Elaborazione Statistica del Linguaggio Naturale

Seminario per il corso di ELN 2002/03

Luca Nicotra

Perchè studiare l'NLP in modo Statistico?

- Fino a circa 5-10 anni fa, NLP era per lo più indagato usando un approccio ***rule-based*** .
- Però, le regole risultano troppo restrittive per caratterizzare l'uso del linguaggio delle persone.
- Questo perchè le persone tendono a modificare e superare le regole per assecondare i loro bisogni comunicativi.
- Sono necessari dei metodi per rendere la modellazione del linguaggio più accurata e i ***metodi statistici*** sembrano fornire una sufficiente stabilità

Suddivisione dell'NLP

- ***Parti del Discorso e Morfologia*** (parole, la loro funzione sintattica nella frase, e le varie forme che puo' assumere).
- ***Struttura delle Frasi e Sintassi*** (regolarita' e vincoli nell'ordine delle parole e nella struttura di parsing)
- ***Semantica***(lo studio del significato delle parole (*semantica lessicale*) e di come i significati delle parole sono combinati per dare significati alle frasi)
- ***Pragmatica***(lo studio di come la conoscenza delle convenzioni del mondo e del linguaggio interferiscano con il significato letterale)

Argomenti della presentazione

- ***Studio dei Termini:***
  - Collocazioni
  - Inferenza statistica
  - Disambiguazione
  - Aquisizione Lessicale
- ***Studio delle Grammatiche:***
  - Modelli di Markov
  - Tagging delle parti del discorso
  - Grammatiche Context Free Probabilistiche

Razionalisti vs Empiristi Approcci al Linguaggio I

- ***Domanda:*** Quale conoscenza a priori dovrebbe essere inserita nei nostri modelli di NLP?
- ***Risposta Rationalista:*** Una parte significativa della conoscenza nella mente umana non è ricavata dai sensi ma è ***fissata*** a priori, presumibilmente per eredità genetica (Chomsky: ***povertà dello stimolo***).
- ***Risposta Empirista:*** Il cervello è in grado di effettuare ***associazioni***, ***riconoscimento di schemi***, e ***generalizzazione***, e, perciò, la struttura del Linguaggio Naturale può essere ***appresa***.

Razionalisti vs Empiristi Approcci al Linguaggio II

- ***La linguistica Chomskyana/generativa*** cerca di descrivere il modulo del linguaggio della mente umana (l'***I-language***) per cui i dati come i testi (l'***E-language***) forniscono solo prove indirette, che possono essere integrate con le intuizioni innate dello speaker.
- Gli approci Empirici sono interessati a descrivere l'E-language come si manifesta in realtà.
- I Chomskyani fanno una distinzione tra ***competenza linguistica*** e ***performance linguistica***. Credono che la competenza linguistica possa essere descritta isolatamente mentre gli Empiricisti rifiutano questa nozione.

L'Approccio di Oggi all'NLP

- Recentemente, c'è stato maggior interesse per le soluzioni ingegneristiche pratiche usando l'apprendimento automatico (knowledge induction).
- Mentre i Chomskyani tendono a concentrari sui giudizi categorici su tipologie di frasi molto rare, l'NLP statistico si concentra sui tipi di frasi comuni.

Perchè l'NLP è difficile?

- NLP è difficile perchè il Linguaggio Naturale è fortemente ambiguo.
- Esempio (purtroppo in Inglese):
- "List the sales of the products produced in 1973 with the products produced in 1972" has 455 parses.
- Perciò, un sistema NLP pratico deve essere in grado di fare decisioni di disambiguazione del senso delle parole, delle categorie delle parole, della struttura sintattica, e del significato semantico.

Metodi che non funzionano bene

- Massimizzare la copertura minimizzando l'ambiguita' non era uno scopo dell'NLP simbolico.
- Inoltre, vincoli sintattici codificati a mano e regole di preferenza richiedono troppo tempo per essere costruiti, non sono scalabili e sono un brittle in the face dell'uso estensivo della metafora nel linguaggio.
- ***Example:*** se codifichiamo

  esseri animati --> **ingoiare** --> oggetto fisico

  I swallowed his story, hook, line, and sinker

  The supernova swallowed the planet.

Cosa possiamo fare con l'NLP Statistico

- Strategie di disambiguazione che si fondano sulla codifica a mano producono un collo di bottiglia nell'acquisizione di conoscenza e si comportano in modo insoddisfacente su testi naturali.
- Un approccio Statistico all'NLP cerca di risolvere questi problemi imparando automaticamente le preferenze lessicali e strutturali dai corpora. In particolare, l'NLP Statistico riconosce che c'e' molta informazione nella relazione tra parole.
- L'uso della statistica offre una buona soluzione al problema dell'amiguita': i modelli statistici sono robusti, generalizzano bene, e si comportano altrettanto bene in presenza di errori e nuovi dati.

Collocazioni

- Una ***collocazione*** e' qualsiasi espressione formata da piu' parole che in qualche modo nel complesso ha un valore che supera la somma delle sue parti.
- Le Collocazioni sono importanti per la traduzione automatica.
- Le Collocazioni possono essere estratte da un testo (ad esempio, si possono estrarre i ***bigram*** che risultano piu' freqeunti). In realta', poiche' questi bigram sono spesso insignificanti (ad esempio "con il", "verso la"), possono essere ***filtrati***.

Collocazioni

- Le collocazioni sono caratterizzate da una ***composizionalità limitata***.
- Larga sovrapposizione tra i concetti di ***collocazione***, ***termine tecnico*** e ***frase terminologica.***

Definizione

- [Una collocazione e' definita come] una sequenza di due o più parole consecutive, che ha caratteristiche di una unità sintattica e semantica, e il cui significato esatto e non ambiguo o connotazione non può essere derivata direttamente dal significato o dalla connotazioni delle sue componenti. [Chouekra, 1988]

Altre Definizioni/Nozioni I

- Le Collocazioni non sono necessariamente adiacenti
- Criteri tipici per le collocazioni: non-composizionalita', non-sostituibilita', non-modificabilita'.
- Le Collocazioni non possono essere tradotte in altri linguaggi.
- Generalizzazioni a casi piu' deboli (forte associzioni di parole ma non necessariamente occorrenze fissate).

Sottoclassi Linguistiche delle Collocazioni

- Particolari costruzioni verbali
- Nomi Propri
- Espressioni Terminologiche

Sommario delle Collocazioni Tecniche per la Rilevazione

- Selezione delle Collocazioni in base alla ***Frequenza***
- Selezione delle Collocazioni in base alla***Media e Varianza*** della distanza tra le parole che le compongono.
- ***Test dell'Ipotesi***
- ***Mutua Informazione***

Frequenza (Justeson & Katz, 1995)

1. Selezionare i bigram che occorrono più frequentemente
2. Passare il risultato attraverso un filtro delle Parti del Discorso.
3. Metodo semplice che funziona bene.

Media e Varianza (Smadja et al., 1993)

- La ricerca basata sulle frequenze lavora bene per espressioni fissate. In realta' molte collocazioni consistono di due parole in relazione tra loro in modo piu' flessibile.
- Il metodo calcola la media e la varianza della distanza tra le due parole nel corpus.
- Se le distanze sono distribuite in modo casuale (cioe', non si tratta di una collocazione), allora la varianza sara' alta.

