Rheinisch Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik VI
Prof. Dr.-Ing. Hermann Ney

Seminar speech recognition and language processing im SS 2003

Wortfehlerminimierung

Torsten Palm

Juli 2003

- Frank Wessel,"Word Posterior Probabilities for Large Vocabulary Continuous Speech Recognition", Dissertation RWTH Aachen, Januar 2002
- A. Stolcke, Y. Knig, M. Weintraub, "Explicit Word Error Rate Minimization in N-best List Rescoring", Proc. Europ. Conf. on Speech Communication and Technology, September 1997, Vol. 2, pages 163-166
- L. Mangu, E. Brill, A. Stolcke, "Finding Consensus Among Words: Lattice-Based Word Error Minimization", Proc. Europ. Conf. on Speech Communication and Technology, September 1999, Vol. 1, pages 495-498
- Lidia Mangu, "Finding consensus in speech recognition", Dissertation Baltimore, Maryland, April 2000

Gliederung

- Einführung und Motivation
- Mustererkennung im Allgemeinen / Kostenfunktionen
- Entscheidungskriterien in der Spracherkennung
  - Satzfehlerminimierung
  - Wortfehlerminimierung / Levensthein-Anpassung
  - Vergleich Satzfehlerminimierung und Wortfehlerminimierung
- Explizite Algorithmen zur Wortfehlerminimierung
  - n-best-Listen-Ansatz (Levensthein-Anpassung)
  - word-lattice-Ansatz (Anpassung nach Wortpositionen)
  - "time-frame-error"-Ansatz (Anpassung nach Aussagenzeitpunkten)
- Vergleich der Ansätze
- Zusammenfassung / Ausblick

Grundlagen eines Spracherkennungssystemes

- gängige Spracherkennungssysteme : erkannter Satz mit der geringsten Wahrscheinlichkeit des Auftretens eines Satzfehlers (=mind. ein Wortfehler)
- Entscheidungskriterium nach Bayes mit der Kostenfunktion Satzfehler (Satzfehlerminimierung)
- andere Optimierungskriterien denkbar, z.B. minimale Anzahl an Wortfehlern (insertions, deletions, replacements) zwischen erkanntem und gesprochenem Satz (Wortfehlerminimierung)
- gängige Satzfehlerminimierung minimiert nicht automatisch die Wortfehler
- Konsequenz: Findung neuer Entscheidungskriterien auf Basis der Wortfehlerminimierung notwendig

Bayessches Entscheidungskriterium

- Grundlage der Mustererkennung: Minimierung der zu erwartenden Kosten die entstehen, wenn eine Beobachtung x einer Klasse c statt einer Klasse k zugeordnet wird
- $E_L(c) = \sum_{k \in C} p(k|x) \cdot L(c,k) \quad c_{opt} = \underset{c \in C}{\operatorname{argmin}}\{E_L(c)\}$

Kostenfunktion L(c,k)

- Kostenfunktion fallspezifisch nach anwendungsabhängigen Anforderungen frei wählbar
- eine Kostenfunktion muss die Zuordnung zu einer "richtigen" Klasse mit weniger Kosten belegen als die Zuordnung zu einer falschen Klasse

Beispiel 1 :

- Gleichheit bzw. Ungleichheit zwischen zwei Klassen lässt sich durch die Kroneckerfunktion $L(c,k) = 1 - \delta(c,k)$ berechnen.

$$
\begin{aligned}
c_{opt} &= \underset{c \in C}{\text{argmin}} \{ \sum_{k \in C} p(k|x) \cdot L(c,k) \} \\
&= \underset{c \in C}{\text{argmin}} \{ \sum_{k \in C} p(k|x) \cdot (1 - \delta(c,k)) \} \\
&= \underset{c \in C}{\text{argmax}} \{ p(c|x) \}
\end{aligned}
$$

Beispiel 2 :

- Zuordnung eines Authentifizierungscodes für einen Hochsicherheitstrakt zu einer der Klassen
  - a :=" Zugang nach erfolgreicher Authentifizierung"
  - b :=" Zugang nach fehlgeschlagener Authentifizierung"
  - c :=" Zugang verweigert nach erfolgreicher Authentifizierung"
  - d :=" Zugang verweigert nach fehlgeschlagener Authentifizierung"
- offensichtlich sollte $L(b, d)$ weitaus höher ausfallen (unberechtigter Zutritt) als $L(c, a)$ (neuer Versuch der Authenfizierung nötig)
- eine Kostenfunktion kann also durchaus komplexer ausfallen als der simple Vergleich auf Gleichheit

