

# **Softwareprojektpraktikum Maschinelle Übersetzung**

**Markus Freitag, Matthias Huck**  
**{huck, freitag}@i6.informatik.rwth-aachen.de**

**Vorbesprechung 3. Aufgabe 12. Mai 2011**

**Human Language Technology and Pattern Recognition  
Lehrstuhl für Informatik 6  
Computer Science Department  
RWTH Aachen University, Germany**

# Wie lässt sich die Qualität von Übersetzungen beurteilen?

## ► Ziele:

- ▷ Vergleich der Performanz verschiedener maschineller Übersetzungssysteme
- ▷ Beurteilung des Effekts inkrementeller Änderungen während der Systementwicklung
- ▷ Optimierung freier Parameter

## ► Menschliche Bewertung:

- ▷ zeitaufwändig
- ▷ teuer
- ▷ Beurteilungen verschiedener Evaluatoren stimmen oft nicht überein
- ▷ Entscheidungen ein und desselben Evaluators sind oft nicht konsistent

# Adequacy und Fluency

## ► Mögliche Kriterien zur Beurteilung von Übersetzungen:

- ▷ **Adequacy:** Gibt die Übersetzung die Bedeutung des Quellsatzes wieder?  
Sind keine Inhalte ausgelassen oder hinzugefügt worden?
- ▷ **Fluency:** Ist die Übersetzung in der Zielsprache ein korrekt formulierter und flüssig lesbarer Satz?

## ► Auf einer Skala von 1 bis 5:

The dog is barking.		
	Adequacy	Fluency
Bellen das Hund.	5	2
Das Wetter ist schön.	1	5
Der Hund bellt.	5	5

# Automatische Evaluationsmetriken

- ▶ Automatisierte Bewertung von Übersetzungen durch Vergleich mit einer oder mehreren von Menschen erstellten Referenzübersetzungen
- ▶ Gute automatische Metriken sollten eine hohe Korrelation mit menschlicher Bewertung aufweisen
- ▶ Verbreitete Metriken:
  - ▷ **F-Measure**
  - ▷ **WER** (**W**ord **E**rror **R**ate)
  - ▷ **PER** (**P**osition-independent **E**rror **R**ate)
  - ▷ **BLEU** (**B**ilingual **E**valuation **U**nderstudy)
  - ▷ **METEOR** (**M**etric for **E**valuation of **T**ranslation with **E**xplicit **O**rdering)
  - ▷ **NIST-Score**
  - ▷ ...

## Beispiel: F-Measure

$$\text{Precision} = \frac{\text{Übereinstimmungen}}{\text{Hypothesenlänge}}$$

$$\text{Recall} = \frac{\text{Übereinstimmungen}}{\text{Referenzlänge}}$$

$$\begin{aligned}\text{F-Measure} &= \frac{\text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})/2} \\ &= \frac{\text{Übereinstimmungen}}{(\text{Hypothesenlänge} + \text{Referenzlänge})/2}\end{aligned}$$

- ▶ in diesem Fall heißt Übereinstimmung, wieviele Wörter in der Hypothese auch in der Referenz gefunden werden können

# Aufgabe 3

- ▶ **Automatische Evaluierung der Ausgabe des maschinellen Übersetzungsprogramms aus Aufgabe 2**
- ▶ **Einlesen einer Referenzübersetzung und einer zu bewertenden Übersetzung**
- ▶ **Levenshtein-Distanz mittels dynamischer Programmierung**
- ▶ **Berechnung von WER und PER**
- ▶ **Berechnung von BLEU**

# Levenshtein-Distanz

- ▶ Minimale Anzahl an Einfügungs-, Auslassungs- und Ersetzungsoperationen, die nötig sind, um die Hypothese so zu editieren, dass sie mit der Referenz übereinstimmt

Hypothese:	B	A	N	A	N	E
Referenz:	A	N	A	N	A	S

---

Hypothese:	B	A	N	A	N	E	
Referenz:	A	N	A	N	A	S	
Operationen:	d	m	m	m	m	s	i

- ▶ insertion (*i*), deletion (*d*), substitution (*s*), match (*m*)

