Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Übersetzung

Tobias Scheffer
Michael Großhans
Paul Prasse
Uwe Dick

Maschinelle Übersetzung

Maschinelle Übersetzung

- Gegeben Text in Quellsprache,
- Generiere äquivalenten Text in Zielsprache.

- Anwendungen:
  - Übersetzung von Webseiten.
  - Ermöglicht Verstehen eines fremdsprachlichen Textes, auch wenn Übersetzung nicht genau ist.
    - Assistenz, Vorschlag für menschlichen Übersetzer.
    - Schriftverkehr von EU-Behörden wird in alle EU-Sprachen übersetzt, erforderliche Genauigkeit ist aber höher als die heutiger Übersetzungssoftware.

Statistische Übersetzung

- Problemstellung:
  - Gegeben Satz in Quellsprache Q,
  - finde den besten Satz in Zielsprache Z:

    $\arg\max_Z P(Z \mid Q)$
- Modellierung von gemeinsamer latenter Struktur S:
  - $\arg\max_Z P(Z \mid Q) = \arg\max_Z \sum_S P(Z, S \mid Q)$
  - Meistens vereinfachend statt Summe über alle latenten Strukturen: $\arg\max_{Z,S} P(Z, S \mid Q)$
- Oft: Aufteilung in (inverses) Übersetzungsmodell und Sprachmodell
  - Satz von Bayes: $\arg\max_Z P(Z \mid Q) = \arg\max_Z P(Q \mid Z) P(Z)$

Teilprobleme

- Modellierung:
  - Welche Klasse von Strukturen, welche Parametrisierung
- Trainingsdaten:
  - Zweisprachige Korpora mit Alignment
- Parameterschätzung:
  - Suchen der besten Parameterwerte gegeben die Trainingsdaten
- Dekodierung:
  - Effiziente Berechnung von $\arg\max_{Z,S} P(Z,S \mid Q)$

Modellierung

- Finite State Transducers
  - Basieren auf *synchroner* regulärer Grammatik:
    - erzeugt simultan 2 Sätze in 2 verschiedenen Sprachen
  - Definition:
    - endliche Menge von Nichtterminalen N, Startsymbol N_1, Endsymbol N_e
    - Terminalsymbole in Quellsprache $\{q^k\}$ (Eingabe) und in Zielsprache $\{z^k\}$ (Ausgabe)
    - Regeln $\{N^i \rightarrow <q^k/z^l>,N^j\}$
    - Wahrscheinlichkeiten für Regeln, $P(N^i \rightarrow <q^k/z^l>,N^j)$
  - Dekodierung: wahrscheinlichste Regelsequenz auf Quellsatz bestimmen (Viterbi); Zielsatz ableiten

Finite State Transducers

- Beispiel:
  - Regel 1: $P(S^1 \rightarrow <The/Das>, S^2) = 0{,}5$
  - Regel 2: $P(S^1 \rightarrow <A/Ein>, S^3) = 0{,}5$
  - Regel 3: $P(S^2 \rightarrow <red/rote>, S^4) = 1$
  - Regel 4: $P(S^3 \rightarrow <red/rotes>, S^4) = 1$
  - Regel 5: $P(S^4 \rightarrow <car/Auto>, S^e) = 1$
- Dekodierung von „The red car“:
  - Finden der besten Regelsequenz mit Viterbi-Algorithmus: R1, R3, R5
  - Ausführen der „Zielsprachenhälfte“ der Regelsequenz: „Das rote Auto“