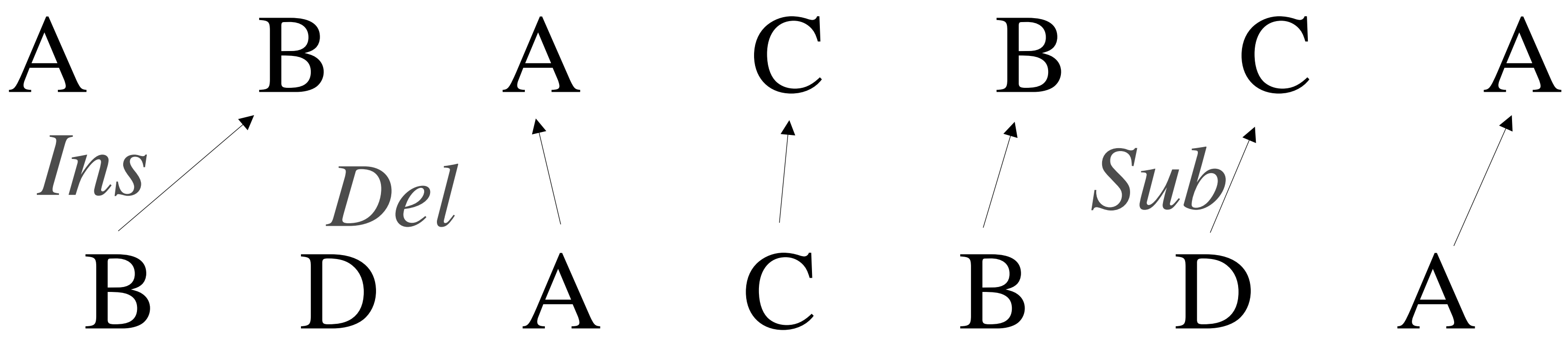
- betrachtete Klassen c bzw. k ab jetzt Sätze, daher v_1^M (Satz der Länge M) bzw. w_1^N (Satz der Länge N)
- gemachte Beobachtung x ab jetzt Folge von Beobachtungsvektoren x_1^T akustischer Beobachtungen zum Zeitpunkt t
- $\{w_1^N\}_{opt} = \underset{N,w_1^N}{\text{argmin}} \sum_{M,v_1^M} p(v_1^M|x_1^T) \cdot L(w_1^N, v_1^M)$

Bayessches Entscheidungskriterium für SEM

- Kostenfunktion $L(v_1^M, w_1^N) = 1 - \delta(v_1^M, w_1^N)$ (Test auf Satzgleichheit)
- $\delta(v_1^M, w_1^N) = 1$ wenn $M = N$ und $v_i = w_i \quad \forall \quad 1 \leq i \leq N$

$$
\begin{aligned}
\{w_1^N\}_{opt} &= \underset{N,w_1^N}{\text{argmin}}\{\sum_{M,v_1^M} (1 - \delta(v_1^M, w_1^N)) \cdot p(v_1^M | x_1^T)\} \\
&= \underset{N,w_1^N}{\text{argmin}}\{1 - p(w_1^N | x_1^T)\} \\
&= \underset{N,w_1^N}{\text{argmax}}\{p(w_1^N | x_1^T)\} = \underset{N,w_1^N}{\text{argmax}}\{p(x_1^T | w_1^N) \cdot p(w_1^N)\}
\end{aligned}
$$

Wortfehlerberechnung über Levensthein-Anpassung



- Der Algorithmus läuft beim Vergleich zweier Sätze der Länge m bzw. n in $O(mn)$
- Die Anzahl der Fehler $\mathcal{L}(w_1^N, v_1^M)$ die nach diesem Algorithmus mindestens auftreten gibt den Wortfehler an
- Der Mensch sieht Sätze die die wenigsten Wortfehler haben als am ähnlichsten an (vgl. Bewertung der Qualität eines Programmes automatischer Texterkennung)

Bayessches Entscheidungskriterium für WEM

- zur Erinnerung : $\{w_1^N\}_{opt} = \underset{N,w_1^N}{\text{argmin}}\{\sum_{M,v_1^M} p(v_1^M|x_1^t) \cdot L(w_1^N, v_1^M)\}$

  allgemeines Bayessches Entscheidungskriterium für Sätze
- jetzt $L(w_1^N, v_1^M) = \mathcal{L}(w_1^N, v_1^M)$
- $\{w_1^N\}_{opt} = \underset{N,w_1^N}{\text{argmin}}\{\sum_{M,v_1^M} p(v_1^N|x_1^T) \cdot \mathcal{L}(v_1^M, w_1^N)\}$
- bisher ist keine Möglichkeit bekannt Kriterium weiter zu vereinfachen durch direktes Berechnen des Wortfehlers (Explizites Durchführen der Levensthein-Anpassung notwendig)
- Frage : findet das neue Bayessche Entscheidungskriterium auf Wortfehlerbasis denselben Satz wie das auf Satzfehlerbasis?

