ISTRUZIONI PROGETTO FASE 4

Data consegna: tre giorni (72 ore) prima della verbalizzazione

Questo documento contiene le istruzione per l'esecuzione della fase 4 del progetto. Per informazioni o domande scrivere a pennacchiotti@info.uniroma2.it.
La quarta fase deve essere eseguita insieme da tutti i membri del gruppo.
I progetti sono presentati per grado di difficoltà crescente. I gruppi possono scegliere autonomamente il progetto da eseguire. Inoltre, i gruppi potranno proporre dei temi di progetto da svolgere. Per proporre un tema e verificarne la fattibilità, inviare una e-mail per fissare un appuntamento, a pennacchiotti@info.uniroma2.it.
Il voto finale della parte progettuale (fase 1,2,3,4) terrà conto della difficoltà del progetto scelto per la fase 4, e della bontà dell'esecuzione.
Per ogni progetto sono indicate le modalità di esecuzioni e i possibili linguaggi utilizzabili.

Invio risultati

Al termine del progetto, inviare i seguenti file utilizzando la e-mail del gruppo a *pennacchiotti@info.uniroma2.it*. Per considerare terminato il progetto devono essere inviati tutti i file richiesti.

1. Breve documento (minimo 2 pagine) che indichi il progetto scelto, e descriva in breve:
    a. Metodologia applicata
    b. Algoritmo ed eventuali formule applicate
    c. Impostazione degli esperimenti
    d. Breve commento ai risultati ottenuti e possibili soluzioni alternative
2. File sorgenti implementati
3. Classi eseguibili o eseguibile funzionanti per lanciare il progetto

Indicare come oggetto della e-mail: "*Progetto parte 4: gruppo [Id gruppo], progetto [id progetto]*".

Progetti semplici

1. RICONOSCITORE MORFOLOGICO SEMPLICE

Linguaggio: qualsiasi (preferibilmente Java o Prolog)
Tool di supporto necessari: nessuno
Corpus: Wikipedia
Implementare un riconoscitore morfologico che utilizzi trasduttori a stati finiti per il parsing dei *nomi* della grammatica italiana. Il riconoscitore deve prendere in input un file contenente in ogni riga un nome, e in output stampare a video l'analisi morfologica delle parole. *Es* di output:

ragazzo → ragazz N SG MASC
ragazza → ragazz N SG FEM
ragazzi → ragazz N PL MASC
ragazze → ragazz N PL FEM

Le classi morfologiche da tenere in considerazione sono quelle indicate nell'esempio soprastante.
Il riconoscitore deve essere in grado di riconoscere i fenomeni morfologici regolari:
- Regolari plurale/singolare maschile/femminile: -o ,-a, -i, -e

Scegliere quindi 30 nomi di esempio dal corpus Wikipedia in maniera casuale (in modo da includere anche nomi irregolari). Applicare il riconoscitore a tale lista di nomi. Calcolare infine l'accuracy, intesa come numero di interpretazioni corrette prodotte dall'analizzatore sul totale delle interpretazioni. Commentare i limiti del riconoscitore implementato e le loro cause.
Attenzione: il trasduttore deve essere implementato esplicitamente! (Ad es, in Java deve essere definita una classe trasduttore composta di stati e di transizioni).

2. CORRELAZIONE DISTRIBUZIONALE SEMPLICE A FREQUENZA

Linguaggio: qualsiasi
Tool di supporto necessari: nessuno
Corpus: Wikipedia
Implementare un semplice sistema per l'estrazione di *parole correlate* da un corpus, basato sul principio della Distributional Hypothesis. Dato in input il corpus in formato testo, il sistema deve produrre in output un file contenente le coppie di parole correlate, ordinate per valori decrescenti della misura di correlazione. Il sistema deve utilizzare il seguente set-up:

- *contesto (features):* finestra delle 5 parole precedenti e delle 5 parole successive
- *misura di associazione:* frequenza
- *misura di correlazione:* coseno

Analizzare le prime 30 coppie più correlate prodotte, e misurare l'accuracy in termini di numero di parole effettivamente correlate sul numero totale di coppie (30).