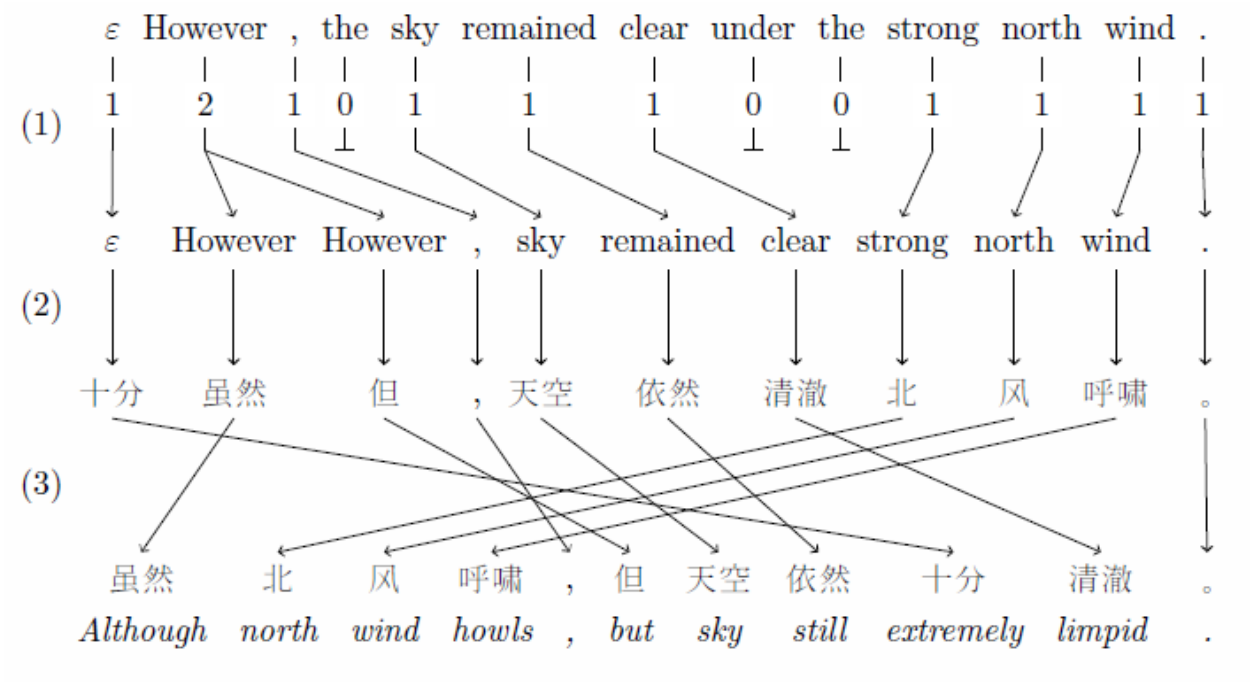
Finite State Transducers

- Jedes Wort wird *nicht* für sich genommen übersetzt, *aber*:
  - Reihenfolge der Wörter bleibt gleich.
- Verschiedene Möglichkeiten, Reihenfolge zu ändern:
  - nachträgliches Umsortieren anhand Sprachmodell der Zielsprache
  - Einlesen der Eingabe in beliebiger Reihenfolge erlauben ($\rightarrow$ folgt auf den nächsten Folien)

Finite State Transducers mit variabler Reihenfolge

- Keine 1-zu-1-Beziehung zwischen den Wörtern
  - Beziehung und Reihenfolge muss explizit modelliert werden
- Satz von Bayes: $\arg\max_Z P(Z \mid Q) = \arg\max_Z P(Q \mid Z) P(Z)$
- 3-stufiges generatives Modell für die Erzeugung des Satzes Q (in Sprache L^Q) aus dem Satz Z:
  1. Fertilität: für jedes Wort in Z wählen, wie viele daraus erzeugt werden (kopieren)
  2. Wort-Übersetzung: Für jede Wortkopie in Z wählen, zu welchem Wort in L^Q es übersetzt wird
  3. Position: Für jedes übersetzte Wort Position im Q wählen

Finite State Transducers mit variabler Reihenfolge



Finite State Transducers mit variabler Reihenfolge

- Dekodierung: Übersetze Sequenz $z_1, \ldots, z_n$ von Wörtern in Sequenz $q_1, \ldots, q_m$
- Vereinfachte Problemstellung:
  - Fertilität =1 für jedes Wort:
    - Beide Sätze gleich lang: $m = n$.
    - Abbildung $\pi : i \mapsto j$ bildet jeden Index i eines Wortes aus Z auf einen Index j aus Q ab.
      - Wahrscheinlichkeit abhängig vom Vorgängerwort: $P\left(\pi\left(i\right) \mid \pi\left(i-1\right)\right)$
      - Zusätzlich injektiv: $i_1 \neq i_2 \Rightarrow \pi\left(i_1\right) \neq \pi\left(i_2\right)$
    - Übersetzung eines Wortes mit Wahrscheinlichkeit $P\left(q_{\pi(i)} \mid z_i\right)$
    - Jeder Satz gleichverteilt in Zielsprache.

