UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA
Facoltà di Ingegneria “Enzo Ferrari” di Modena

Corso di Laurea Magistrale in Ingegneria Informatica (270/04)

Individuazione automatica non supervisionata del significato di parole tramite risorse testuali multi-lingua

Relatore:
Prof. Sonia Bergamaschi

Correlatore:
Prof. Rada Mihalcea

Candidato:
Lorenzo Albano

Anno Accademico 2012/2013

Ai miei genitori per i sacrifici fatti, e a Serena
per avermi accompagnato e sostenuto attraverso
questo cammino.

Indice

Introduzione

E' stato dimostrato che l'utilizzo di *word senses*[1] al posto delle semplici *words*[2] influisce positivamente sulle prestazioni delle attività di Natural Language Processing, quali ad esempio *information extraction* [8], *information retrieval* [40], *machine translation* [42] e *data integration* [36][30]. Il lavoro di determinare computazionalmente il corretto senso di una parola all'interno di un contesto è uno degli argomenti di ricerca cardine nella *Linguistica Computazionale* e nel *Natural Language Processing*. La ragione della sua importanza è da ricercarsi nell'ambiguità propria del linguaggio naturale, il quale è composto da un gran numero di parole aventi significati multipli dipendenti dal contesto. Si considerino ad esempio le seguenti frasi:

- (a) The *plant* is dried (La pianta è appassita)
- (b) There was an industrial *plant* here (C'era un impianto industriale qui)

E' evidente che l'occorrenza della parola *plant* nelle due frasi denota differenti significati: rispettivamente un organismo vivente ed un luogo in cui avviene un processo industriale. Sfortunatamente identificare il senso specifico che una parola assume all'interno di un contesto è solo apparentemente un compito facile, soprattutto per una macchina. Infatti mentre un essere umano, il più delle volte, riesce ad indurre facilmente il senso di una parola ambigua, una macchina ha bisogno di processare dati testuali non strutturati e di trasformarli in strutture dati che devono essere analizzate allo scopo di rilevare il significato sottostante. Il processo automatico di identificazione del senso delle parole ambigue è chiamato **Word Sense Disambiguation**.

[1] Significati delle parole
[2] Parole

Per poter fare della WSD occorre la disponibilità di *Sense Inventory* come il database WordNet [23]. Un *Sense Inventory* somiglia molto ad un dizionario tradizionale: raggruppa parole in insiemi di sinonimi detti *synsets* e fornisce definizioni brevi e generali per ciascun gruppo. Il DBGroup (Database Group) dell'Università di Modena e Reggio Emilia ha condotto un'intensa attività di ricerca nel campo della Disambiguazione tramite l'uso di *Sense Inventory*, in particolare WordNet [4] [6] [35] [37].

Comunque, queste sorgenti dati tendenzialmente soffrono di alcune limitazioni:

- copertura limitata ad un dominio specifico;
- eccessiva granularità nel discernere i significati;
- tendenza a rappresentare solo significati lessicografici e ad ignorare nomi propri;
- alto costo di mantenimento ed aggiornamento dei dati.

Per questo sono state proposte metodologie alternative per il discernimento dei significati delle parole che fanno uso di testo non annotato, note come tecniche di *unsupervised Word Sense Induction*. Nella WSI il problema dell'identificazione dei sensi multipli di un termine ambiguo viene concettualizzato come un problema di *clustering*, dove ciascun cluster rappresenta un significato distinto della parola.

In questo lavoro di tesi viene proposta una variante alle esistenti tecniche di WSI che fa uso di una rappresentazione multi-lingua del testo al fine di sfruttare l'informazione aggiuntiva insita nel processo di traduzione (automatico o manuale che sia). Questo tipo di rappresentazione porta un duplice vantaggio: in primo luogo, durante il processo di traduzione vi è un tentativo di disambiguazione della parola attraverso l'assegnazione di una traduzione diversa a seconda del contesto in cui tale termine compare; in secondo luogo abbiamo un arricchimento del numero di *features* utilizzabili per l'individuazione del significato corretto della parola, dal momento che possiamo attingere a più lingue contemporaneamente.

Una parte del lavoro di Tesi è stato svolto presso il laboratorio LIT (Language & Information Technologies) della University of North Texas di Denton, Texas.