Satzfehlerminimierung vs. Wortfehlerminimierung I

Wir betrachten Sätze über dem Alphabet $\{a, b\}$ der Länge 1 bzw. 2

$\mathcal{L}(v_1^M, w_1^N)$	ab	ba	a	b
ab	0	2	1	1
ba	2	0	1	1
a	1	1	0	1
b	1	1	1	0

w_1	w_2	$p(w_1 w_2 \mid x_1^T)$	$1 - p(w_1 w_2 \mid x_1^T)$	$\sum_{v_1^M} p(v_1^M \mid x_1^T) \cdot \mathcal{L}(v_1^M, w_1 w_2)$
a	b	0.35	0.65	$0 \cdot 0.35 + 1 \cdot 0.2 + 1 \cdot 0.15 + 2 \cdot 0.3 = 0.95$
a	$\square$	0.2	0.8	$1 \cdot 0.35 + 0 \cdot 0.2 + 1 \cdot 0.15 + 1 \cdot 0.3 = 0.8$
$\square$	b	0.15	0.85	$1 \cdot 0.35 + 1 \cdot 0.2 + 0 \cdot 0.15 + 1 \cdot 0.13 = 0.85$
b	a	0.3	0.7	$2 \cdot 0.35 + 1 \cdot 0.2 + 1 \cdot 0.15 + 0 \cdot 0.3 = 1.05$

- optimaler Satz nach Satzfehler : "ab" (Fehlererwartung von 0.65)
- optimaler Satz nach Wortfehler : "a" (Fehlererwartung von 0.8)
- Fazit : Ein minimieren der zu erwartenden Satzfehlerrate minimiert nicht automatisch die zu erwartende Wortfehlerrate
- Bei Anforderung einer minimierten Wortfehlerratenerwartung ist es notwendig explizit Wortfehlerminimierung zu betreiben

- bisher ist keine Kriterium zur direkten Berechnung des Wortfehlers zwischen zwei Sätzen bekannt
- Entscheidungskriterium im Allgemeinen nicht zu vereinfachen, explizites Durchführen der Levensthein-Anpassung nötig
- Levensthein-Anpassung hat Komplexität $O(n^2)$; hinzu kommt Aufsummierung über alle Hypothesen
- Wortschatz von 10.000 und Satzlänge von 5 ergibt Komplexität von $O(10.000^5 \cdot n^2)$
- Konsequenz : es müssen Ansätze gefunden werden, wodurch die Komplexität eingeschränkt wird

Idee: Vorauswahl der Hypothesen treffen und explizite Wortfehlerminimierung auf eingeschränktem Hypothesenraum (n-best-Liste = n Sätze mit maximaler Satz-posteriori-Wahrscheinlichkeit) durchführen.

- **Ausgangspunkt 1**: Hypothesenraum in Form einer n-best-Liste
- **Ausgangspunkt 2**: $\{w_1^N\}_{opt} = \underset{N, w_1^N \in n-best}{\text{argmin}} \left\{ \sum_{M, v_1^M \in n-best} p(v_1^M | x_1^T) \cdot \mathcal{L}(v_1^M, w_1^N) \right\}$
  - $P(w_1^N | x_1^T) = \dfrac{P(w_1^N) \cdot P(x_1^T | w_1^N)^{1/\lambda}}{\sum_{M, v_1^M \in n-best} P(v_1^M) \cdot P(x_1^T | v_1^M)^{1/\lambda}}$

    Neuberechnung der Satz-posteriori-Wahrscheinlichkeiten in Bezug auf Elemente der n-best-Liste, Neuerung: Abschwächung des akustischen Modells durch Exponent $1/\lambda$ da schlechter als Sprachmodell geschätzt
  - $\mathcal{L}(v_1^M, w_1^N)$ erhalten wir wie bekannt durch explizites paarweises Ausführen der Levensthein-Anpassung auf den n-best-Listen-Elementen.

performance des n-best-Listen-Ansatzes

		WER [%]	SER [%]
Switchboard	Standardansatz	52.7	84.0
	n-best-Ansatz	52.2	84.4
CallHome Spanish	Standardansatz	68.4	80.9
	n-best-Ansatz	67.8	81.2

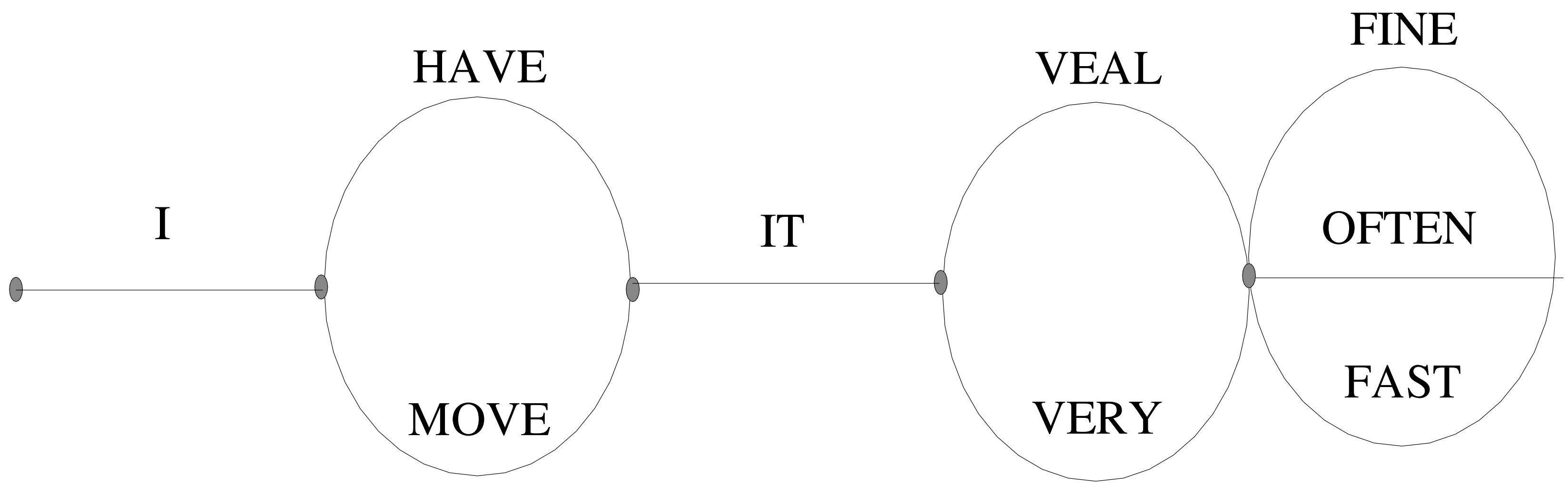
- Verbesserung der Wortfehlerrate um 0.5% bzw 0.6% absolut gesehen
- Anstieg der Satzfehlerrate klar, da nicht mehr Optimierungsziel