Test dell'Ipotesi I: Sommario

- Una alta frequenza e una bassa varianza possono essere casuali. Vogliamo determinare se la occorrenza simultanea e' casuale o se avviene in piu' spesso di quanto dovrebbe in una distribuzione casuale.
- Questo e' un problema calssico nella Statistica: il ***Test dell'Ipotesi.***
- Formuliamo una ***ipotesi nulla*** H_0 (nessuna associazione oltre a quelle casuali) e calcoliamo la probabilita' che una collocazione venga riscontrata se H_0 era vera, e quindi ***rifiutiamo*** H_0 se p e' troppo bassa, mentre riteniamo H_0 possibile ,in caso contrario.

Test dell'Ipotesi II: Il t test

- Il t test utilizza la media e la varianza di un campione di misure, dove l'ipotesi nulla e' che il campione sia estratto da una distribuzione con media μ.
- Il test utilizza la differenza tra le medie osservate e le medie attese, scalate dalla varianza dei dati, e ci dice quanto e' probabile estrarre un campione di tale media e varianza assumendo che sia estratto da una distribuzione normale di media μ.
- Per applicare il t test alle collocazioni, pensiamo al corpus di test come una sequenza di N bigram.

Test del Chi-Quadro di Pearson I: Metodo

- L'uso del t test e' stato criticato perche' assume che le probabilita' siano approssimativamente normalmente distribuite (non vero, in genere).
- Il test del Chi-Quadro di non fa questa assunzione.
- L'essenza del test e' di comparare frequenze osservate con frequenze attese per testarne l'indipendenza. Se la differenza tra le frequenze attese e le frequenze rilevate e' grande, allora rigettiamo l'ipotesi nulla di indipendenza.

Testi del Chi-Quadro di Pearson II: Applicazioni

- Uno dei primi utilizzi del test del Chi quadrato nell'NLP Statistico è stata l'identificazione di coppie di traduzioni in corpora allineati (Church & Gale, 1991).
- Una applicazione più recente è l'utilizzo del Chi quadrato come una metrica per la similarità tra corpus (Kilgariff and Rose, 1998)
- In ogni caso, il test del Chi quadrato non dovrebbe essere utilizzato nei corpora piccoli.

Tassi di Verisimiglianza I: All'interno di un singolo corpus (Dunning, 1993)

- I tassi di verosimiglianza sono piu' appropriati per dati sparsi rispetto al test del Chi-Quadro. Inoltre, sono piu' facilmente interpretabili della statistica del Chi-Quadro.
- Applicando il test del grado di verosimiglianza per la ricerca di collocazioni, esaminiamo le due seguenti spiegazioni per la frequenza di occorrenza del bigram w1 w2:
  - L'occorrenza di w2 e' indipendente dalla precedente occorrenza di w1
  - L'occorrenza di w2 e' dipendente dalla precedente occorrenza di w1

Gradi di Verosimiglianza II: Tra due o piu' corpora (Damerau, 1993)

- Tassi di ***frequenze relative*** tra due o piu' corpora differenti possono essere usati per trovare collocazioni che sono caratteristici di un corpus quando paragonati ad altri corpora.
- Questo approccio e' molto utile per la scoperta di collocazioni di uno specifico ambito.

Mutua Informazione

- Una misura basata sulla Teoria dell'Informazione per scoprire collocazioni è la ***mutua informazione puntiforme*** (Church et al., 89, 91)
- La Mutua Informazione Puntiforme è, in breve, una misura di quanto una parola ci dice dell'altra.
- La mutua informazione puntiforme funziona piuttosto male con dati sparsi

Inferenza Statistica: Modelli n-gram su Dati Sparsi

- L'Inferenza Statistica consiste nel prendere dei dati (generati in base ad una distribuzione di probabilità sconosciuta) e quindi fare delle inferenze sulla distribuzione.
- Ci sono tre punti da considerare:
  - Dividere i dati di training in classi di equivalenza
  - Trovare un buono stimatore statistico per ogni classe di equivalenza
  - Combinare stimatori multipli.

Formare Classi di Equivalenza I

- ***Problema di Classificazione***: cercare di predire la caratteristica obiettivo in base alle diverse caratteristiche di classificazione.
- ***Assunzione di Markov***: Solo il precedente contesto locale influenza la prossima entrata: (n-1)th Markov Model or n-gram
- Dimensione dei modelli n-gram vs numero dei parametri: vorremmo utilizzare un n grande, ma il numero dei parametri cresce esponenzialmente con n.
- Esiste un'altro modo di formare classi di equivalenza della storia, ma richiedono metodi più complessi ==> qui useremo le n-gram.

Stimatori Statistici I: Sommario

- ***Obiettivo***: Derivare una buona stima di probabilità per le caratteristiche obiettivo basandosi sui dati osservati
- ***Esempio***: Da $P(w_1,..,w_n)$ predire $P(w_n|w_1,..,w_{n-1})$
- ***Soluzioni che prenderemo in esame***:
  - Stima di Massima Verosimiglianza
  - Leggi di Laplace, Lidstone e Jeffreys-Perks
  - Held Out Estimation
  - Cross-Validation
  - Stima di Good-Turing

Stimatori Statistici II: *Stima di Massima Verisimiglianza*

- $P_{MLE}(w_1,..,w_n)=C(w_1,..,w_n)/N$, dove $C(w_1,..,w_n)$ è la frequenza della n-gram $w_1,..,w_n$
- $P_{MLE}(w_n|w_1,..,w_{n-1})= C(w_1,..,w_n)/C(w_1,..,w_{n-1})$
- Questa stima viene chiamata ***Stima di Massima Verisimiglianza*** (MLE) perchè è la scelta dei parametri che assegna la più alta probabilità al corpus usato per l'apprendimento.
- MLE solitamente non è adatto per l'NLP per la sparsità dei dati ==> Uso di techniche di ***Discounting*** o ***Smoothing***.

Stimatori Statistici III: Tecniche di Smoothing: *Laplace*

- $P_{LAP}(w_1,..,w_n)=(C(w_1,..,w_n)+1)/(N+B)$, where $C(w_1,..,w_n)$ is the frequency of n-gram $w_1,..,w_n$ and B is the number of bins training instances are divided into. ==> ***Adding One*** Process
- L'idea è di date una piccola probabilità agli eventi non visti.
- In ogni caso, in applicazioni di NLP veramente sparse, la Legge di Laplace in realtà assegna probabilità troppo elevate agli eventi non visti.

Stimatori Statistici IV: Tecniche di Smoothing: *Lidstone e Jeffrey-Perks*

- Poichè aggiungendo uno potremmo aggiungere troppo, possiamo aggiungere un valore minore λ.
- $P_{LID}(w_1,..,w_n)=(C(w_1,..,w_n)+\lambda)/(N+B\lambda)$, dove $C(w_1,..,w_n)$ è la frequenza della n-gram $w_1,..,w_n$ e B è il numero di gruppi in cui le istanze di addestramento vengono divise, e $\lambda>0$. ==> ***Legge di Lidstone***
- Se $\lambda=1/2$, la Legge di Lidstone corrisponde alla speranza della verisimiglianza e viene chiamata ***Expected Likelihood Estimation*** (ELE) o la Legge di ***Jeffreys-Perks***.