Per l'esecuzione degli esperimenti è stato utilizzato il software **SenseClusters** di Ted Pedersen [32]. SenseClusters è un sistema di *Word Sense Discrimination*

disponibile gratuitamente che utilizza un approccio al *clustering* completamente non supervisionato: non utilizza nessuna conoscenza aggiuntiva rispetto a quella disponibile in un *corpus* non strutturato composto da semplice *plain text* e raggruppa le istanze di una data *target word*[3] basandosi solo sulla loro mutua similarità contestuale (*context clustering*). E' un sistema completo che fornisce supporto per la selezione di *features* da ampi *corpora*, differenti schemi di rappresentazione del contesto, vari algoritmi di *clustering* e la possibilità di valutare i *clusters* individuati.

Un aspetto cruciale ha riguardato la scelta del *corpus* da utilizzare. Si è scelto di utilizzare JRC-Acquis, un *corpus* di tipo parallelo sviluppato dal Joint Research Center di Ispra, in Italia che fornisce più di 8000 documenti tratti da testi di leggi dell'Unione Europea in più di 20 lingue.

Per l'esecuzione degli esperimenti, l'adattamento dei dati, la raccolta e l'analisi dei risultati sono stati realizzati degli script in linguaggio *Perl*, linguaggio molto adatto all'elaborazione di dati di tipo testuale.

La valutazione dei risultati è stata svolta su istanze di test generate casualmente dallo stesso software *SenseClusters* e manualmente annotate con il senso corretto (*supervised evaluation*), selezionato da un insieme *coarse-grained*[4] di significati; da questo tipo di valutazione si è osservato come l'utilizzo di *features* multi-lingua all'interno di algoritmi di Word Sense Induction, porti tendenzialmente ad un incremento delle prestazioni rispetto all'utilizzo di testo mono-lingua.

La tesi risulta quindi strutturata in questo modo:

- **Capitolo 1: Stato dell'arte**: Si presenta una breve panoramica sulla WSD e sulle tecniche esistenti di WSI;

- **Capitolo 2: Clustering**: Viene fatta una presentazione delle tecniche di *clustering* e delle misure di valutazione della bontà degli algoritmi di clustering;

- **Capitolo 3: Il software SenseClusters**: Viene analizzato il software SenseClusters e gli strumenti da esso offerti per l'esecuzione del *word clustering*.

[3] Termine di cui si vuole eseguire la disambiguazione
[4] A grana grossa, cioè generali

- **Capitolo 4: Il corpus Multilingua JRC-Acquis**: Si descrive il corpus JRC-Acquis e se ne analizzano punti di forza e punti deboli;
- **Capitolo 5: Multilingual Word Sense Induction**: Si analizzano le impostazioni sperimentali, si descrivono i test eseguiti e vengono presentati i risultati ottenuti.

Capitolo 1

Stato dell'arte

1.1 Word Sense Disambiguation

La *Word Sense Disambiguation* è un problema aperto nel campo del *Natural Language Processing* incentrato sul processo di identificazione del senso assunto da una parola *polisemica*[1] in una frase.

Formalmente dato un testo T, definito come una sequenza di parole (W_1, W_2, ..., W_n), possiamo descrivere la WSD come l'operazione di assegnare il senso appropriato a tutte o ad alcune delle $W_i \in T$, cioè di identificare un mapping A dalle parole ai sensi in modo tale che $A(i) \subseteq Senses_D(W_i)$, dove $Senses_D(W_i)$ è l'insieme dei sensi presenti in un dizionario D per la parola W_i e $A(i)$ è quel sottoinsieme di sensi dei W_i che risultano appropriati all'interno del contesto T. Il mapping A può assegnare più di un significato a ciascuna parola, sebbene tipicamente viene selezionato solo il significato più appropriato, cioè $|A(i)| = 1$.

E' stata definita come un *AI-complete problem*, cioè un problema la cui difficoltà è equivalente alla complessità nel risolvere problemi centrali dell'*Intelligenza Artificiale*, come ad esempio il Test di Turing.