1:1-Zuordnung von Sätzen in Bezug auf Wortposition

Idee: Statt Hypothesenraum einzuschränken Komplexität verbessern indem statt der Levensthein-Anpassung weniger komplexe Anpassungsart angewandt wird (in der die Kosten dem Wortfehler nach Levensthein aber dennoch nahekommen))

- Position i in einem Satz v_1^N wird der Position i in einem Satz w_1^N zugeteilt, ggfls. substitution-Fehler (lineare Laufzeit)
- keine deletions oder insertions mehr, die das dynamische Durchführen der Levensthein-Anpassung notwendig machen
- kürzere Sätze werden mit dem leeren Wort an entsprechender Position aufgefüllt
- Voraussetzung: Hypothesenraum in Form eines lattices, in dem die Bedingung der identischen Satzlänge aller Sätze erfüllt ist

gewünschte Form des Hypothesenraumes



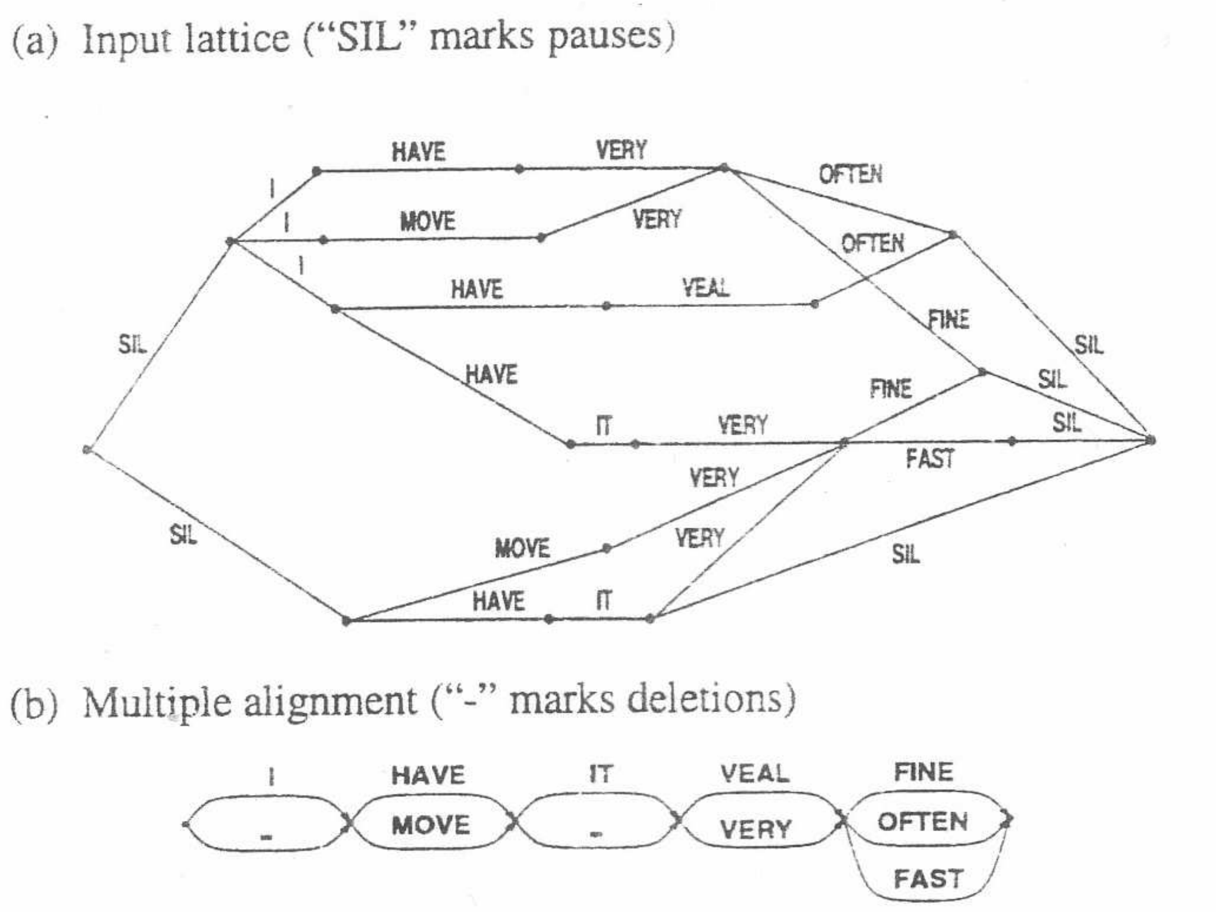
word-lattice-Ansatz - Entscheidungskriterium