Finite State Transducers mit variabler Reihenfolge

- Vereinfachte Problemstellung:
  - gesucht: $\max_{\substack{z_1,\ldots,z_m \\ \pi(1),\ldots,\pi(m)}} \prod_{i=1}^{m} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$

$$\max_{\substack{z_1,\ldots,z_m \\ \pi(1),\ldots,\pi(m)}} \prod_{i=1}^{m} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$$

Max-Operator aufteilen

$$= \max_{\substack{z_m \\ \pi(m)}} P\left(q_{\pi(m)} \mid z_m\right) \max_{\substack{z_1,\ldots,z_{m-1} \\ \pi(1),\ldots,\pi(m-1)}} P\left(\pi(m) \mid \pi(m-1)\right) \cdot \prod_{i=1}^{m-1} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$$

Rekursive Definition möglich?

Finite State Transducers mit variabler Reihenfolge

- Vereinfachte Problemstellung:
  - gesucht: $\max_{\substack{z_1,\ldots,z_m \\ \pi(1),\ldots,\pi(m)}} \prod_{i=1}^{m} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$

$$\max_{\substack{z_1,\ldots,z_{m-1} \\ \pi(1),\ldots,\pi(m-1)}} P\left(\pi(m) \mid \pi(m-1)\right) \cdot \prod_{i=1}^{m-1} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$$

$$= \max_{\substack{z_{m-1} \\ \pi(m-1)}} P\left(\pi(m) \mid \pi(m-1)\right) P\left(q_{\pi(m-1)} \mid z_{m-1}\right) \cdot$$

$$\max_{\substack{z_1,\ldots,z_{m-2} \\ \pi(1),\ldots,\pi(m-2)}} P\left(\pi(m-1) \mid \pi(m-2)\right) \prod_{i=1}^{m-2} P\left(\pi(i) \mid \pi(i-1)\right) P\left(q_{\pi(i)} \mid z_i\right)$$

Rekursive Definition möglich! Effizient berechenbar? Nein, ignoriert Nebenbedingung!

Finite State Transducers mit variabler Reihenfolge

- Nebenbedingungen:
  - Nebenbedingung $i_1 \neq i_2 \Rightarrow \pi(i_1) \neq \pi(i_2)$ koppelt Wahrscheinlichkeiten $P(\pi(i) \mid \pi(i-1))$
  - Für Maximierung über $\pi(i)$ sind Zwischenwerte aller Kombinationen für $\pi(1), \ldots, \pi(i-1)$ notwendig
  - Ansatz: Beam-Search für suboptimale Lösung
    - Setze $C = \{(1), \ldots, (m)\}$
    - Für $i = 2, \ldots, m$
      - Für alle $\pi \in C$: bilde $D(\pi) = \{(\pi, k) \mid k \notin \pi\}$
      - Berechne $D = \bigcup_{\pi \in C} D(\pi)$ und setze $C = D$
    - Gebe $\pi \in C$ mit höchster Wahrscheinlichkeit aus

Beam-Search: Merke nur die besten N_1 Permutationen in D

Beam-Search: Merke nur die besten N_2 Permutationen in C

Teilprobleme

- Modellierung:
  - z.B. Finite State Transducers
- Trainingsdaten:
  - Zweisprachige Korpora mit Alignment
- Parameterschätzung:
  - Suchen der besten Parameterwerte gegeben die Trainingsdaten
- Dekodierung
  - Effiziente Berechnung von $\arg\max_{Z,S} P(Z,S \mid Q)$

Trainieren des Übersetzungsmodells

- Trainingsdaten: Parallele Korpora, z.B.
  - Parlamentsreden aus Kanada, EU, Hong Kong.
  - Multilinguale Zeitungsmeldungen
- Paralleles Korpus muss „aligniert“ sein,
  - Alignierung auf Wortebene: Zuordnung von Textpositionen, soweit die Terme einander entsprechen.
    - automatisch z.B. durch Inferenz der wahrscheinlichsten Positionen und Fertilität.
  - Alignierung auf Satzebene: Zuordnung von ganzen Sätzen, welche einander entsprechen.
    - Kann automatisch ausgeführt werden, um mehr Trainingsdaten zu generieren.

Alignierung

- Alignierung auf Satzebene:
  - 1:1 – Satz im Quelldokument entspricht einem Satz im Zieldokument.
  - n:m – n Sätze im Quelldokument entsprechen m Sätzen im Zieldokument.
  - n, m im Bereich 0...3.
- Ansätze zum Schätzen der Wahrscheinlichkeit einer Alignierung auf Satzebene:
  - Längenbasiert
  - Buchstaben-N-Gramm-basiert
  - Lexikalisch

Alignierung

- Längenbasiert
  - Finde Alignierung $((Q_1, Z_1), ..., (Q_n, Z_n))$, wobei Q_i und Z_i 0, 1 oder 2 Sätze enthalten, so dass Q_i und Z_i möglichst gleich groß sind.
  - Beste Alignierung wird mit dynamischer Programmierung bestimmt.
- Buchstaben-N-Gramme
  - Finde Alignierung, bei dem die Q_1, Z_1 möglichst viele gleiche Buchstaben-N-Gramme enthalten.
- Lexikalisch
  - Basierend auf Wort-zu-Wort-Übersetzung; Z_i soll möglichst viele wörtliche Übersetzungen aus Q_i enthalten.