La sua conclamata difficoltà dipende da numerosi fattori; in primo luogo un *WSD task* si presta a differenti formalizzazioni per quanto riguarda problemi fondamentali, come l'approccio alla rappresentazione di un *word sense*[2] (che varia dalla semplice enumerazione di un insieme finito di sensi alla generazione *rule-*

[1] Che può assumere significati differenti
[2] Significato associato ad una parola

based di nuovi significati), la granularità di un *sense inventory*[3], la natura del testo (che può essere *domain-specific* o generale), l'insieme di *target words* da disambiguare, etc. In secondo luogo la WSD si basa fortemente sulla *Conoscenza*. Le sorgenti di Conoscenza variano considerabilmente da *corpora*[4] non annotati o annotati con word senses, fino a risorse strutturate come ad esempio dizionari *machine-readable*, reti semantiche, etc.

La WSD può essere vista come un attività di classificazione: i *word senses* rappresentano le classi e un metodo automatico di classificazione è utilizzato per assegnare ciascuna occorrenza di una *word* a una o più classi, basandosi sull'*evidenza* derivante dal *contesto* e da sorgenti di conoscenza esterne. A differenza dei problemi di classificazione tipici, che usano un insieme di classi predefinite, nella WSD l'insieme delle classi cambia per ciascuna parola da disambiguare. Esistono due varianti del problema:

- *Targeted WSD*: viene eseguita la disambiguazione su un ristretto insieme di *target words*[5], solitamente una per frase;
- *All Words WSD*: al sistema è richiesto di disambiguare tutte le parole all'interno di un testo (nomi, verbi, aggettivi e avverbi).

1.1.1 Sense Representation

Un *word sense* è un significato comunemente accettato di una parola. La determinazione del *sense inventory* da adottare è un problema chiave della Word Sense Disambiguation; si considerino le seguenti frasi:

- (c) I can hear *bass* sounds;
- (d) They like grilled *bass*.

La parola *bass* viene utilizzata nelle frasi precedenti con due significati differenti: toni a bassa frequenza (c) e tipo di pesce (d). Le due parole sono legate da una relazione di **omonimia**, dal momento che vengono scritte allo stesso modo ma assumono significati completamente differenti tra loro. Inoltre è possibile fare

[3] Vocabolario dei sensi
[4] Collezione di testi selezionati e organizzati per facilitare analisi di tipo linguistico
[5] Parole da disambiguare

una distinzione molto più sottile dei significati di una parola; ad esempio la parola *bass*, nella sua accezione musicale potrebbe riferirsi alla parte inferiore della scala musicale come anche alla parte più grave di un brano musicale polifonico. Questo tipo di ambiguità è detta **polisemia**. Sfortunatamente sensi di questo tipo possono essere creati ad ogni livello di granularità e portare di conseguenza ad una inventario di significati eccessivamente *fine-grained*[6], il che può rappresentare un problema per quelle applicazioni che non richiedono un livello elevato di granularità. Sono stati sviluppati due diversi tipi di approccio al problema della granularità mirati alla creazione di una distinzione di tipo *coarse-grained*[7] all'interno di WordNet:

- *Manuale*: utilizzato dal progetto OntoNotes [14], mira alla creazione di una distinzione dei sensi sottomettendo iterativamente nuove partizioni dei sensi di una data parola a degli annotatori umani finchè il 90% degli annotatori non concorda su un partizionamento;
- *Automatico*: effettua il raggruppamento di sensi sematicamente simili utilizzando tecniche di WSD. Il *clustering* viene ottenuto mappando automaticamente i sensi di WordNet ad un dizionario di riferimento di tipo *machine-readable* con una distinzione dei sensi di tipo gerarchica [24].

1.1.2 Tecniche

Esistono tre approcci alla Word Sense Disambiguation:

- **Supervised WSD**: vengono utilizzati metodi di *machine learning* per apprendere un classificatore per la *target word* da un *training set* annotato, cioè un insieme di esempi codificati come vettori i cui elementi rappresentano le *features*, con una speciale etichetta che rappresenta la classe appropriata (senso);
- **Knowledge-based WSD**: sfruttano risorse come dizionari, thesauri e ontologie per determinare il senso di una parola in un contesto. I migliori sistemi basati su questa tipologia di WSD sfruttano WordNet o altre risorse per costruire uno schema semantico e sfruttano le proprietà strutturali di tale grafico al fine di scegliere il significato appropriato;

[6] A grana fine, cioè una distinzione molto dettagliata
[7] Distinzione poco dettagliata

- **Unsupervised WSD**: detta anche **Word Sense Induction** è mirata all'individuazione automatica dei sensi senza alcun utilizzo di sorgenti di dati manualmente annotate, ma sfruttando solamente *corpora* non annotati (si veda la prossima sezione).