Unter den gemachten Vorüberlegungen lässt sich der Wortfehler zwischen zwei Sätzen vereinfachen und somit auch das entstehende Entscheidungskriterium $(M = N = const.)$

$$
\begin{aligned}
\{w_1^N\}_{opt} &= \underset{w_1^N}{\text{argmin}}\{\sum_{v_1^N} p(v_1^N|x_1^T) \cdot L(v_1^N, w_1^N)\} \\
&= \underset{w_1^N}{\text{argmin}}\{\sum_{v_1^N} p(v_1^N|x_1^T) \cdot \sum_{i=1}^{N}(1 - \delta(w_i, v_i))\} \\
&= \underset{w_1^N}{\text{argmin}}\{\sum_{v_1^N} p(v_1^N|x_1^T) \cdot \sum_{i=1}^{N} 1 - \sum_{v_1^N} p(v_1^N|x_1^T) \cdot \sum_{i=1}^{N} \delta(w_i, v_i)\} \\
&= \underset{w_1^N}{\text{argmin}}\{N - \sum_{n=1}^{N} p(w_n|x_1^T)\} \\
&= \underset{w_1^N}{\text{argmax}}\{\sum_{n=1}^{N} p(w_n|x_1^T)\}
\end{aligned}
$$

Ausgangspunkt: Hypothesenraum in Form eines lattice

Ziel: multiple alignment (lattice mit Sätzen identischer Länge)



word-lattice-Ansatz - Transformation II

- Das Ausgangslattice besteht aus einer Kantenmenge E
- Jede Kante $e \in E$ repräsentiert ein Wort $Word(e)$ und hat einen Anfangsknoten $Inode(e)$ und Endknoten $Fnode(e)$
- Eine Kante $e \in E$ hat die Wahrscheinlichkeit $p(e)$, die die Summe aller Satzhypothesen ist, die durch diese Kante laufen
- Zwei Kanten $e, f \in E$ stehen in zeitlicher Relation $\leq$ ("es gibt eine Satzhypothese in der Word(e) vor Word(f) steht"), wenn

$$
\begin{aligned}
e \leq f \quad \Leftrightarrow \quad & e = f && \vee \\
& Fnode(e) = Inode(f) && \vee \\
& \exists e' \in E \quad mit \quad e \leq e' \quad und \quad e' \leq f
\end{aligned}
$$

word-lattice-Ansatz - Transformation III

- **Ziel**: Vereinigen von Kanten zu Kantenmengen $[e_i]$ mit $1 \leq i \leq N$ (Kandidaten für Position i im Satz), $p([e_i])$ ist die Summe der Wahrscheinlichkeiten aller zugehörigen Kanten
- beim Zusammenfassen darf die $\leq$-Ordnung nicht gestört werden, d.h.

$$e \leq f \quad \wedge \quad e \in [e_i] \quad \wedge \quad f \in [e_j] \quad \Rightarrow \quad i \leq j$$

- Werden Kantenmengen vereinigt so verschmelzen Anfangs- und Endknoten der beteiligten Kanten, neue $\leq$-Relationen entstehen
- Kanten e und f die in einer gleichen Äquivalenzklasse e_i vereinigt werden sollten sich phonetisch zumindest ähneln, da die akustische Beobachtung für Position i abgesehen von leichten links-rechts-Verschiebungen konstant ist

intra-word-clustering: Vereinigung von relationsfreien Kanten bzw. Kantenmengen die das gleiche Wort repräsentieren und zeitlich nahe beieinander liegen,

- Ähnlichkeitsmaß:

$$SIM(E_1, E_2) = \max_{e_1 \in E_1, e_2 \in E_2} overlap(e_1, e_2) \cdot p(e_1) \cdot p(e_2)$$

- $overlap(e_1, e_2)$ ist die zeitliche Überlappung der beiden Kanten normiert durch die Summe der beiden Kantenlängen
- iteratives Vereinigen der Mengen E_1 und E_2 mit maximalem $SIM(E_1, E_2)$ bis keine gültige Vereinigung mehr möglich
- nach dem abgeschlossenen intra-word-clustering der Kanten eines Wortes werden die einzelnen entstandenen Kantenmengen als jeweils eine einzige Kante betrachtet, sie erhalten jeweils die Summe der Einzelwahrscheinlichkeiten der enthaltenen Kanten als Kantenwahrscheinlichkeit

inter-word-clustering: Vereinigung von relationsfreien Kanten bzw. Kantenmengen die phonetisch ähnliche Worte repräsentieren