Teilprobleme

- Modellierung:
  - z.B. Finite State Transducers
- Trainingsdaten:
  - Alignierung auf Satzebene z.B. längenbasiert
  - Alignierung auf Wortebene z.B. durch wahrscheinlichste Zuordnung
- Parameterschätzung:
  - Suchen der besten Parameterwerte gegeben die Trainingsdaten
- Dekodierung:
  - Effiziente Berechnung von $\arg\max_{Z,S} P(Z,S \mid Q)$

Lernen des Übersetzungsmodells

- Trainingsdaten mit Alignierung auf Wortebene:
  - Wahrscheinlichkeiten für Fertilität, Position, Wortübersetzung durch Abzählen
- Trainingsdaten mit Alignierung auf Satzebene:
  - Wahrscheinlichkeiten des Modells müssen aus parallelem Korpus mit alignierten Sätzen gelernt werden.
  - Problem:
    - Schätzen der Parameter des generativen Modells erfordert Alignierung auf Wortebene.
    - Für Alignierung auf Wortebene braucht man Parameter des generativen Modells
  - EM-Algorithmus.

Lernen des Übersetzungsmodells

- EM-Algorithmus.
- Starte mit heuristisch initialisiertem generativen Modell
- Wiederhole bis Konvergenz:
  - Finde wahrscheinlichste Übersetzung für jeden Satz.
    - Liefert Informationen zu Position, Fertilität und Wortübersetzung.
  - Schätze Parameter für Fertilität, Wortübersetzungen und Positionen (alles Multionomialverteilungen) durch Abzählen

Teilprobleme

- Modellierung:
  - z.B. Finite State Transducers
- Trainingsdaten:
  - Alignierung auf Satzebene z.B. längenbasiert
  - Alignierung auf Wortebene z.B. durch wahrscheinlichste Zuordnung
- Parameterschätzung:
  - EM-Algorithmus.
- Dekodierung:
  - Effiziente Berechnung von $\arg\max_{Z,S} P(Z,S \mid Q)$

Wortsalat

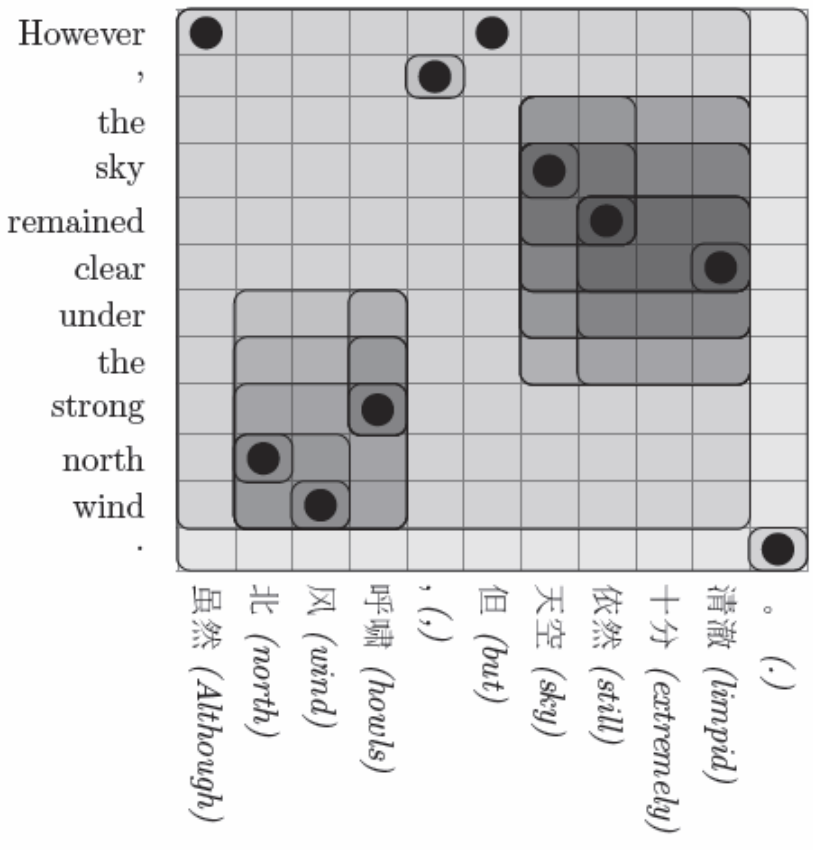
- Bisher: wortbasierte Modelle
- Problem:
  - exponentiell viele Möglichkeiten, Wörter anzuordnen
  - Modellierung der Positionsverteilung für die übersetzten Wörter große Schwachstelle
  - $\rightarrow$ Wortsalat
- Besser: direkt größere Einheiten bearbeiten
  - Phrasenbasierte Modelle
  - Phrasen sind Paare von Wortsequenzen in Quell- und Zielsatz, die nur innerhalb aligniert sind