1.1.3 Conoscenza

La conoscenza è una componente fondamentale nella *Word Sense Disambiguation* ed è uno dei fattori fortemente collegati alle performance: è stato dimostrato che maggiore è la quantità di conoscenza di alta qualità, più alte sono le prestazioni degli algoritmi di WSD. Nello specifico:

- mella *supervised WSD* le performance possono essere aumentate considerabilmente quando sono disponibili centinaia di esempi di *training* per ciascuna *word* [22];
- è stato mostrato che la *knowledge-based WSD* beneficia fortemente della presenza di più di 100 relazioni semantiche per ciascun *word sense*, raggiungendo fino al 90% di *accuracy*[8] [25].

Possiamo suddividere le sorgenti di conoscenza in due grandi categorie: risorse **strutturate** e risorse **non strutturate**.

Tra le risorse di tipo **strutturato** troviamo:

- *Thesauri*: forniscono anche informazioni circa le relazioni tra le *words* come la *sinonimia*[9], l'*antonimia*[10] e possibili ulteriori relazioni;
- *Machine Readable Dictionaries* (MRDs): sono diventati una sorgente popolare i conoscenza per il Natural Language Processing dagli anni '80, quando sono stati creati i primi dizionari elettronici. WordNet solitamente viene considerato un passo avanti ai semplici MRDs, dal momento che include una ricca rete semantica di concetti;
- *Ontologie*: sono una rappresentazione formale, condivisa ed esplicita di una concettualizzazione di un dominio di interesse, che normalmente include

[8] Misura di performance tipicamente utilizzata nel campo dell'*Information Retrieval*
[9] Relazione tra due termini diversi che hanno lo stesso significato
[10] Relazione tra due termini che hanno significato opposto

una tassonomia[11] e un insieme di relazioni semantiche. WordNet può essere considerato un'ontologia.

Tra le risorse di tipo **non strutturato** troviamo:

- *Corpora*: collezione di testi che possono essere *sense-tagged* oppure in formato *raw*.
- *Collocation Resources*: registrano la tendenza di alcune parole ad occorrere regolarmente con altre.
- *Stoplists*: liste di parole che non sono utili ai fini della WSD.

1.1.4 Multilinguismo

Attualmente la maggior parte della ricerca nella Word Sense Disambiguation è condotta nella lingua Inglese perchè è la lingua per cui vi è la più grande disponibilità di risorse. Tuttavia, recentemente c'è stato un crescente interesse nei confronti delle altre lingue. Soprattutto, l'attenzione viene sempre più rivolta all'esecuzione di WSD tra lingue, un task definito come **cross-lingual WSD**, in cui viene fornita in input una frase in una lingua sorgente e il sistema di WSD deve fornire in output i *word senses* codificati in una lingua *target*. Il *sense inventory* è ottenuto utilizzando delle traduzioni raccolte da un *parallel text*[12] e non da *sense labels* predefiniti. L'assunzione alla base - supportata da diversi studi [15, 26] - è che le distinzioni dei sensi di una parola in una frase in un linguaggio sorgente sono determinate dalle differenti traduzioni della parola in altre lingue. Questo approccio offre numerosi benefici in confronto alla tradizionale WSD mono-lingua: può essere applicato a qualsiasi lingua di interesse, può essere facilmente integrato in applicazioni reali (es. *machine translation*) e affronta il problema della granularità dei sensi. Tuttavia è richiesto un corpus bi-lingua che abbia un'ampia copertura, un requisito non facile da soddisfare.

Una sfida molto impegnativa per la ricerca è quella che riguarda la reale **multilingual WSD**, nella quale i sensi vengono ritornati lessicalizzati in più lingue.