- Ähnlichkeitsmaß:

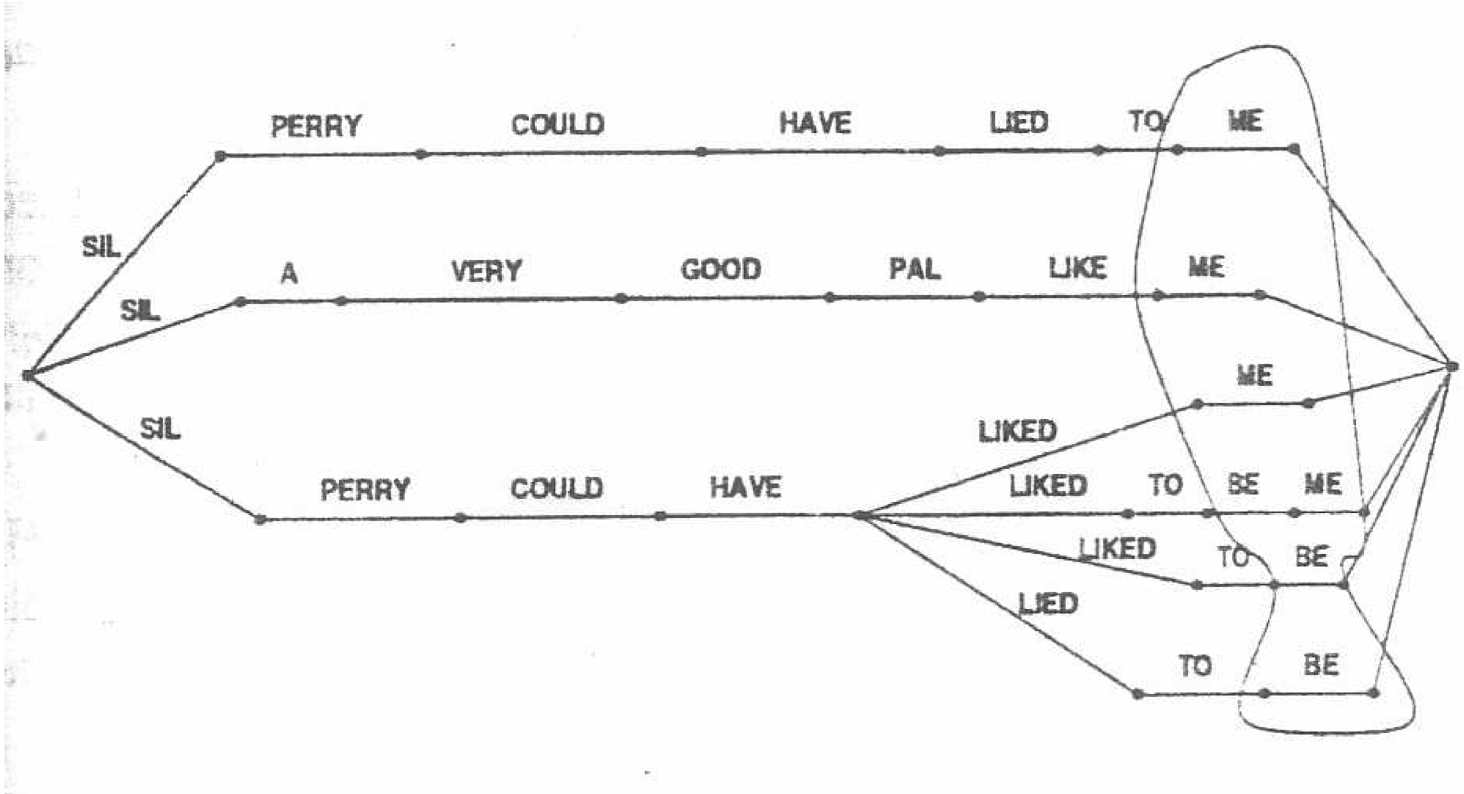
$$SIM(F_1, F_2) = \underset{w_1 \in Words(F_1), w_2 \in Words(F_2)}{\text{avg}} sim(w_1, w_2) \cdot p(\{e \in F_1 : Word(e) = w_1\}) \cdot p(\{e \in F_2 : Word(e) = w_2\})$$

- $sim(w_1, w_2)$ ist z.B. $1 - phonemedistance(w_1, w_2)$, wobei $phonemedistance(w_1, w_2)$ wie der Wortfehler bei Sätzen nur auf Phonemebene bei Worten berechnet wird

- iteratives Vereinigen der Mengen F_1 und F_2 mit maximalem $SIM(F_1, F_2)$ bis keine gültige Vereinigung mehr möglich