Phrasen-basierte Modelle

- Konstruktion von Phrasen formell:
- Gegeben ein Satzpaar $\langle Q, Z \rangle$ mit Alignierung ~, so dass gilt $q_{k_1} \sim z_{k_2}$ gdw. das k_1-te Wort im Quellsatz mit dem k_2-ten Wort im Zielsatz aligniert ist
- Dann ist $\langle q_{i_1} \ldots q_{j_1}, z_{i_2} \ldots z_{j_2} \rangle$ ein Phrasenpaar, wenn gilt
  1. $q_{k_1} \sim z_{k_2}$ für irgendein $k_1 \in [i_1, j_1]$ und $k_2 \in [i_2, j_2]$
  2. $q_{k_1} \not\sim z_{k_2}$ für alle $k_1 \notin [i_1, j_1]$ und $k_2 \in [i_2, j_2]$
  3. $q_{k_1} \not\sim z_{k_2}$ für alle $k_1 \in [i_1, j_1]$ und $k_2 \notin [i_2, j_2]$

Phrasen-basierte Modelle

- Punkte: Alignierung auf Wortebene
- Rechtecke: Phrasen



Phrasen-basierte Modelle

- Statt Wahrscheinlichkeiten für Wort-zu-Wort-Übersetzung Phrasen-zu-Phrasen-Übersetzungen lernen
  - Funktioniert besser als wortbasierte Modelle
  - Wörter sind Spezialfälle von Phrasen, daher immer Fallback möglich bei unbekannten Phrasen
- Weiterhin: Reihenfolge der Phrasen immer noch schwierig („Phrasensalat“)
- Lösung: Sortierung direkt in die latente Struktur übernehmen ($\rightarrow$ Synchrone PCFGs)

Synchrone PCFGs

- Simultane Erzeugung von 2 Sätzen aus 2 kontextfreien Sprachen
- Definition:
  - endliche Menge von Nichtterminalen N, Startsymbol N_1
  - Terminalsymbole in Quellsprache $\{q_k\}$ und in Zielsprache $\{z_k\}$
  - Regeln $\{N_i \rightarrow \langle \alpha, \beta, \sim \rangle\}$, wobei α Folge aus Quellterminalen und -nichtterminalen ist, β eine Folge aus Zielterminalen und -nichtterminalen, $\sim$ ein Alignment zwischen den Nichtterminalen in α und β
  - Wahrscheinlichkeiten für Regeln, $P\left(N_i \rightarrow \langle \alpha, \beta, \sim \rangle\right)$

Synchrone PCFGs

- Beispiel (Alignierung ~ durch Indizes verdeutlicht)