Stimatori Statistici V: Tecniche Robuste: *Stima Held Out*

- Per ogni n-gram, $w_1,..,w_n$, calcoliamo $C_1(w_1,..,w_n)$ e $C_2(w_1,..,w_n)$, le frequenze di $w_1,..,w_n$ nei dati di addestramento e nei dati held out, rispettivamente.
- Sia N_r il numero di bigram con frequenza r nel testo di addestramento.
- Sia T_r il numero totale di volte in cui tutte le n-gram che sono apparse r volte nel testo di addestramento sono apparse nei dati held out.
- Una stima per la probabilità di una di queste n-gram è: $P_{ho}(w_1,..,w_n)= T_r/(N_rN)$ dove $C(w_1,..,w_n) = r$.

Stimatori Statistici VI: Tecniche Robuste: *Cross-Validation*

- La stima Held Out è utile se ci sono molti dati disponibili. Altrimenti, è utile usare ogni parte dei dati sia come dati di addestramento che come dati held out.
- ***Deleted Estimation*** [Jelinek & Mercer, 1985]: Sia N_r^a il numero di n-grams che ricorrono r volte nella parte a-esima dei dati di addestramento e sia T_r^{ab} il numero totale di occorrenze di quei bigram della parte a nella parte b. $Pdel(w_1,..,w_n)= (T_r^{01}+T_r^{10})/N(N_r^0+N_r^1)$ dove $C(w_1,..,w_n) = r$.
- ***Leave-One-Out*** [Ney et al., 1997]

Stimatori Statistici VI: Approcci collegati: *Stimatore di Good-Turing*

- Se $C(w_1,..,w_n) = r > 0$, $P_{GT}(w_1,..,w_n) = r^*/N$ where $r^*=((r+1)S(r+1))/S(r)$ e $S(r)$ è una stima smoothed della speranza di N_r.
- If $C(w_1,..,w_n) = 0$, $P_{GT}(w_1,..,w_n) \approx N_1/(N_0N)$
- ***Good-Turing Semplice*** [Gale & Sampson, 1995]: Come curva di smoothing, usa $N_r=ar^b$ (with $b < -1$) e stima a e b con una semplice regressione lineare con la forma logaritmica di questa equazione:
- $\log N_r = \log a + b \log r$, se r è grande. Per bassi valori di r, usare direttamente l'N_r misurato.

Combinare Stimatori I: Sommario

- Se ci sono diversi modi in cui la storia ci può predire cosa viene dopo, allora potremmo volerli combinare nella speranza di produrre un modello persino migliore.
- Metodi di Combinazione Considerati:
  - Interpolazione Lineare Semplice
  - Il Backing Off di Katz
  - Interpolazione Lineare Generale

Combinare Stimatori II: Interpolazione Lineare Semplice

- Un modo per risolvere la sparsità nei modelli trigram è di combinarli con modelli bigram e unigram che soffrono meno della sparsità dei dati.
- Questo può essere fatto per mezzo della ***interpolatione lineare*** (chiamata anche ***finite mixture models***). Quando le funzioni che vengono interpolate usano tutte un sottoinsieme delle informazioni di condizionamento della funzione maggiormente discriminante, il metodo viene detto ***interpolatione cancellata***.
- $P_{li}(w_n|w_{n-2},w_{n-1})=\lambda_1P_1(w_n)+ \lambda_2P_2(w_n|w_{n-1})+ \lambda_3P_3(w_n|w_{n-1},w_{n-2})$ dove $0\leq\lambda_i\leq1$ e $\Sigma_i \lambda_i =1$
- I pesi possono essere impostati automaticamente usando l'algoritmo di Massimizzazione dell'Attesa (Expectation-Maximization (EM)).

Combinare Stimatori II: Modello Backing Off di Katz

- Nei modelli back-off, modelli differenti vengono consultati in ordine in base alla loro specificità.
- Se la n-gram che ci interessa è apparsa più di k volte, allora viene usata la stima della n-gram ma una parte della stima MLE viene discounted (è riservata per le n-gram non viste).
- Se la n-gram è apparsa k volte o meno, allora usiamo una stima di una n-gram più breve (probabilità di back-off), normalizzata per la probabilità rimanente e la quantità di dati coperti da questa stima. Il processo continua ricorsivamente.

Combinare Stimatori II: Interpolazione Lineare Generale

- Nella Interpolazione Semplice Lineare, i pesi erano un singolo numero, ma è possibile definire un modello più generale e potente in cui i pesi siano una funzione della storia.
- Per k funzioni di probabilità Pk, la forma generale per un modello di interpolazione è: $P_{li}(w|h) = \Sigma_i^k \lambda_i(h)\, P_i(w|h)$ dove $0 \le \lambda_i(h) \le 1$ e $\Sigma_i \lambda_i(h) = 1$

Disambiguazione del Significato delle Parole

- ***Problema:*** molte parole hanno significati diversi ==> c'è ambiguità nel modo in cui vengono interpretate.
- ***Obiettivo:*** determinare quale dei significati di una parola ambigua viene evocato in un uso particolare della parola. Questo viene fatto guardando al contesto dell'uso della parola.
- ***Nota:*** molto spesso i diversi significati di una parola sono fortemente in relazione.

Sommario della Discussione

- ***Metodologia***
- ***Disambiguazione Supervisionata***: basata su un insieme di apprendimento etichettato.
- ***Disambiguazione Basata su Dizionario***: basata su risorse lessicali come dizionari o thesauri.
- ***Disambiguazione Non Supervisionata***: basata su corpora non etichettati.

Preliminari Metodologici

- ***Apprendimento Supervisionato contro Non Supervisionato***: nell'apprendimento supervisionato è conosciuta l'etichetta del significato di una parola. Nell'apprendimento non supervisionato, non è conosciuta.
- ***Pseudoparole***: usate per generare valutazioni artificiali dei dati per confronti e test dei miglioramenti degli algoritmi di processamento di testi.
- ***Limiti Superiori e Inferiori alla Performance***: usati per scoprire quanto bene si comporta un algoritmo in relazione alla difficoltà del compito.

Disambiguazione Supervisionata

- ***Insieme di Addestramento***: esempi in cui ogni occorrenza della parola ambigua w viene annotata con una etichetta semantica ==> Problema di Classificazione.
- ***Approci***:
  - Classificazione Bayesiana: il contesto delle occorrenze viene trattato come un insieme di parole senza struttura, ma integra informazioni da molte parole.
  - Teoria dell'Informazione: guarda solo alle caratteristiche informative nel contesto. Queste caratteristiche possono essere sensibili alla struttura del testo.
  - Ci sono molti più approcci (Machine Learning…).