[11] Classificazione degli elementi di una lingua in liste atte a evidenziare le regole di combinazione dei termini

[12] Testo con accanto la sua traduzione

Sono stati sviluppati anche dei metodi che sfruttano *parallel texts* per arricchire le *features* disponibili da utilizzare per la disambiguazione, sia di tipo *supervised* che sfruttano un sistema di votazione composto da classificatori di Bayes allenati su diverse combinazioni di più lingue diverse [2], sia di tipo *unsupervised* che utilizzano l'algoritmo di Lesk [18] su testo multi-lingua [27].

L'approccio multi-lingua offre importanti spunti applicativi e può risultare fondamentale soprattutto nel campo dell'integrazione dati. E' comune infatti la necessità di dover integrare sorgenti dati bi-lingue, soprattutto per i paesi europei non di lingua inglese, in cui spesso ci si trova ad operare su dati sia in lingua nazionale che in lingua inglese. Si pensi al progetto europeo SEWASIE (Semantic Webs and AgentS in Integrated Economies) [7] dove si è utilizzato il sistema di *data integration* MOMIS [5] per la costruzione di un'ontologia di aziende nel settore meccanico e tessile utilizzando sorgenti dati in italiano e in inglese [3].

1.1.5 Performance

Un noto problema della Word Sense Disambiguation riguarda le performance. Nei test di tipo *all-words* l'*accuracy* dello stato dell'arte si aggira intorno al 65% ed è stato confermato che uno dei maggiori ostacoli alle alte performance è dato dalla rappresentazione dei sensi. Negli anni recenti sono stati fatti dei progressi che hanno portato ad un significativo incremento dell'accuracy, dal 65% all'82-83% [9, 31] in contesto di tipo *all-words* e quando la distinzione tra i *word senses* è di tipo *coarse-grained*.

1.2 Word Sense Induction

Dati i limiti principali della Word Sense Disambiguation, quali l'eccessiva granularità dei sensi e l'alto costo di gestione e aggiornamento dei dizionari (*knowledge acquisition bottleneck*), la Word Sense Induction appare come un'attraente alternativa. Questo nuovo approccio è basato sulla **distributional hypothesis**, cioè l'idea che una data parola, usata con un significato specifico, tenda a co-occorrere con lo stesso sottoinsieme di parole vicine [13]. Con la WSI si è in grado di indurre *word senses* dal testo attraverso il *clustering* delle occorrenze delle parole, senza alcun bisogno di avere a disposizione del testo annotato o risorse *machine*

readable come dizionari, thesauri o ontologie. Si noti comunque che lo scopo della WSI è differente da quello della *supervised WSD*, dal momento che si mira ad identificare dei *sense clusters*[13], piuttosto che assegnare dei *sense labels*.

1.2.1 Approcci alla WSI

I principali approcci al problema della *Word Sense Induction* proposti in letteratura sono:

- **Word clustering**: in cui si fa il clustering delle parole che sono semanticamente simili;
- **Context clustering**: dove l'ipotesi di fondo è che il profilo distribuzionale delle parole esprime implicitamente la loro semantica;
- **Co-occurrence Graphs**: si costruiscono e analizzano grafici di co-occorrenze per identificare l'insieme di sensi di un dato termine.

Word Clustering

Vengono considerate le parole come *feature vector*, utilizzando una funzione di similarità per poter eseguire il clustering delle parole. Un noto algoritmo di *word clustering* [19] utilizza come misura di similartità tra due *words* il contenuto di mutua informazione tra le loro singole *features*, date da dalle dipendenze sintattiche che occorrono nel corpus (come ad esempio soggetto-verbo, verbo-oggetto, aggettivo-nome, etc.): più dipendenze le due parole condividono, maggiore è il contenuto di informazione. Per la distinzione tra i sensi viene applicato un algoritmo di *clustering*. Sia W la lista delle parole simili ordinata secondo il grado di similarità con w_0; un albero di similarità T, inizialmente composto dal solo nodo w_0, viene creato. Successivamente per ogni $i \in \{1, ..., k\}$, $w_i \in W$ viene aggiunto come figlio di w_j nell'albero T, dove w_j è la parola più simile a w_i tra le parole in $\{w_0, ..., w_{i-1}\}$. Dopo un passo di *potatura* ciascun ramo di T è considerato un senso distinto di w_0.