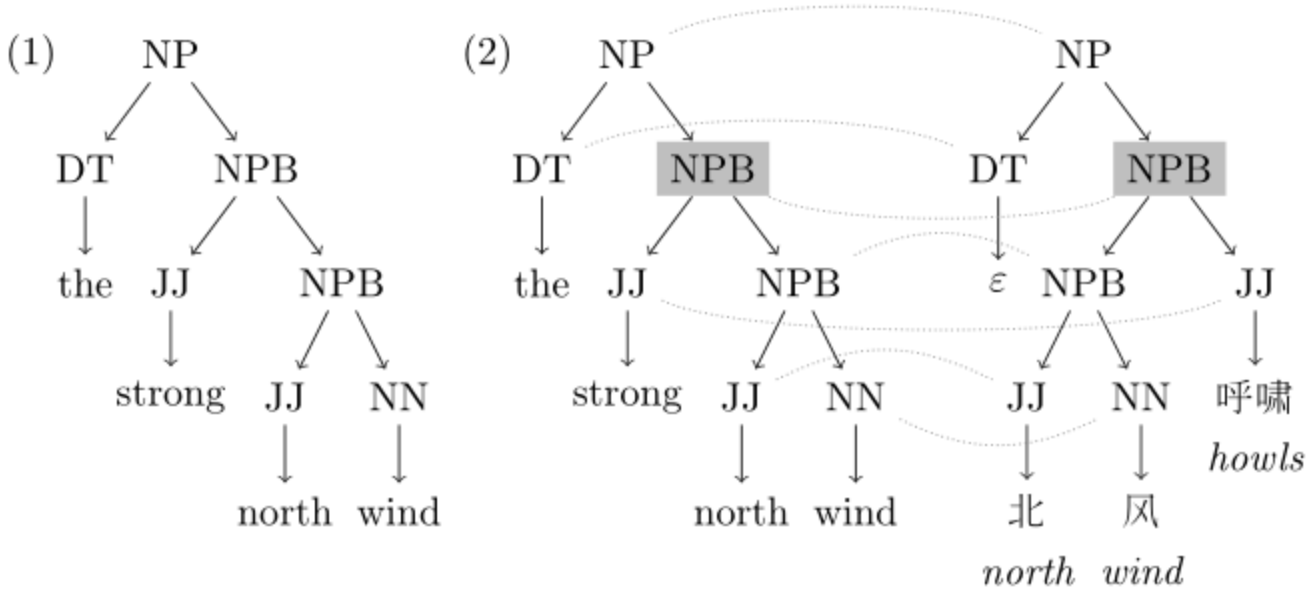
$$
\begin{aligned}
\text{NP} &\longrightarrow \text{DT}_{\boxed{1}}\text{NPB}_{\boxed{2}} \;/\; \text{DT}_{\boxed{1}}\text{NPB}_{\boxed{2}} \\
\text{NPB} &\longrightarrow \text{JJ}_{\boxed{1}}\text{NN}_{\boxed{2}} \;/\; \text{JJ}_{\boxed{1}}\text{NN}_{\boxed{2}} \\
\text{NPB} &\longrightarrow \text{JJ}_{\boxed{2}}\text{NPB}_{\boxed{1}} \;/\; \text{NPB}_{\boxed{1}}\text{JJ}_{\boxed{2}} \\
\text{DT} &\longrightarrow \text{the} \;/\; \varepsilon \\
\text{JJ} &\longrightarrow \text{strong} \;/\; \text{呼啸} \\
\text{JJ} &\longrightarrow \text{north} \;/\; \text{北} \\
\text{NN} &\longrightarrow \text{wind} \;/\; \text{风}
\end{aligned}
$$

Synchrone PCFGs: Parameterschätzung

- Regeln müssen gegeben sein
- Alignierung auf Wortebene muss bekannt sein
  - z.B. mit einfacherem Übersetzungsmodell berechnet
- EM-Algorithmus: abwechselnd
  - Für jede Regel: die erwartete Anzahl der Verwendungen für das Erzeugen der Trainingsdaten berechnen (mit Inside-Outside-Algorithmus (Forward-Backward für PCFGs))
  - Parameter schätzen durch Abzählen

Synchrone PCFGs: Dekodierung

- Gegeben Satz in Quellsprache:
  - wahrscheinlichsten Parse-Baum finden, unter Berücksichtigung nur des Quellsprachenteils der Regeln
  - Zielsprachenteil ableiten unter Berücksichtigung der Alignierung



Synchrone PCFGs

- Vorteil: elegante Handhabung von Reihenfolgenänderung
- Nachteil: Regeln müssen bekannt sein
  - Definition von synchronen Regeln schwierig
  - besonders für Sprachen mit stark unterschiedlicher Grammatik
- Erweiterung: synchrone PCFGs auf Phrasenbasis mit automatisch generierten Regeln