Disambiguazione Supervisionata: Classificaizone Bayesiana I

- ***Idea di (Gale et al, 1992):*** guardare alle parole attorno ad una parola ambigua in una ampia finestra contestuale. Ogni parola del contesto potenzialmente contribuisce con informazione utile a capire quale significato viene assunto più probabilmente dalla parola ambigua. Il classificatore non fa alcuna selezione delle caratteristiche. Invece, combina le prove da tutte le caratteristiche.
- ***Regola di decisione di Bayes***: Decide s' se $P(s'|C) > P(s_k|C)$ per $s_k \neq s'$.
- $P(s_k|C)$ viene calcolato con la Regola di Bayes.

Disambiguazione Supervisionata: Classificazione Bayesiana II

- ***Assunzione Naïve di Bayes***: $P(C|s_k) = P(\{v_j | v_j \text{ in } C\} | s_k) = \Pi_{v_j \text{ in } C} P(v_j | s_k)$
- L'assunzione Naïve di Bayes non è corretta nel contesto del processamento del testo, ma è utile.
- ***Decisionrule for Naïve Bayes***: Decide s' se $s' = \text{argmax}_{s_k} [\log P(s_k) + \Sigma_{v_j \text{ in } C} \log P(v_j | s_k)]$
- $P(v_j | s_k)$ e $P(s_k)$ vengono calcolate per mezzo della Stima di Massima Verosimiglianza, forse con uno smoothing appropriato, dal corpus di addestramento etichettato..

Disambiguazione Supervisionata: Un Approccio basato sulla Teoria dell'Informazione

- ***Idea di (Brown et al., 1991):*** trovare una singola caratteristica contestuale che indichi in modo affidabile quale significato della parola ambigua viene utilizzato.
- L'algoritmo ***Flip-Flop*** viene usato per disambiguare tra significati differenti di una parola utilizzando la mutua informazione come misura.
- $I(X;Y)=\Sigma_{x\in X}\Sigma_{y\in Y}p(x,y)\log p(x,y)/(p(x)p(y))$
- L'algoritmo lavora cercando una partizione dei significati che massimizzi la mutua informazione. L'algoritmo si ferma quando l'aumento diventa insigificante.

Disambiguazione Basata su Dizionario: Sommario

- Esamineremo tre metodi differenti:
  - Disambiguazione basata sulla definizione dei significati.
  - Disambiguzione basata su Vocabolario dei Sinonimi
  - Disambiguazione basata su traduzione in un corpus di un secondo linguaggio.
- Mostreremo anche come un esame accurato delle proprietà delle distribuzioni dei significati può portare a dei miglioramenti significativi nella disambiguzione.

Disambiguazione basata sualla definizione dei significati

- ***(Lesk, 1986: Idea)***: le definizioni di un dizionario di una parola probabilmente sono un buon indicatore del significato che definisce.
- Esprimere le sotto-definizioni del dizionario della parola ambigua come un insieme di gruppi (bag-of-words) e le parole che occorrono nel contesto di una parola ambigua come un singolo gruppo (bags-of-words) partendo dalle sue definizioni del dizionario.
- Disambiguare le parole ambigue scegliendo le sotto-definizioni della parola ambigua che ha la più alta sovrapposizione con le parole che occorrono nel suo contesto.

Disambiguazione Basata su Dizionario dei Sinonimi

- ***Idea:*** le categorie semantiche di una parola in un contesto determinano la categoria semantica del contesto come un tutt'uno. Questa categoria determina quale significato della parola viene utilizzato.
- ***(Walker, 87):*** ad ogni parola viene assegnato uno o più codici contesto che corrispondono ai suoi differenti significati. Per ogni codice contesto, contiamo il numero di parole (provenienti dal contesto) che hanno lo stesso codice contesto corrispondente alla conta più alta.
- ***(Yarowski, 92):*** ha adattato l'algoritmo per parole che non si verificano nel thesaurus ma sono molto Informative. E.g., Navratilova --> Sports

Disambiguazione basata sulla traduzione in un corpus in un secondo linguaggio

- ***(Dagan & Itai, 91, 91)'s Idea***: le parole possono essere disambiguate guardando a come vengono tradotte in altri linguaggi.
- Esempio: la parola "interest" ha due traduzioni in Tedesco: 1) "Beteiligung" (legal share--50% a interest in the company) 2) "Interesse" (attenzione, concern—il suo interesse in matematica).
- Per disambiguare la parola "interest", identifichiamo la parola in cui ricorre, cerchiamo in un corpus Tedesco istanze della frase, e assegnamo lo stesso significato associato con l'uso Tedesco della parola in quella frase.

Un significato per discorso, un significato per collocazione

- ***Idea di (Yarowsky, 1995)***: ci sono vincoli tra occorrenze diverse di una parola ambigua all'interno di un corpus che può essere sfruttato per la disambiguazione:
  - ***Un significato per discorso***: Il significato di una parola obiettivo è fortemente consistente all'interno di un dato documento.
  - ***Un significato per collocazione***: parole vicine forniscono indizi forti e consistenti del senso di una parola obiettivo, in relazione alla distanza relativa, all'ordine e alle relazioni sintattiche.

Disambiguazione Non Supervisionata

- ***Idea:*** disambiguare i significati delle parole senza ricorrere a strumenti di supporto come dizionari o thesauri e in assenza di un testo etichettato. Semplicemente clusterizzare i contesti di una parola ambigua in un insieme di gruppi e discriminare tra questi gruppi senza etichettarli.
- ***(Schutze, 1998):*** Il modello probabilistico è lo stesso modello Bayesiano utilizzato per la classificazione supervisionata, ma le $P(v_j \mid s_k)$ vengono stimate utilizzando l'algoritmo di EM.

Acquisizione Lessicale

- ***Obiettivo:*** Sviluppare algoritmi e tecniche statistiche per riempire i buchi nei dizionari consultabili dalle macchine cercando gli schemi di occorrenza delle parole nei corpora con molto testo.
- Acquisire collocazioni e disambiguazione del senso delle parole sono esempi di acquisizione lessicale, ma ce non sono molti altri tipi.
- ***Esempi del problema della acquisizione lessicale:*** preferenze selezionali, frame di sottocategorizzazione, categorizzazione semantica.

A cosa serve l'Acquisizione Lessicale?

- Il Linguaggio evolve, cioè nuove parole e nuovi usi di vecchie parole vengono continamente inventati.
- I Dizionari Tradizionali erano scritti per gli scopi di utenti umani. I Lexicon sono dizionari formattati per computer. Oltre al formato, i lexicon possono essere utili se contengono informazione quantitativa. L'acquisizione lessicale può fornire tale informazione.
- I Dizionari Tradizionali segnano un netto confine tra informazione lessicale e non-lessicale. Può essere utile eliminare questa distinzione.

Sommario

- Questione Metodologica: Misure di Valutazione
- Sottocategorizzazione dei Verbi
- Ambiguita' di Attachment
- Preferenze Selezionali
- Similarità Semantica

Misure di Valutazione

- Precisione e Richiamo
- Misura F
- Precisione e Richiamo versus Accuratezza ed Errore
- Fallout
- Curva Receiver Operating Characteristic (ROC)

Sottocategorizzaizone dei Verbi I

- I verbi esprimono la loro categoria semantica usando differenti mezzi sintattici. Un insieme particolare di categorie sintattiche con cui può apparire un verbo viene detto frame di sottocategorizzazione.
- La maggior parte dei dizionari non contengono informazione sui frame di sottocategorizzazione.
- Il sistema di apprendimento dei frame di sottocategorizzazione di (Brent, 93) cerca di decidere in base alle prove del corpus se un verbo *v* prende il frame *f*. Funziona in due passi.