# Berechnung der Levenshtein-Distanz

## ► Dynamische Programmierung

		B	A	N	A	N	E
	0	1	2	3	4	5	6
A	1	1	1	2	3	4	5
N	2	2	2	1	2	3	4
A	3	3	2	2	1	2	3
N	4	4	3	2	2	1	2
A	5	5	4	3	2	2	2
S	6	6	5	4	3	3	3

## ► Referenz vertikal, Hypothese horizontal

- ▷ Match: Nimm Kosten von diagonal links oben
- ▷ Substitution: Nimm Kosten von diagonal links oben, addiere 1
- ▷ Insertion: Nimm Kosten von links, addiere 1
- ▷ Deletion: Nimm Kosten von oben, addiere 1

## ► Eintragung in Tabellenzelle ist Minimum aus Match/Substitution, Insertion und Deletion



# WER und PER

- WER: Dividiere durch Referenzlänge

**Levenshtein-Distanz** = Einfügungen + Auslassungen + Ersetzungen

$$\text{WER} = \frac{\text{Levenshtein-Distanz}}{\text{Referenzlänge}}$$

- PER: Zähle Anzahl der Übereinstimmungen unabhängig von der Wortreihenfolge

$$\text{PER} = 1 - \frac{\text{Übereinstimmungen} - \max(0, \text{Hypothesenlänge} - \text{Referenzlänge})}{\text{Referenzlänge}}$$

# BLEU

- ▶ modified n-gram Precision  $p_n$  einer Hypothese  $C$ :

$$p_n = \frac{\sum_{\mathbf{n}\text{-gram} \in C} \text{Count}_{\text{match}}(\mathbf{n}\text{-gram})}{\sum_{\mathbf{n}\text{-gram}' \in C} \text{Count}(\mathbf{n}\text{-gram}')} \quad (1)$$

- ▶ Brevity Penalty  $BP$  ( $c$  Länge der Hypothese,  $r$  Länge der Referenz):

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \quad (2)$$

- ▶  $BLEU$  (meist mit  $N = 4$ )

$$BLEU = BP \cdot \exp \left( \sum_{n=1}^N \frac{1}{N} \log p_n \right) \quad (3)$$

# Modified n-gram Precision

- ▶ **Beispiel: modified n-gram Precision  $p_n$  einer Hypothese  $C$** 
  - ▷ Hypothese: the the the the the the the
  - ▷ Reference: the cat is on the mat
- ▶ **modified 1-gram Precision:**
  - ▷ the kommt in der Referenz nur zweimal vor
  - ▷  $p_n = 2/7$
- ▶ **modified 2-gram Precision:**
  - ▷ the the kommt in der Referenz niemals vor
  - ▷  $p_n = 0/7 = 0$

# Brevity Penalty

## ► Brevity Penalty:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \quad (4)$$

- Problem bei zu kurzen Hypothesen: modified n-gram Precision sehr hoch
- Bsp: Hypothese: the cat, Reference: the cat is on the mat
- modified n-gram Precision =  $2/2 = 1$
- Lösung: Brevity Penalty bestraft zu kurze Hypothesen

# Einzelaufgaben

- ▶ Jeder Praktikumsteilnehmer ist für die Verwendung von einem Tool zuständig
- ▶ Kurze (eine DIN-A4 Seite) Dokumentation für Teamteilnehmer (und uns) über deren Nutzung (mit Beispiel)
- ▶ Software-Dokumentation: *doxygen*
- ▶ Architektur: Übersicht der Module mit *dot*
- ▶ Programmierung: Profiling mit *gprof*, Analyse und Implementationsvorschläge
- ▶ Planung: Repository, z.B. mit *git* oder *svn*, und einen Bug-Tracker, z.B. *ditz*

# Fragen?

## Viel Erfolg!