Successivamente viene proposto un diverso approccio al problema, chiamato *clustering by committee* (CBC) [20], che può essere descritto dai seguenti passi:

[13] Gruppi di termini con lo stesso significato

- viene ancora calcolato l'insieme di parole simili come sopra, utilizzando la medesima misura di similarità e viene costruita una matrice di similarità;

- dato un insieme di *words* E, viene applicata una procedura ricorsiva per determinare insiemi di *clusters* - detti *committees* - delle parole in E. A questo scopo viene utilizzato un algoritmo di *clustering* di tipo *average link*. In ciascuno step, le *word* residue (cioè quelle non simili a sufficienza con nessuno dei centroidi delle *committees*) vengono identificate e si procede, tramite dei tentativi ricorsivi, al tentativo di identificazione di nuove *committees* da esse;

- infine nel passo di *sense discrimination* si assegna ciascuna *target word* $w \in E$ al cluster più simile, basandosi sulla similarità del suo *feature vector* con il centroide di ciascuna *committee*;

- Dopo che una parola w è stata assegnata a una *committee* c, le *features* in comune tra w e gli elementi in c vengono rimosse dalla rappresentazione di w, in modo da rendere possibile, nella successiva iterazione, l'identificazione di sensi poco frequenti della stessa parola.

Context Clustering

Ciascuna occorrenza di una *target word* in un corpus viene rappresentata come un *context vector*. I vettori sono in seguito clusterizzati in gruppi, ciascuno dei quali identifica un senso della *target word*. Un approccio storico di questo tipo è basato sull'idea di *word space* [34], cioè un spazio vettoriale le cui dimensioni sono *words*. Il vettore per la parola w è derivato dai vicini di w in un corpus, cioè quelle parole che co-occorrono con w in una frase. Una parola w in un corpus viene quindi rappresentata come un vettore la cui j-esima componente conta il numero di volte che la parola w_j co-occorre con w all'interno di un dato *contesto* (una frase o gruppo di frasi). L'ipotesi sottostante questo modello è che il profilo distribuzionale delle parole esprime implicitamente la loro semantica.

La figura 1.1(a) mostra due esempi di *word vectors* in uno spazio bi-dimensionale, *restaurant* = (210, 80) e *money* = (100, 250), dove la prima dimensione rappresenta il numero di co-occorrenze con la parola *food* e la seconda il numero di co-occorrenze con la parola *bank*.

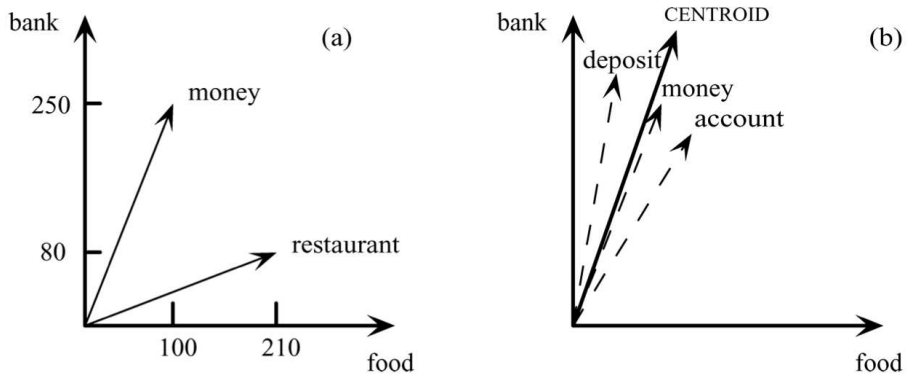

Figura 1.1: (a) Un esempio di due word vectors restaurant =(210, 80) e money = (100, 250). (b) Un context vector per stock, calcolato come il centroide dei vettori delle parole che occorrono nello stesso contesto

La *similarità* tra due parole può essere misurata geometricamente attraverso la *cosine similarity* tra i corrispondenti vettori **v** e **w**:

$$sim(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{w} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{m} v_i w_i}{\sqrt{\sum_{i=1}^{m} v_i^2 \sum_{i=1}^{m} w_i^2}} \tag{1.1}$$

dove m è il numero di *features* in ciascun vettore. I vettori di ciascuna *word* nel contesto vengono messi insieme per formare la **matrice di co-occorrenza**. A questo punto si costruisce il *context vector*, cioè il vettore che rappresenta il contesto di una specifica occorrenza di una *target word*; esso viene calcolato come il *centroide*[14] dei vettori delle *words* che occorrono nel *target context*. Un esempio di *context vector* è mostrato nella Figura 1.1(b). Una volta ottenuto questo tipo di rappresentazione si procede alla *sense discrimination* raggruppando i *context vectors* tramite un algoritmo di *clustering*.