Sottocategorizzazione dei Verbi II

Sistema di Apprendimento di Brent:

- ***Indizi:*** Definire uno schema regolare di parole e categorie sintattiche che indicano la presenza del frame con un'alta sicurezza. Per un particolare indizio c^j definiamo una probabiltà d'errore ε_j che indica quanto probabilmente sbaglieremo nell'assengare il frame ***f*** al verbo ***v*** basandoci sull'indizio c^j.
- ***Test dell'Ipotesi:*** Definiamo l'ipotesi nulla, H_0, come: "il frame non è appropriato per il verbo". Rifiuta questa ipotesi se l'indizio c^j indica con alta probabilità che la nostra H_0 è errata.

Sottocategorizzazione dei Verbi III

- Il sistema di Brent è preciso ma non ha buone performance nel richiamo.
- Il sistema di Manning(Manning, 93) si rivolge a questo problema utilizzando un tagger e eseguendo la ricerca di indizi sull'output del tagger.
- Il metodo di Manning può apprendere un gran numero di frame di sottocategorizzazione, perfino quelli che hanno indizi a bassa affidabilità.
- I risultati di Manning sono ancora bassi e un modo per migliorarli è quello di utilizzare conoscenza a priori.

Ambiguità di Attachment I

- Quando cerchiamo di determinare la struttura sintattica di una frase, spesso ci sono frasi che possono esssere collegate a due o più nodi differenti dell'albero. Qual'è quello corretto? Un semplice modello per questo problema consiste nel calcolare il seguente tasso di verisimiglianza: $\lambda(v, n, p) = \log(P(p|v)/P(p|n))$ dove $P(p|v)$ è la probabiltà di vedere un PP con ***p*** dopo il verbo ***v*** e $P(p|n)$ è la probabilità di vedere un PP con ***p*** dopo il nome ***n***.
- Debolezza di questo modello: ignora il fatto che , quando le altre considerazioni risultano equivalenti, c'è una preferenza per attaccare le frasi in basso nel parse tree.

Ambiguita' di Attachment II

- Il vincolo preferenziale per attachment bassi nel parse tree è formalizzato da ***(Hindle and Rooth, 1993)***
- Il modello si pone le seguenti domande:
- Va_p: C'è un PP che comincia ***p*** e che segue il verbo ***v*** che si attacca a ***v*** ($Va_p=1$) oppure no ($Va_p=0$)?
- Na_p: C'è un PP che comincia per ***p*** e che segue il nome ***n*** che si attacca a n ($Na_p=1$) oppure no ($Na_p=0$)?
- Calcoliamo $P(Attach(p)=n|v,n)=P(Na_p=1|n)$ e $P(Attach(p)=v|v,n)=P(Va_p=1|v)\ P(Na_p=0|n)$.

Ambiguita' di Attachment III

- P(Attach(p)=v) e P(Attach(p)=n) possono essere calcolato per mezzo del tasso di verisimiglianza λ dove $\lambda(v, n, p) = \log (P(Va_p=1/v)\, P(Na_p=0/n))/ P(Na_p=1/n)$
- Stimiamo le necessarie probabilità usando la stima di massima verisimiglianza:
- $P(Va_p=1/v)=C(v,p)/C(v)$
- $P(Na_p=1/n)=C(n,p)/C(n)$

PP Attachment

- Ci sono alcune limitazioni nel metodo di Hindle e Rooth:
- In qualche caso sono utili informazioni diverse da v, n e p.
- Ci sono altri tipi di PP attachment oltre al caso base di un PP immediatamente dopo un oggetto NP.
- Ci sono altri tipi di attachments altogether: N N N o V N P. Il formalismo di Hindle and Rooth è più difficile da applicare in questi casi per la sparsità dei dati.
- In certi case, c'è ***indeterminatezza di attachment.***

Preferenze Selezionali I

- La maggior parte dei verbi preferiscono gli argomenti di un tipo particolare (ad esempio, le cose che abbaiano sono cani). Queste regolarità sono chiamate ***preferenze selezionali*** o ***restrizioni selezionali***.
- Le preferenze selezionali sono utili per alcune ragioni:
  - Se una parola è una forma mancante dal nostro dizionario leggibile dal computer, aspetti del suo significato possono essere inferiti da restrizioni selezionali.
  - Le preferenze selezionali possono essere usate per dare un punteggio a parse differenti di una frase.

Preferenze Selezionali II

- L'idea di ***Resnik (1993, 1996)*** per le Preferenze Selezionali usa la nozione di ***forza di preferenza selezionale*** e ***associazione selezionale***. Ci interessiamo al Problema ***<Verb, Direct Object>.***
- La ***Forza di Preferenza Selezionale***, ***S(v)*** misura quanto fortemente un verbo influenza il suo oggetto diretto.
- S(v) viene definito coem la ***Divergenza KL*** tra la distribuzione a priori dell'oggetto diretto (per I verbi in generale) e la distribuzione degli oggetti diretti del verbo che stiamo cercando di caratterizzare.
- Facciamo 2 assunzioni in questo modello: 1) solo il nome di testa dell'oggetto viene considerato; 2) piuttosto che trattare con nomi individualmente, trattiamo classi di nomi.

Preferenze Selezionali III

- The ***Associazioni Selezionali*** tra un verbo e una classe viene definita come la proporzione che il contributo di questa classe a S(v) cotribuisce alla forza di preferenza totale di S(v).
- Ci sono anche regole per assegnare forze di associazione a nomi as opposed to noun classes. Se un nome è in una singola classe, allora la sua forza di associazione è quella di quella classe. Se appartiene a diverse classi, allora la sua forza di associazione è qeulla della classe a cui appartiene che ha la più alta forza di associazione.
- Infine, c'è una regola per sitmare la probabiltà che un oggetto diretto in una classe di nomi c occorra dato un verbo v.

Similarità Semantica I

- La Comprensione dei Testi o Information Retrieval può trarre molto vantaggio da un sistema in grado di acquisire significati.
- L'acquisizione di significati non è possibile a questo punto, cosìi ci si concentra nel assegnare ***similarità*** tra una nuova parola e altre parole già conosciute.
- La somiglianza semantica non è una nozione così intuitiva e chiara come potremmo pensare: sinonimi? Stesso dominio semantico? Intercambiabilità contestuale?
- Spazi Vettoriali vs Misure Probabilistiche

Similarità Semantica II: Misure di Spazi Vettoriali

- Le parole possono essere espresse in spazi differenti: ***document space***, ***word space*** and ***modifier space***.
- Misure di similarità tra vettori binari: ***matching coefficient***, ***Dice coefficient***, ***Jaccard*** (or ***Tanimoto***) ***coefficient***, ***Overlap coefficient*** and ***cosine***.
- Misure di similarità per ***spazi vettoriali*** a valori reali: ***cosine***, ***Euclidean Distance***, ***normalized correlation coefficient***

Similarità Semantica II: Misure Probabilistiche

- Il problema delle misure basate su spazi vettoriali è che, a parte il coseno, operano su dati binari. Il coseno, d'altra parte, assume uno spazio Euclideo che non è ben motivato quando abbiamo a che fare con la conta delle parole.
- Un modo migliore per vedere la conta di una parola si ottiene rappresentandoli come distribuzioni di probabilità.
- Quindi possiamo confrontare due distribuzioni di probabilità usando le seguenti misure: ***Divergenza KL***, ***Raggio di Informazione*** (***Irad***) and ***Norma L_1***.

