

Sequitur G2P

Joint-sequence Anwendung für Graphem-zu-Phonem Konversion

Martin Schwietzke

Institut für Linguistik

1. Juli 2008



Agenda

- 1 Intro
- 2 G2P Techniken
- 3 Sequitur G2P
- 4 Demo
- 5 Quellen

Graphem zu Phonem

- keine 1 zu 1 Übersetzung von G-zu-P in natürlicher Sprache
- z.B. Qualm = [kvalm], am = [am] vs. Damm = [dam] etc.
- G2P Konversion zielt darauf ab Aussprache für ein Wort in geschriebener Form zu finden
- wichtig für TTS, Sprach-Synthese, Sprach-Erkennung

kurze Geschichte

- erste Ansätze seit 1984 (Lucassen und Mercer) mit Wahrscheinlichkeitstheorie
- Ansätze seit 1987 (Senjowski und Rosenberg) mit diversen machine learning Techniken
- später mit neuronalen Netzwerken
- seit den 90ern verstärktes Interesse da TTS, SR, SS immer wichtiger
- bis heute noch keine perfekten Ergebnisse

Automatische G2P Konversion

- zuerst für TTS gedacht
- 1.: Eingabetext tokenisieren
- 2.: Eingabetext normalisieren (Abkürzungen, Numerale, Akronyme etc. auflösen)
- 3.: Phonemsequenzen erstellen
- 4.: Phonemsequenzen steuern Speech-Synthesizer

dictionary look-up

- Wörterbuch mit entsprechenden Phonemsequenzen für jeden Eintrag
- effektiv da look-up recht schnell
- viele Nachteile:
 - Phonemsequenzen müssen per Hand erstellt werden (teuer!!!)
 - bei großen Datenmengen wird Speicherung schwierig (z.B. mobiler Einsatz)
 - begrenzte Wortmenge, neue Wörter werden nicht erkannt (ständige Updates)

rule-based

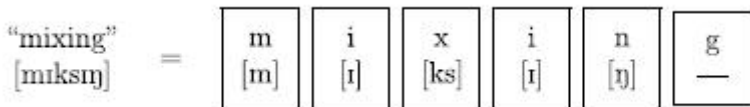
- als endliche Automaten implementiert
- brauchen ein Wörterbuch was als Ausnahmeliste fungiert
- Nachteile:
 - Regeln erstellen ist schwierig und benötigt viel linguistisches Wissen
 - Irregularitäten brauchen Ausnahmeregeln bzw. Ausnahmelisten
 - Verflechtung der Regeln erhöhen Komplexität in Entwicklung und Wartung
 - fehleranfällig wenn Regeln vergessen oder vertauscht

data-driven

- Idee: ausreichend Beispiele sollen Aussprache für unbekannte Wörter durch Analogien vorhersagbar machen
- es brauchen keine Regeln mehr erstellt werden sondern nur Beispiele geliefert werden
- Problem: passende Algorithmen (er)finden
- 2 Ziele:
 - Lexikon-Kompression: Speicherplatz (und Rechenaufwand) durch Fehlerreduzierung bei bekannten Daten minimieren
 - Generalisierung: Erhöhung der (begrenzten) Abdeckung eines Wörterbuches durch Fehlerreduzierung bei unbekanntem Daten

lokale Klassifikation

- viele G2P Methoden benötigen vorheriges Alignment (vorgeschalteter Schritt)
- 1-to-n Alignment (Anzahl der Phoneme = 0, 1 , größer 1)



- Alignment
 - per Hand
 - durch Suche mit vordefinierten Constraints
 - iterative Abschätzung von Alignment Wahrscheinlichkeiten

lokale Klassifikation

- typisch: Input wird von links nach rechts abgearbeitet (aber auch rückwärts möglich)
- für jedes Token wird ein Phonem aus einer Menge von zugelassenen Phonemen ausgewählt
- kontextbasierte Auswahl von Output-Phonem(en)

lokale Klassifikation

- lokale Klassifikation weil Entscheidung über ein Phonem eins nach dem anderen
- üblicherweise mit neuronalen Netzen oder Entscheidungsbäumen
- viele verschiedene Ausführungen (z.B. letter code-book, dynamische Expansion etc.)

Pronunciation by Analogy

- alle data-driven G2P Konversionstechniken können eigtl damit bezeichnet werden
- aber gemeint sind meistens Ansätze, die das “nächster Nachbar“-Prinzip verfolgen
- d.h. Trainingslexikon wird nach Worten oder Wortteilen abgesucht, die dem Input ähneln (Analogien)

Pronunciation by Analogy

- da meist komplette Wörter gesucht werden anderer Ansatz als lokale Klassifikation
- auf statistischen Modellen beruhend
- viele verschiedene graphentheoretische Ausführungen (z.B. Generierung von "Ausprache-Gittern", gewichtete Graphen etc.)