Progetti di difficoltà media

3. RICONOSCITORE MORFOLOGICO COMPLESSO

Linguaggio: qualsiasi
Tool di supporto necessari: nessuno
Corpus: Wikipedia
Implementare un riconoscitore morfologico che utilizzi trasduttori a stati finiti per il parsing dei *nomi* della grammatica italiana. Il riconoscitore deve prendere in input un file contenente in ogni riga un nome, e stampare a video l'analisi morfologica delle parole. Es di output:

ragazzo → ragazz N SG MASC
ragazza → ragazz N SG FEM
ragazzi → ragazz N PL MASC
ragazze → ragazz N PL FEM

Le classi morfologiche da tenere in considerazione sono quelle indicate nell'esempio soprastante.
Il riconoscitore deve essere in grado di riconoscere i seguenti fenomeni morfologici:

- Regolari plurale/singolare maschile/femminile: -o ,-a, -i, -e
- Irregolari:
  - Nomi terminanti in *–tore* e *–dore* sono N SG MASC (output: lavoratore→lavora N SG MASC)
  - Nomi terminanti in *–tori* e *–dori* sono N PL MASC ” ” ” ” ”
  - Nomi terminanti in *–trice* e *–drice* sono N SG FEM ” ” ” ” ”
  - Nomi terminanti in *–trici* e *–drici* sono N PL FEM ” ” ” ” ”
  - Nomi terminanti in *–cia* e *–gia* sono N SG FEM (output: farmacia→farma N SG FEM)
  - Nomi terminanti in *–cie* e *–gie*sono N PL FEM ” ” ” ” ”

Scegliere quindi 50 nomi di esempio dal corpus Wikipedia in maniera casuale casualmente (in modo da includere anche nomi irregolari). Applicare il riconoscitore a tale lista di nomi. Calcolare infine l'accuracy, intesa come numero di interpretazioni corrette prodotte dall'analizzatore sul totale delle interpretazioni. Commentare i limiti del riconoscitore implementato e loro cause.
Attenzione: il trasduttore deve essere implementato esplicitamente! (Ad es, in Java deve essere definita una classe trasduttore composta di stati e di transizioni).

Facoltativo: Implementare il riconoscitore in modo da funzionare anche come generatore, ovvero, prenda in input un file contenente in ogni riga una struttura morfologica (*es.* ragazz N SG MASC) e produca in output la parola corrispondente (*es.* ragazzo)

4. PERFORMANCE MORFOLOGIA

Linguaggio: Java
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Scrivere in Java un misuratore di performance per l'analizzatore morfologico di Chaos, che prenda in input due XDG (gold standard e annotazione di Chaos) e produca in output le performance di Chaos in base al gold standard, espresse in termini di accuracy (interpretazioni morfologiche corrette prodotte da Chaos sul numero totale di interpretazioni di Chaos).
Testare il misuratore sul corpus di Wikipedia.

5. CORRELAZIONE DISTRIBUZIONALE SEMPLICE (A FREQUENZA e PMI)

Linguaggio: qualsiasi
Tool di supporto necessari: nessuno
Corpus: Wikipedia
Implementare un semplice sistema per l'estrazione di *parole correlate* dal corpus, basato sul principio della Distributional Hypothesis. Dato in input il corpus in formato testo, il sistema deve produrre in output un file contenente le coppie di parole correlate, ordinate per valori decrescenti della misura di correlazione. Il sistema deve utilizzare due set-up, in cui la misura di correlazione possa essere sia la frequenza (a) che la pointwise mutual information (b):

- *contesto (features):* finestra delle 5 parole precedenti e delle 5 parole successive
- *misura di associazione:* frequenza, PMI
- *misura di correlazione:* coseno

Analizzare le prime 30 coppie più correlate prodotte seguendo il metodo (a) ed il metodo (b), e misurare l'accuracy per i due metodi in termini di numero di parole effettivamente correlate sul numero totale di coppie. Discutere i differenti valori di accuracy ottenuti, e le loro possibili cause.

6. ESTRAZIONE DI TERMINI

Linguaggio: qualsiasi
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Estrarre dal corpus tutte le espressioni terminologiche. Dato il corpus in formato CHA, il sistema deve produrre in output i termini estratti, in ordine descrescente della loro frequenza nel corpus. I termini devono essere identificati dalle seguenti espressioni regolari:

(AR(S|P))? NC(S|P)? *(nome o articolo-nome)*
(AR(S|P))? AG(S|P)? NC(S|P)? *(aggettivo-nome o aggettivo-articolo-nome)*
(AR(S|P))? NC(S|P)? ((PSE)|(PAS)|(PAP)) NC(S|P)? *(nome-prep-nome o articolo-nome-prep-nome)*

Utilizzare una *stop-list* per scartare i termini contenenti parole comuni semanticamente non interessati della lingua italiana che non rendono il termine utile. Ad esempio "*quel libro"* non è un termine mentre "*libro di chimica*" è un termine.
Analizzare i primi 50 termini estratti, e valutare l'accuracy dell'estrattore come numero di termini corretti estratti sul numero totale di termini estratti. Si ricorda che un termine è tale se risponde alla seguente definizione:*"il termine indica un concetto importante all'interno di un dato dominio cognitivo"*. Si ricorda inoltre che il dominio del corpus Wikipedia è di tipo *alimentare*.