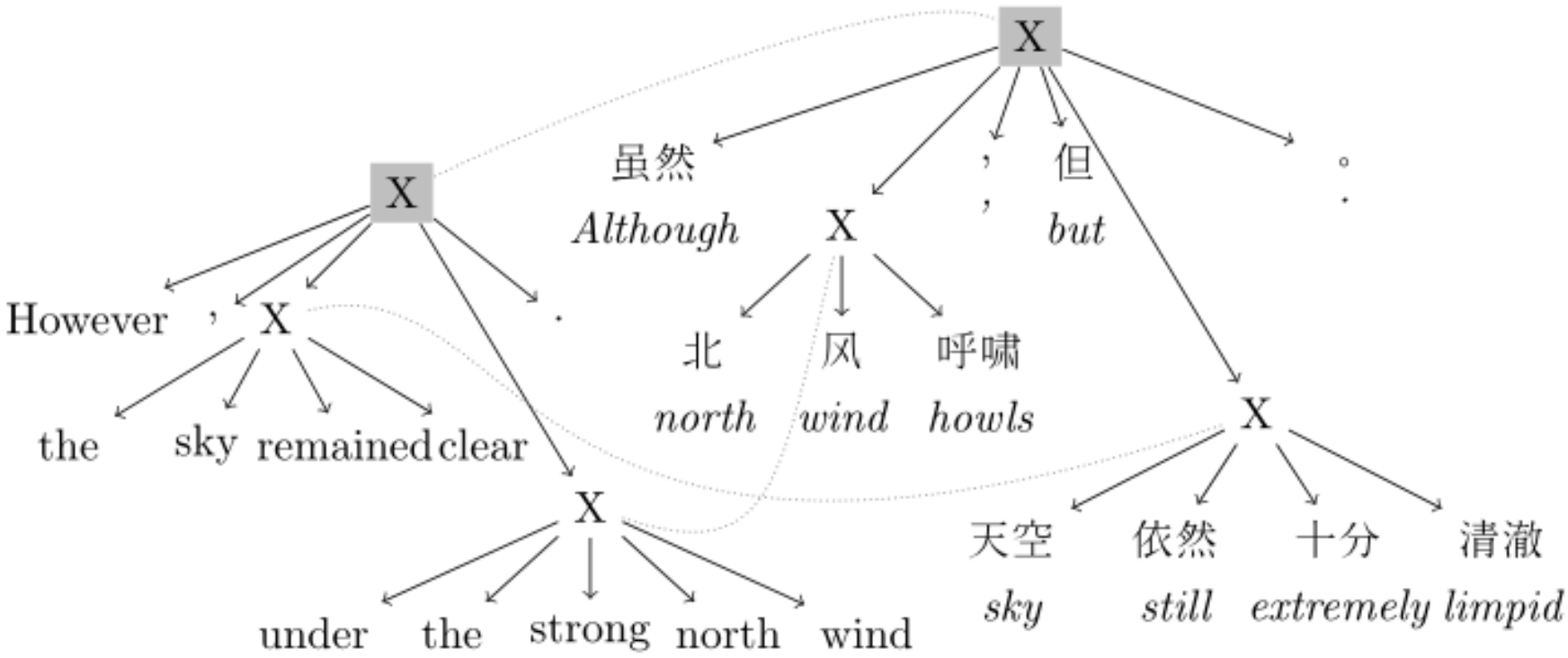
Hierarchische Phrasenübersetzung

- Ein einziges generisches Nichtterminalsymbol
- Regeln werden aus Phrasenpaaren automatisch extrahiert

- Beispiel:

$$X \longrightarrow \text{However , } X_{\boxed{1}} X_{\boxed{2}} \text{ . } / \text{虽然 } X_{\boxed{2}} \text{ , 但 } X_{\boxed{1}} \text{ 。}$$
$$X \longrightarrow \text{under the strong north wind } / \text{北 风 呼啸}$$
$$X \longrightarrow \text{the sky remained clear } / \text{天空 依然 十分 清澈}$$