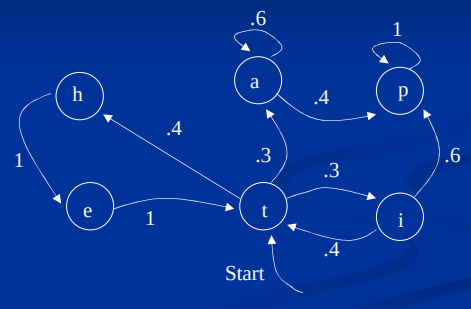
Modelli di Markov

- I modelli di Markov sono strumenti statistici utili per l'NLP poiché possono essere utlizzati per applicazioni di tagging delle parti del discorso.
- Il loro primo utlizzo fu per la modellazione della sequenze di lettere in opere della letteratura Russa.
- In seguito furono sviluppate come strumenti statistici generali.
- Piu' specificamente, modellano una sequenza (per esempio nel tempo) di variabili aleatorie che non sono necessariamente indipendenti.
- Si fondano su due assunzioni: ***Orizzonte Limitato*** e ***Invarianza Temporale***.

Assunzioni di Markov

- Sia $X=(X_1, .., X_t)$ una sequenza di variabili aleatorie che assumono valori in un insieme finito $S=\{s_1, \ldots, s_n\}$, lo spazio degli stati, le proprieta' di Markov sono:
- ***Orizzonte Limitato***: $P(X_{t+1}=s_k \mid X_1, .., X_t)=P(X_{t+1} = s_k \mid X_t)$ cioe', il tag di una parola dipende solamente dal tag precedente.
- ***Invarianza Temporale***: $P(X_{t+1}=s_k \mid X_1, .., X_t)=P(X2 =s_k \mid X_1)$ cioe', la dipendenza non cambia nel tempo.
- Se X possiede queste proprieta', allora si dice che X e' una catena di Markov.

Esempio di una Catena di Markov



Modelli Nascosti di Markov (HMM)

- In un HMM, non si conosce la sequenza di stati attraversata dal modello, ma solo una sua funzione probabilistica.
- Esempio: Il distributore di bibite pazzoide: puo' essere in due stati, uno in cui preferisce la coca cola, e uno in cui preferisce il te freddo, ma cambia tra i due stati casualmente dopo ogni acquisto in base a qualche probabilita'.
- La domanda e': Qual'e' la probabilita' di vedere una particolare sequenza di output sapendo lo stato di partenza?

Perche' usare i Modelli Nascosti di Markov?

- Gli HMM sono utili quando possiamo interpretare gli eventi osservati come generati probabilisticamente da eventi sottostanti. Esempio: Part-of-Speech-Tagging.
- Gli HMM possono essere addestrati in modo efficiente usando l'Algoritmo di EM.
- Un altro campo in cui gli HMM sono utili e' quello della generazione di parametri per l'interpolazione lineare di modelli n-gram.

Forma Generale di un HMM

- Un HMM e' definito da una quintupla (S, K, Π, A, B) dove S e K sono gli stati e l'alfabeto di output, e Π, A, B sono le probabilita' dello stato iniziale, delle transizioni tra stati, e dell'emissione di simboli, rispettivamente.
- Data la definizione di un HMM, possiamo simulare l'esecuzione di un processo di Markov e produrre una sequenza di output usando l'algoritmo mostrato nella prossima diapositiva.
- In realta' a noi, piu' della simulazione, ci interessa assumere che un insieme di dati sia stato generato da una HMM, per quindi essere in grado di calcolare le probabilita' e la probabile sequenza di stati sottostante.

Un programma per un Processo di Markov

t:= 1;

Comincia nello stato s_i con probabilita' π_i (cioe', $X_1=i$)

Forever do

Spostati dallo stato s_i allo stato s_j con probabilita' a_{ij} (cioe', $X_{t+1} = j$)

Emetti il simbolo osservabile $o_t = k$ con probabilita' b_{ijk}

t:= t+1

End

Le tre Domande Fondamentali per i HMM

- Dato un modello μ=(A, B, Π), come calcoliamo efficacemente quanto probabile e' una certa osservazione, cioe', $P(O|\mu)$?
- Data la sequenza di osservazioni O e un modello μ, come scegliamo la sequenza di stati $(X_1, \ldots, X_{T+1})$ che meglio spiega le osservazioni?
- Data una sequenza di osservazioni O, e lo spazio dei possibili modelli ottenuto variando i parametetri del modello $\mu = (A, B, \pi)$, come troviamo il modello che meglio spiega i dati?

Trovare la probabilita' di una osservaizone I

- Data la sequenza di osservazioni O=(o_1, …, o_T) e un modello μ= (A, B, $\prod$), vogliamo sapere come calcolare efficentemente P(O| μ). Questo processo viene chiamato ***decodifica***.
- Per ogni sequenza di stati X=(X1, …, XT+1), troviamo: $P(O|\mu)=\Sigma_{X1\ldots XT+1} \pi_{X1} \prod_{t=1}^{T} a_{XtXt+1} b_{XtXt+1ot}$
- Questa e' semplicemnte la somma delle probabilita' dell'osservazione in base ad ogni possibile sequenza di stati.
- In ogni caso, la valutazione diretta di questa espressione e' molto inefficiente.

Trovare la probabilita' di una osservazione II

- Per evitare questa complessita', possiamo usare tecniche di ***programmazione dinamica*** o ***memorizzazione***t.
- In particolare, usiamo l'algoritmo del ***treillis*** .
- Creiamo un array quadrato di stati disposti lungo il tempo e calcoliamo le probabilita' di essere ad ogni stato in ogni momento in termini delle probabilita' di essere in ogni stato al tempo precedente.
- Un treillis puo' salvare la probabilita' di tutti i sottocammini iniziali del HMM che finiscono in un certo stato ad un certo istante temporale. La probabilita' di sottocammini piu' lunghi puo' quindi essere ricavata in termini di sottocammini piu' brevi.