Progetti difficili

7. PERFORMANCE SINTASSI

Linguaggio: Java
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Scrivere in Java un misuratore di performance per l'analizzatore sintattico di Chaos, che prenda in input due XDG (gold standard e annotazione di Chaos) e produca in output le performance di Chaos in base al gold standard, espresse in termini di precision e recall sugli ICD, ed in termini di precisiom, recall, labelled precision e labelled recall sui costituenti complessi.
Testare il misuratore sul corpus di Wikipedia.

8. RELATION EXTRACTION

Linguaggio: qualsiasi
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Implementare un semplice sistema per l'estrazione della relazione *PART-OF* e *IS-A* dal corpus utilizzando una strategia basata su patterns sintattico-lessicali (utilizzare quindi i dati di POS prodotti da Chaos). Il sistema deve prendere in input un insieme di patterns sintattico-lessicali, ad esempio:

e deve restituire in output le coppie di termini legati dai patterns, ordinati per frequenza. Ad esempio per la relazione PART-OF:

dipartimento_di_informatica , università
volante,automobile

Un termine deve essere formato da una delle possibili sequenze di POS, indicate con espressioni regolari:

(AR(S|P))? NC(S|P)? *(nome o articolo-nome)*
(AR(S|P))? AG(S|P)? NC(S|P)? *(aggettivo-nome o aggettivo-articolo-nome)*

Facoltativo: riconoscere come termini anche espressioni con preposizioni (ad es. *"il cane di Mario"*)

9. CORRELAZIONE DISTRIBUZIONALE SEMPLICE (A PMI)

Linguaggio: qualsiasi
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Implementare un semplice sistema per l'estrazione di *parole correlate* dal corpus, basato sul principio della Distributional Hypothesis. Dato in input il corpus in formato testo, il sistema deve produrre in output un file contenente le coppie di parole correlate, ordinati per valori decrescenti della misura di correlazione. Il sistema deve utilizzare due set-up, in cui una *parola* va intesa (a) la forma superficiale della parola (*es.*ragazz**i**) (b) la forma base della parola (*es.*ragazzo):

- *contesto (features):* finestra delle 5 parole precedenti e delle 5 parole successive
- *misura di associazione:* PMI
- *misura di correlazione:* coseno

Analizzare le prime 30 coppie più correlate prodotte seguendo il metodo (a) ed il metodo (b), e misurare l'accuracy per i due metodi in termini di numero di parole effettivamente correlate sul

numero totale di coppie (30). Discutere i differenti valori di accuracy ottenuti, e le loro possibili cause.
Per estrarre la forma base di una parola (*lemma*) utilizzare il metodo *getFirstLemma()* della classe *Constituent* di Chaos.

10. SIMILARITA' DISTRIBUZIONALE (A FREQUENZA e PMI)

Linguaggio: qualsiasi
Tool di supporto necessari: Chaos
Corpus: Wikipedia
Implementare un semplice sistema per l'estrazione di *parole simili* dal corpus, basato sul principio della Distributional Hypothesis. Il sistema deve riconoscere i *verbi* simili oppure i *nomi* simili (scelta libera lasciata agli studenti).
Dato in input il corpus in formato CHA, il sistema deve produrre in output un file contenente le coppie di parole simili, ordinate per valori decrescenti della misura di similarità. Il sistema deve utilizzare due set-up, in cui la misura di similarità possa essere sia la frequenza (a) che la pointwise mutual information (b):

- *contesto (features):*
  - *per i verbi:* relazioni V-Sog, V-Ogg, V-NP istanziate sui nomi del corpus
  - *per i nomi:* relazioni Sog-V, Ogg-V, NP-NP, NP-PP istanziate sui verbi e i nomi del corpus
- *misura di associazione:* frequenza, PMI
- *misura di correlazione:* coseno

Analizzare le prime 30 coppie più simili prodotte seguendo il metodo (a) ed il metodo (b), e misurare l'accuracy per i due metodi in termini di numero di parole effettivamente correlate sul numero totale di coppie (30). Discutere i differenti valori di accuracy ottenuti, e le loro possibili cause.