- In Kantenmengen, in denen Kanten die verschiedene Worte repräsentieren vereinigt sind, werden die einzelnen Kanten weiterhin getrennt mit eigenen Wahrscheinlichkeiten betrachtet

pruning: Lösen von eher unwahrscheinlichen Kantenverbindungen die sinnvolles clustering verhindern



Die jeweiligen Kantenmengen von "BE" und "ME" werden sinnvollerweise durch inter-word-clustering vereinigt. Dies ist nur möglich wenn die Relation "BE"≤"ME" nicht besteht

Auffüllen der Lücken

- durch lattice-Bearbeitung werden Kanten kürzerer Sätze auseinandergezogen, Lücken entstehen
- Lücken bestehen in Mengen $[e_i]$ deren Wahrscheinlichkeit nicht 1 beträgt
- Auffüllen der Lücken durch Einfügen einer Kante e mit $Word(e) = \epsilon$ und $p(e) = 1 - p([e_i])$

Ablesen der consensus hypothese (Satz mit minimalem Wortfehler)

- zur Erinnerung: ${w_1^N}_{opt} = \underset{w_1^N}{\text{argmax}} \{\sum_{n=1}^N p(w_n|x_1^T)\}$
- wähle aus jeder Menge $[e_i]$ das Wort, dessen zugehörige Kante an der Position die höchste Wahrscheinlichkeit hat

performance des word-lattice-Ansatz

	Switchboard WER [%]	BroadcastNews WER [%]	WS97 dev-test WER [%]
Standardansatz	38.5	33.1	38.5
n-best-Ansatz	37.9	33.0	37.9
lattice-consensus	37.3	32.5	37.1
n-best-consensus	N/A	N/A	37.4

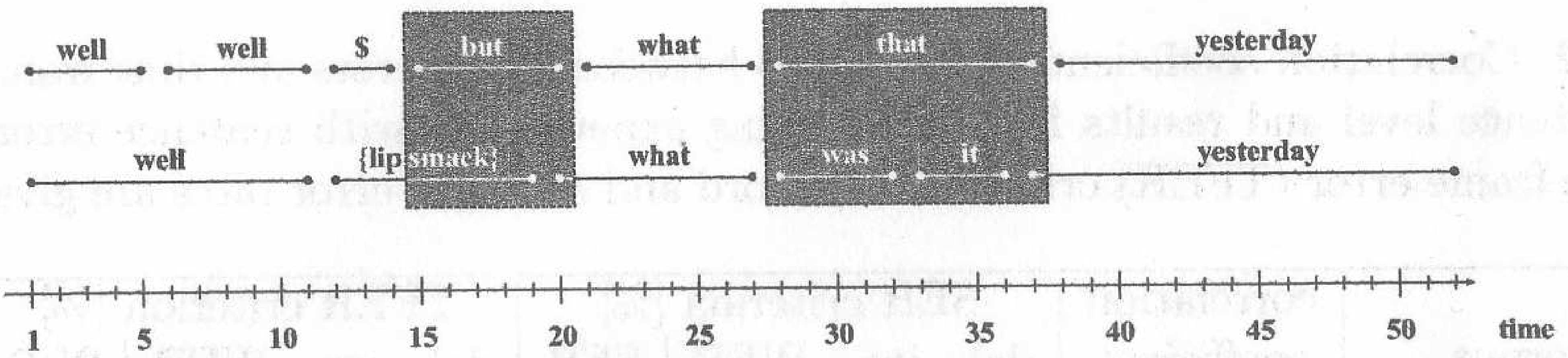
- stärkere Verbesserung der Wortfehlerrate bei lattice-consensus als bei einfachem n-best-Ansatz (0.5%-0.8% absolut), zurückzuführen auf die gemachten Modifikationen in der Satzanpassung
- performance des n-best-consensus besser als n-best-Ansatz (Verbesserung durch Anpassungsmodifikation), aber schlechter als lattice-consensus (Vergrößerung des Hypothesenraumes bei lattice-consensus trägt also auch aktiv zur Verbesserung bei)

Idee: anstatt Hypothesen so zu bearbeiten, daß sie gleichlang und positionsweise zugeordnet werden, Satzhypothesen im lattice in Bezug auf Zeitpunkte einander anpassen

Konsequenz: nicht allein die Wortfolge w_1^N ist interessant, sondern auch Beginn und Ende eines jeden Wortes, daher ab jetzt $[w;t]_1^N$ wobei t jeweils die Wortgrenze von w angibt

Vorteil: auf jedem beliebigen lattice durchführbar, Längengleichheit der Hypothesen nicht gefordert

1:1-Zuordnung von Sätzen in Bezug auf Zeitpunkte - II



Problem: obwohl in Satz 1 die Folge "well, well" und in Satz 2 ein langgezogenes einzelnes "well" gegenüberstehen wird kein "time-frame-error" erkannt; Wortübergänge werden nicht berücksichtigt

Kostenfunktion auf Basis des time-frame-errors

$$\mathcal{C}([w;t]_1^N, [v;\tau]_1^M) = \sum_{n=1}^{N} \frac{\sum_{\hat{t}=t_{n-1}+1}^{t_n} \quad [1 - \sum_{m=1:\tau_{m-1}\leq\hat{t}\leq\tau_m}^{M} \delta(w_n, v_m)]}{1 + \lambda \cdot (t_n - t_{n-1} - 1)}$$

Der Parameter λ dient hierbei der Möglichkeit einer Normalisierung in Bezug auf die betrachtete Wortlänge.