Trovare la probabilita' di una osservaizone III: La *forward procedure*

- Una ***variabile forward***, $\alpha_i(t)= P(o_1o_2\ldots o_{t-1}, X_t=i/ \mu)$ ***viene salvata in*** (s_i, t) nel trellis ed esprime la probabilita' totale di finire nello stato si al tempo t.
- Le variabili Forward vengono calcolate nel modo seguente:
- ***Inizializzazione***: $\alpha_i(1)= \pi_i$, $1\leq i \leq N$
- ***Induzione***: $\delta_j(t+1)=\Sigma_{i=1}^{N}\alpha_i(t)a_{ij}b_{ijot}$, $1\leq t\leq T$, $1\leq j\leq N$
- ***Totale***: $P(O|\mu)= \Sigma_{i=1}^{N}\alpha_i(T+1)$
- Questo algoritmo richide $2N^2T$ moltiplicazioni (piu' o meno un metodo diretto richiede $(2T+1).N^{T+1}$

Trovare la Probabilita' di una Osservazione IV: La *backward procedure*

- La backward procedure calcola le ***variabili backward*** che sono le probabilita' totali di vedere il resto della sequenza di osservazioni dato che siamo nello stato s_i al tempo t.
- Le variabili Backward sono utili per il problema della stima dei parametri.

Trovare la Probabilita' di una Osservazione V:La *backward procedure*

- Siano $\beta_i(t) = P(o_t \ldots o_T \mid X_t = i, \mu)$ le variabili backward.
- Le variabili Backward possono essere calcolate spostandoci all'inditro attraverso il treillis nel modo seguente:
- ***Inizializzazione***: $\beta_i(T+1) = 1,\ 1 \le i \le N$
- ***Induzione:*** $\Sigma_{j=1}^{N} a_{ij} b_{ijo_t} \beta_j(t+1),\ 1 \le t \le T,\ 1 \le i \le N$
- ***Totale:*** $P(O \mid \mu) = \Sigma_{i=1}^{N} \pi_i \beta_i(1)$
- Le variabili Backward possono essere anche combinate con le variabili forward:
  $P(O \mid \mu) = \Sigma_{i=1}^{N} \alpha_i(t) \beta_i(t),\ 1 \le t \le T+1$

Trovare la migliore sequenza di stati I

- Un metodo consiste nel trovare gli stati individualmente:
- Per ogni t, $1 \le t \le T+1$, vogliamo trovare X_t che massimizza $P(X_t \mid O, \mu)$.
- Sia $\gamma_i(t) = P(X_t = i \mid O, \mu) = P(X_t = i, O \mid \mu)/P(O \mid \mu) = (\alpha_i(t)\beta_i(t)/\Sigma_{j=1}^{N} \alpha_j(t)\beta_j(t))$
- Lo stato individualmente piu' probabile e'
  $\hat{X}_t = argmax_{1 \le i \le N} \gamma_i(t),\ 1 \le t \le T+1$
- Questa quantita' massimizza il numero atteso di stati che saranno decisi correttamente. Pero', puo' portare ad una sequenza di stati piuttosto improbabile.

Trovare la Migliore Sequenza di Stati II: L'Algoritmo di Viterbi

- L' ***Algoritmo di Viterbi*** calcola in modo efficiente la sequenza piu' probabile di stati.
- Comunemente, vogliamo trovare il percorso globalmente piu' probabile, cioe': $argmax_X P(X \mid O, \mu)$
- Per fare questo, e' sufficiente per un O fissato : $argmax_X P(X, O \mid \mu)$
- ***Definiamo:***
  $\delta_i(t) = max_{X1 \ldots Xt-1} P(X_1 \ldots X_{t-1}, o_1 \ldots o_{t-1}, X_t = j \mid \mu)$
  $\psi_j(t)$ salva il nodo dell'arco in entrata che porta al cammino piu' probabile.

Trovare la migliore sequenza di stati II: L'Algoritmo di Viterbi

L'algoritmo di Viterbi funziona nel modo seguente:

- ***Inizializzazione***: $\delta_j(1) = \pi_j,\ 1 \le j \le N$
- ***Induzione:*** $\delta_j(t+1) = max_{1 \le i \le N} \delta_i(t) a_{ij} b_{ijo_t},\ 1 \le j \le N$
  Store backtrace:
  $\psi_j(t+1) = argmax_{1 \le i \le N} \delta_i(t) a_{ij} b_{ijo_t},\ 1 \le j \le N$
- ***Terminazione e path readout:***
  $X_{T+1} = argmax_{1 \le i \le N} \delta_j(T+1)$
  $Xt = \psi_{Xt+1}(t+1)$
  $P(X) = max_{1 \le i \le N} \delta_j(T+1)$

Stima dei parametri I

- Data una certa sequenza di osservazioni, vogliamo trovare i valori dei parametri del modello $\mu=(A, B, \pi)$ che meglio spiegano quello che e' stato osservato.
- Usando la Stima di Massima Verosimiglianza, possiamo trovare i valori che massimizzano ***$P(O|\mu)$, cioe' $\text{argmax}_{\mu} P(O_{training}|\mu)$***
- Non dobbiamo calcolare direttamente tale espressione per trovare ***μ che massimizzi $P(O|\mu)$.*** In realta', possiamo massimizzarlo localmente per mezzo di un algoritmo di hill-climbing iterativo conosciuto come ***Baum-Welch*** o ***Algoritmo Forward-Backward***. (un caso speciale dell'Algoritmo di EM)

Stima dei Parametri II: Algoritmo Forward-Backward

- Non sappiamo qual'e' il modello, ma possiamo trovare la probabilita' della sequenza di osservazioni utilizzando un modello (ad esempio scelto in modo casuale).
- Osservando tali risultati possiamo ricavare quali sono le transizioni tra stati e le emissioni di simboli che sono state probabilmente utilizzate maggiormente.
- Aumentando la loro probabilita', possiamo determinare un nuovo modello che ci dia una probabilita' maggiore per la sequenza osservata.

Grammatiche Context Free Probabilistiche

- I modelli N-gram e gli HMM di Tagging ci permettono di processare una frase in modo lineare.
- Tuttavia, persino le frasi piu' semplici richiedono un modello non lineare che rifletta la struttura gerarchica delle frasi, piuttosto che l'ordine delle parole.
- Le Grammatiche Context Free Probabilistiche sono i modelli probabilistici piu' semplici e piu' naturali per le strutture ad albero e gli algoritmi per il loro addestramento sono strettamente collegati a quelli per i HMM.
- Bisonga notare, in ogni caso, che ci sono altri modi di costruire modelli probabilistici di una struttura sintattica.

Definizione formale di una PCFG

Una PCFG consta di:

- Un insieme di terminali, $\{w^k\}$, k= 1,…,V
- Un insieme di non terminali, N^i, i= 1,…, n
- Un simbolo di partenza N^1
- Un insieme di regole, $\{N^i \dashrightarrow \xi^i\}$, (dove ξ^i e' una sequenza di terminali e nonterminali)
- Un corrispondente insieme di probabilita' sulle regole tali che: $\forall i \Sigma_j P(N^i \dashrightarrow \xi^j) = 1$

- La probabilita' di una frase (in base ad una grammatica G) e' dato da:
  $P(w_{1m}, t)$ dove t e' il parse tree della frase
  $= \Sigma_{\{t: yield(t)=w1m\}} P(t)$

Assunzioni del Modello

- ***Invarianza spaziale:*** La probabilita' di un sottoalbero non dipende da dove si trovano nella stringa le parole che sono dominate da esso.
- ***Context Free:*** La probabilita' di un sottoalbero non dipende dalle parole non dominate da un sottoalbero.
- ***Ancestor Free:*** La probabilita' di un sottoalbero non dipende dai nodi della derivazione fuori dal sottoalbero.