probabilistische Ansätze

- Lucassen und Mercer (1984):
 - 1.: 1-to-n Alignment von Trainingsdaten mit kontext-unabhängigem Modell
 - 2.: Vorhersage des nächsten Phonems anhand eines "Graphem-Phonem-Gitters"
 - 3.: Erzeugung von binären Feature Funktionen
 - 4.: Erzeugung eines Regressionsbaumes mit Verteilungswahrscheinlichkeiten der Phoneme

probabilistische Ansätze

- seither stetige Verbesserungen dieses Ansatzes
- z.B. Phonem-Trigramme, Einbindung morphologischer Parse-Bäume bis hin zu 7-grammen mit phonotaktischen Regeln
- dies hat zu sogenannten "joint-sequence"-Modellen geführt

joint-sequence

- Grundidee: Relation von Input- und Output-Sequenzen wird aus einer allgemeinen Sequenz von joint-units generiert
- joint-units beinhalten Input- und Output-Symbole
- einfachster Fall: jede unit hat 0 oder 1 Input-Symbole und somit 0 oder 1 Output-Symbole

joint-sequence

- entspricht der Definition eines FST
- wenn units mehrere Input- und Output-Symbole haben spricht man von “co-sequence“ oder “joint-multigram“
- kann theoretisch auf jedes monotone Übersetzungsproblem angewendet werden ohne linguistisches Wissen zu haben
- Anwendung auf G2P: Sequitur G2P

Sequitur G2P

- von Maximilian Bisani und Hermann Ney an RWTH Uni in Aachen entwickelt
- Implementation als Open Source verfügbar
- joint-units = graphones (in diesem Zusammenhang aber auch Begriffe wie grapheme-to-phoneme correspondences (GPC) oder graphonemes gebräuchlich)

Sequitur G2P

- singular graphone: 1:1 Relation von Graphem und Phonem

“mixing”
[mɪksɪŋ] =

m	i	x	ing
[m]	[ɪ]	[ks]	[ɪŋ]

- = co-segmentation und m-to-n Alignment
- Graphone werden aus Trainingsdaten abgeleitet oder von Hand erstellt

Sequitur G2P

- m-to-n Alignment auch so möglich:

“mixing”
[mɪksɪŋ] =

m	i	x	—	i	n	g
[m]	[ɪ]	[k]	[s]	[ɪ]	—	[ŋ]

- beide Varianten sind gleichwertig
- Variante 2 wird 01-to-01 (oder FST)-Alignment genannt

Sequitur G2P Implementation (Einblick)

- Kernstück der Implementation ist der Expectation Maximization (EM) Algorithmus

```
for  $M = 1$  to  $M_{\max}$ :  
  initialize  $M$ -gram model with  $(M-1)$ -gram model  
   $p_M(q|h) = p_{M-1}(q|\bar{h})$   
  initialize the additional discount parameter  
   $d_M = d_{M-1}$   
  repeat until  $\mathcal{L}(\mathcal{O}_h)$  stops increasing:  
    compute evidence according to (11)  
    if  $\mathcal{L}(\mathcal{O}_h)$  did not increase:  
      adjust discount parameters  $d_1, \dots, d_{M-1}$  by direction set method  
       $\mathbf{d} = \operatorname{argmax}_{\mathbf{d}'} \mathcal{L}(\mathcal{O}_h; \mathbf{d}')$   
    update model according to (15) and (18)
```

Sequitur G2P vs. andere

Data set	Author	PER [%]	WER [%]
Beep	This work	3.38 ± 0.03	20.08 ± 0.15
Celex	= Bisani and Ney (2002)	3.98	
	= Vozila et al. (2003)	3.68	17.13
	= Chen (2003)	2.7	
	= This work	2.50 ± 0.11	11.42 ± 0.43
OALD	Pagel et al. (1998) with POS	6.03	21.87
	Pagel et al. (1998) w/o POS		23.34
	= Chen (2003)		18.9
	= This work	3.54 ± 0.19	17.49 ± 0.78
NETtalk 15k	Andersen et al. (1996)		47.0
	Jiang et al. (1997)	8.1	34.2
	Chen (2003)		34.6
	This work	8.26 ± 0.32	33.67 ± 1.10
NETtalk 18k	Torkkola (1993)	9.2	
	Yvon (1996)		36.04
	Galescu et al. (2001)	9.00	36.07
	This work	7.83 ± 0.16	31.79 ± 0.54
NETtalk 19k	Marchand and Damper (2000)		34.5
	Chen (2003)		32.1
	This work	7.66 ± 0.31	31.00 ± 1.09
CMUdict	Galescu and Allen (2002)	7.0	28.5
	= Chen (2003)	5.9	24.7
	= This work	5.88 ± 0.18	24.53 ± 0.65
Pronlex	= Chen (2003) conditional ME	8.00	31.8
	= Chen (2003) joint ME	7.15	27.3
	= This work	6.78 ± 0.31	27.33 ± 1.04

Demo

- Helios login

Quellen

- Bisani, M., Ney, H., 2008. Joint-sequence models for grapheme-to-phoneme conversion, Elsevier B.V.
- www.sciencedirect.com
- www.elsevier.com
- <http://de.wikipedia.org>