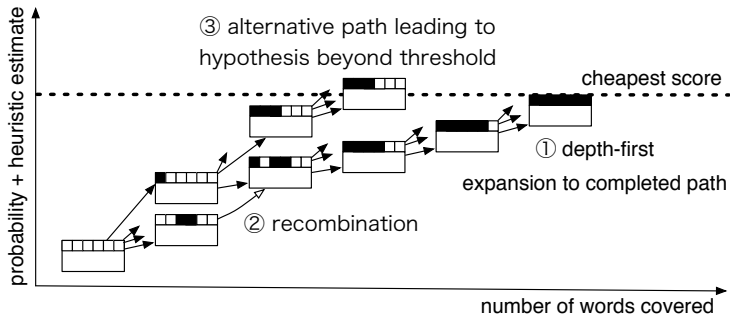


- Hypothesis score and future cost estimate are combined for pruning
 - left hypothesis starts with hard part: **the tourism initiative**
score: -5.88, future cost: -6.1 → total cost -11.98
 - middle hypothesis starts with easiest part: **the first time**
score: -4.11, future cost: -9.3 → total cost -13.41
 - right hypothesis picks easy parts: **this for ... time**
score: -4.86, future cost: -9.1 → total cost -13.96

Other Decoding Algorithms

- A* search
- Greedy hill-climbing
- Using finite state transducers (standard toolkits)

A* Search



- Uses *admissible* future cost heuristic: never overestimates cost
- Translation agenda: create hypothesis with lowest score + heuristic cost
- Done, when complete hypothesis created

Greedy Hill-Climbing

- Create one complete hypothesis with depth-first search (or other means)
- Search for better hypotheses by applying change operators
 - change the translation of a word or phrase
 - combine the translation of two words into a phrase
 - split up the translation of a phrase into two smaller phrase translations
 - move parts of the output into a different position
 - swap parts of the output with the output at a different part of the sentence
- Terminates if no operator application produces a better translation

- Standardmerkmale in PBMT
- Berechnung des Übersetzungs-Scores
- Überblick über den Übersetzungsprozess
- Beam Search
 - Zusammenfassung (Recombination) von Hypothesen
 - Pruning
 - Begrenzung der Umordnung (Distortion Limit)
 - zukünftige Kosten (Future Cost)
- andere Decoding-Algorithmen