Alcune Caratteristiche delle PCFG

- Una PCFG ci da una qualche idea della plausibilita' di differenti parse. In ogni caso, le probabilita' sono basate su fattori strutturali e non su fattori lessicali.
- PCFG sono adatte per l'induzione di grammatiche.
- PCFG sono robuste.
- PCFG danno un modello probabilistico del linguaggio.
- La potenza predittiva di una PCFG tende ad essere maggiore di un HMM. Nonostante cio' in pratica, e' peggiore.
- PCFG non sono buoni modelli da soli ma possono essere combinati con un modello tri-gram.
- PCFG hanno certi limiti che possono non essere appropriati.

Domande sulle PCFG

- Proprio come per i HMM, ci sono tre domande di base a cui vorremmo rispondere:
- Qual'e' la probabilita' di una frase w_{1m} in base ad una grammatica G: $P(w_{1m}|G)$?
- Qual'e' il parse piu' probabile per una frase: $\mathbf{argmax}\ t\ P(t|w_{1m},G)$?
- Come possiamo scegliere delle probabilita' per le regole di una grammatica G in modo che massimizzino la probabilita' di una frase, $\mathbf{argmax}_G\ P(w1m|G)$?

Restrizione

- Ci limitiamo a considerare il caso di ***Grammatiche in Forma Normale di Chomsky***, che hanno solo regole unarie e binarie della forma:
  - $N^j \rightarrow N^j\ N^k$
  - $N^j \rightarrow w^j$

1. I parametri di una PCFG nella Forma Normale di Chomsky Normal sono:
   - $P(N^j \rightarrow N^r\ N^s \mid G)$, una matrice di n3 parametri
   - $P(N^j \rightarrow w^k|G)$, nV parametri

   (dove n e' il numero di nonterminali e V e' il numero di terminali)

- $\Sigma_{r,s}\ P(N^j \rightarrow N^r\ N^s) + \Sigma_k\ P\ (N^j \rightarrow w^k) = 1$

Dai HMMs alle Probabilistic Regular Grammars (PRG)

- Una ***PRG*** ha uno stato di partenza N^1 e regole della forma:
  - $N^i \rightarrow w^j N^k$
  - $N^i \rightarrow w^j$
- Cio' e' simile a quanto avevamo per una HMM tranne per il fatto che in una HMM, abbiamo $\forall n \sum_{w1n} P(w_{1n}) = 1$ mentre in una PCFG, abbiamo $\sum_{w \in L} P(w) = 1$ dove L e' il linguaggio generato dalla grammatica.
- Le PRG sono legate ai HMM per il fatto che una PRG e' una HMM alla quale dobbiamo aggiungere uno stato' di partenza e uno stato di arrivo (o sink state).

Dalle PRG alle PCFG

- Nei HMM, eravamo in grado di fare calcoli in modo efficiente in termini di probabilita' in avanti e all'indietro.
- In un parse tree, le probabilita' in avanti corrispondono a tuttto quello che sta sotto ad un certo nodo (nodo incluso), mentre le probabilita' all'indietro corrispondono alla probabilta' di tutto quello al diffuori di un certo nodo.
- Introduciamo le probabilita' Esterne (α_j) e Interne (β_j):
  - $\alpha_j(p,q)=P(w_{1(p-1)}, N_{pq}^j, w_{(q+1)m}/G)$
  - $\beta_j(p,q)=P(w_{pq}/N_{pq}^j, G)$

Le Probabilita' di una Stringa I: Usare le Probabilita' Interne

- Usiamo l'***Algoritmo Inside***, un algoritmo fondato sulla programmazione dinamica basato sulle probabilita' interne: $P(w_{1m}/G) = P(N^1 ==>* w_{1m}/G) = P(w_{1m}/N_{1m}^1, G)=\beta_1(1,m)$
- ***Caso Base:*** $\beta_j(k,k) = P(w_k/N_{kk}^j, G)=P(N^j \rightarrow w_k/G)$
- ***Induzione:*** $\beta_j(p,q) = \sum_{r,s}\sum_{d=p}^{q-1} P(N^j \rightarrow N^r N^s) \beta_r(p,d) \beta_s(d+1,q)$

Le Probabilita' di una Stringa II: Usare le Probabilita' Esterne

- Usiamo l'Algoritmo Outside basato sulle probabilita' esterne: $P(w_{1m}/G)=\sum_j \alpha_j(k,k) P(N^j \rightarrow w_k)$
- ***Caso Base:*** $\alpha_1(1,m)= 1$; $\alpha_j(1,m)=0$ for $j \neq 1$
- ***Caso Induttivo:*** calcolo di $\alpha_j(p,q)$
- Simimente ***alle*** HMM, possiamo combinare le probabilita' interne ed esterne: $P(w_{1m}, N_{pq}/G)= \sum_j \alpha_j(p,q) \beta_j(p,q)$

Trovare il Parse piu' probabile per una frase

- L'algoritmo trova il parse tree parziale di probabilita' maggiore espandendo una certa sottostringa le cui radice e' un certo non terminale.
- $\delta_i(p,q)$ = il parse con la piu' alta probabilta' interna di un sottoalbero $N_{pq}^{\ i}$
- ***Inizializzazione:*** $\delta_i(p,p) = P(Ni \dashrightarrow w_p)$
- ***Induzione:*** $\delta_i(p,q) = \max_{1\le j,k\le n, p\le r<q} P(N^i \dashrightarrow N^j N^k)\ \delta_j(p,r)\ \delta_k(r+1,q)$
- ***Store backtrace:*** $\psi_i(p,q) = \text{argmax}_{(j,k,r)} P(N^i \dashrightarrow N^j N^k)\ \delta_j(p,r)\ \delta_k(r+1,q)$
- ***Terminazione:*** $P(t^*) = \delta_1(1,m)$

Addestrare una PCFG

- ***Restrizioni:*** Assumiamo che l'insieme di regole sia dato e cerchiamo di trovare le probabilita' ottimali da assegnare alle diverse regole grammaticali.
- Come per le HMM, usiamo un Algoritmo di Addestramento EM detto, Algoritmo ***Inside-Outside***,che ci permette di addestrare i parametri di una PCFG su frasi non annotate di un linguaggio.
- ***Assunzione Base:*** una buona grammatica e' tale da rendere la frase del corpus di addestramento probabile ==> cerchiamo la grammatica che massimizzi la verosimiglianza dei dati di addestramento.

Problemi con l'Algoritmo Inside-Outside

- Estremamente Lento: Per ogni frase, ogni iterazione dell'addestramento e' $O(m^3n^3)$.
- I Massimi Locali sono un problema molto maggiore di quanto lo fossero nei HMM
- Un apprendimento soddisfacente richiede molti piu' nonterminali di quanti non ne siano necessari teoricamente per descrivere il linguaggio.
- Non c'e' garanzia che i nonterminali appresi siano motivati linguisticamente.