Hierarchische Phrasenübersetzung

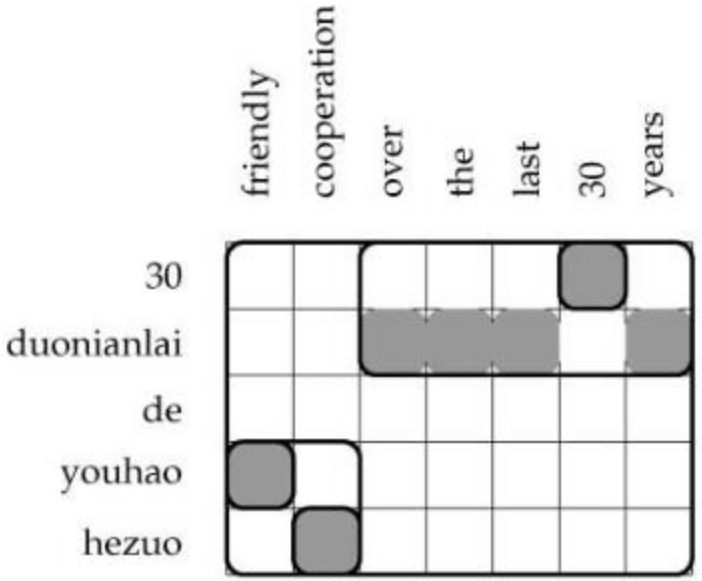


Hierarchische Phrasenübersetzung

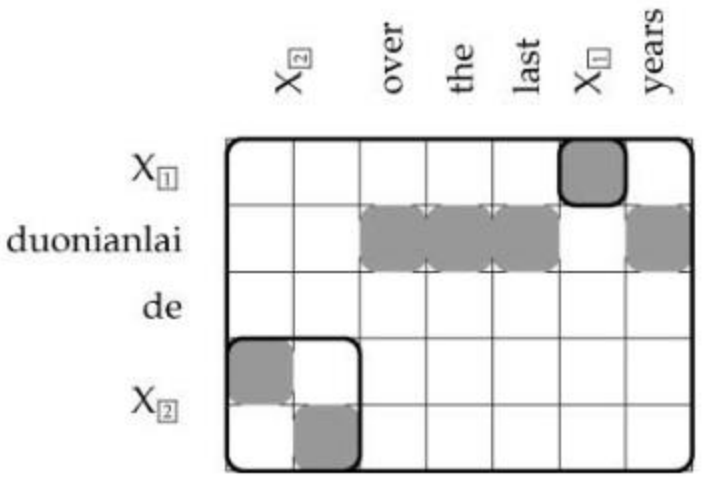
- Algorithmus zum Generieren der Regeln (Alignierung wieder durch Kästchen dargestellt):
  1. Für alle Phrasenpaare $\langle q_{i_1} \ldots q_{j_1}, z_{i_2} \ldots z_{j_2} \rangle$:
     - füge die Regel $X \rightarrow \langle q_{i_1} \ldots q_{j_1}, z_{i_2} \ldots z_{j_2} \rangle$ hinzu
  2. Für alle Regeln $X \rightarrow \langle \alpha, \beta \rangle$:
     - Wenn $\langle q_{i_1} \ldots q_{j_1}, z_{i_2} \ldots z_{j_2} \rangle$ ein Phrasenpaar ist, so dass gilt $\alpha = \alpha_1 q_{i_1} \ldots q_{j_1} \alpha_2, \beta = \beta_1 z_{i_2} \ldots z_{j_2} \beta_2$, dann füge die Regel $X \rightarrow \langle \alpha_1 X_{\boxed{k}} \alpha_2, \beta_1 X_{\boxed{k}} \beta_2 \rangle$ für ein neues k hinzu
  3. Wiederhole Schritt 2, bis keine neuen Regeln mehr hinzukommen
- In der Praxis: hinterher mit Hilfe von Heuristiken Regeln filtern (sonst zu viele)

Regelerzeugung: Beispiel

- Ausgangspunkt:
  Alignment und Phrasen



- Nach 2
  Ersetzungsschritten:



Teilprobleme

- Modellierung:
  - z.B. Finite State Transducers
- Trainingsdaten:
  - Alignierung auf Satzebene z.B. längenbasiert
  - Alignierung auf Wortebene z.B. durch wahrscheinlichste Zuordnung
- Parameterschätzung:
  - EM-Algorithmus.
- Dekodierung:
  - z.B. Phrasenbasierte oder hierarchische Modelle

Methoden: Überblick und Ausblick

- Phrasenbasierte Modelle fast immer überlegen
- Sowohl Übersetzung mit Finite State Transducern als auch hierarchische Übersetzung wird weiterentwickelt
- Schwierigkeit: nicht-lokale Kontextinformation
  - prinzipiell Einbeziehung von beliebigen Features möglich
  - aber: erschwert Dekodierung
  - Tradeoff zwischen Ausdrucksstärke der Modelle und effizienter Dekodierbarkeit

Evaluierung von Übersetzern

- Evaluierung
  - Menschliche Übersetzer sehen sich die maschinellen Übersetzungen an und vergeben Punkte.
  - N-Gramm-Co-Occurrence-Statistik: Wie viele N-Gramme einer Referenzübersetzung tauchen in der maschinellen Übersetzung auf?

- Jährlicher gemeinsamer Wettbewerb:
  - NIST (National Institute for Standards and Technology): MetricsMATR
  - Workshop on Statistical Machine Translation (WMT)
  - Evaluierung von Übersetzungssystemen und Metriken

Fragen?