Herleitung des neuen Entscheidungskriteriums

$$
\begin{aligned}
\{[w;t]_1^N\}_{opt} &= \underset{[w;t]_1^N}{\text{argmin}}\{\sum_{[v;\tau]_1^M} \mathcal{C}([w;t]_1^N,[v;\tau]_1^M)\cdot p([v;\tau]_1^M|x_1^T)\} \\
&= \underset{[w;t]_1^N}{\text{argmin}}\{\sum_{n=1}^{N} \frac{\sum_{\hat{t}=t_{n-1}+1}^{t_n} \quad [1-p_{\hat{t}}(w_n|x_1^T)]}{1+\lambda\cdot(t_n-t_{n-1}-1)}\} \\
&=: \underset{[w;t]_1^N}{\text{argmin}}\{\sum_{n=1}^{N} S([w_n;t_{n-1}+1,t_n])\}
\end{aligned}
$$

- $\mathcal{S}([w_n;t_{n-1}+1,t_n])$ kann für jede Worthypothese offline, also unabhängig vom Auswertungsprozess berechnet werden, die Auswertung läuft deshalb ausserordentlich schnell

- Interpretation von $\mathcal{S}([w_n;t_{n-1}+1,t_n])$: "normalisierte Wahrscheinlichkeit, daß w_n im entsprechenden Zeitrahmen falsch ist"

performance der time-frame-error-WEM

		Satzfehler		time frame error	
corpus	Vokabulargröße	WER [%]	SER [%]	WER [%]	SER [%]
ARISE	985	15.8	24.3	15.0	23.6
Verbmobil	7128	33.6	70.2	32.3	71.2
NAB 20k	19987	13.2	79.4	12.9	79.7
NAB 64k	64736	11.1	74.8	10.7	75.8
BN	65491	33.3	91.4	32.3	91.6

- deutliche Verbesserungen der WER bei allen corpora
- Verbesserungen zwischen 2.3% und 5.1% relativ gesehen
- deutlichste Verbesserungen bei corpora mit Spontansprache

Vergleich der Verfahren - I

- Das primäre Ziel der Verbesserung des Wortfehlers erfüllen alle 3 Ansätze deutlich
- Unterschiede in Ansatz und Anzahl der betrachteten Hypothesen
- Auch wenn bei den Verfahren neue Anpassungsarten zwischen Sätzen erstellt wurden, wurde das Ziel der Wortfehlerminimierung erreicht
- Grund: starke Korrellation des Wortfehlers nach Levensthein mit der 1:1-Zuordnung nach Position (durchschnittliche Abweichung von 0.15% absolut) bzw. dem "time-frame-error" (Abweichung zwischen 0.05% und 0.15% absolut an verschiedenen corpora)

Vergleich der Verfahren - II

	Wortfehlerminimierungkriterium		
	n-best-list	word-lattice-consensus	"time-frame-error"
Anpassungsart/ Fehlerart	Levensthein	1:1 nach Wortposition	1:1 nach Zeitpunkt
Nachteile	begrenzter Hypothesenraum	lediglich angenährt	lediglich angenährt

neue Ansatzideen

- Berechnungen beruhen auf klassischer Berechnung der Satz-posteriori-Wahrscheinlichkeiten über Zerlegung in akustisches Modell und Sprachmodell
- komplexere und somit genauere Berechnungen sind in Aussicht
- Verbesserung der jeweiligen performance im Bereich des Möglichen

Anhang - corpora I

- **Switchboard** ist ein corpus der 2400 Telefongespräche zwischen 543 Amerikanern über 70 verschiedene Themen, von denen 50 regelmäßig angewandt wurden, umfasst.
- **CallHome Spanish** ist ein corpus der 120 spontan geführte Telefongespräche zwischen spanischen Verwandten enthält.
- **BroadcastNews** ist ein corpus der Aufzeichnungen von Radio- bzw. Fernsehsendungen von ABC, CNN, NPR, PBS im Zeitraum von Januar 1992 bis Juni 1996 enthält (Vokabulargröße 65491)
- **WS97 dev-test** ist ein corpus der 2427 Aussprüche aus 19 Gesprächen enthält und dabei ca. 18.000 Worte umfasst

- **ARISE** ist ein corpus der Mensch-Maschine-Gespräche, die über Telefon aufgenommen wurden enthält (Vokabulargröße 985)
- **Verbmobil** ist ein deutscher corpus, der verschiedene Anzahlen zwischenmenschlicher Gespräche enthält (Vokabulargröße 7128)
- **NAB 20k / NAB 64k** sind corpora, die Aufnahmen hoher Qualität von vorgelesenen Zeitungsartikel enthalten (Vokabulargröße 20.000 bzw. 64.000 Worte)