Un problema nella costruzione di *context vectors* è dato dall'enorme quantità di dati di *training* che sono richiesti per poter ottenere una rappresentazione significativa delle co-occorrenze. Questo problema può essere limitato estendendo il *feature vector* di ciascuna parola con le parole presenti nella glossa del suo significato (sebbene questo renda l'approccio di tipo semi-supervisionato).

[14]Media normalizzata

Co-occurrence Graphs

Mentre gli approcci discussi finora lavorano principalmente assegnando una singola parola ad un cluster, alcuni recenti approcci alla *sense induction* riformulano lo spazio del problema sotto forma di grafo, operando con i collegamenti tra le parole piuttosto che con le parole isolate. Queste tecniche continuano ad usare gli algoritmi di *clustering*, ma gli elementi da raggruppare e le *features* considerate sono definite nello spazio dei grafi invece che nello spazio dei *feature vector*.

Formalmente lo spazio del problema viene rappresentato come un *hypergraph*[15] $H = (V, F)$, dove V è un insieme di vertici e F è un insieme di *hyperedges*, ciascuno dei quali include $n \geq 1$ vertici; nel caso della *word sense induction* ciascun vertice rappresenta una *word* e ciascun *hyperedge* rappresenta un insieme di parole correlate sintatticamente che co-occorrono in una frase o in un più largo contesto.

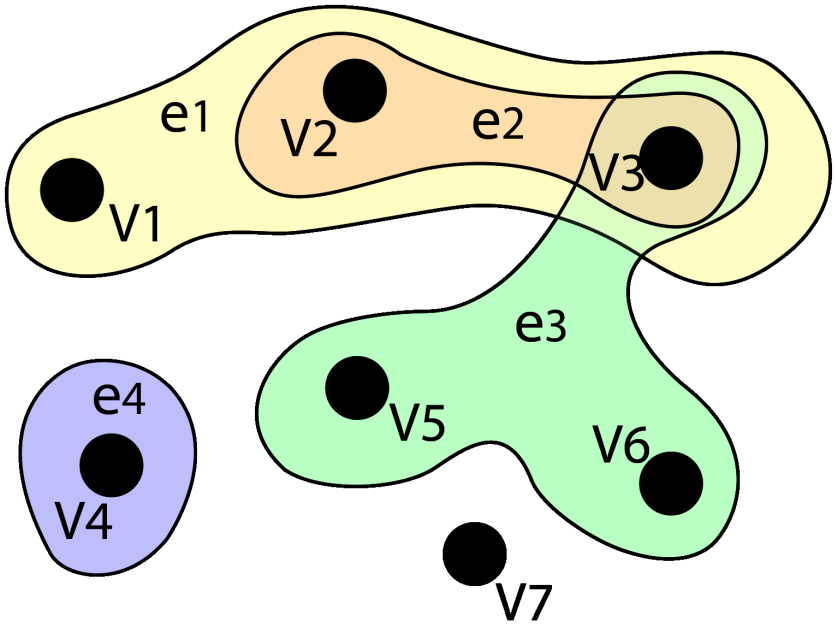

Figura 1.2: Esempio di un *hypergraph model*

Il processo di costruzione di un ipergrafo per una target word w e un corpus p è il seguente:

[15] ipergrafo

- w è rimossa da p:
- Ciascun paragrafo p_i viene annotato con *tag* di tipo *part-of-speech* e solo i nomi vengono tenuti;
- I nomi vengono filtrati considerando la minima frequenza dei nomi (parametro p_1) e i p vengono filtrati secondo una dimensione minima del paragrafo (parametro p_2);
- I nomi correlati vengono raggruppati in *hyperedges* e mantenuti se il loro *support* eccede un certo parametro p_3

$$support(f) = \frac{freq(a,b,c)}{n} \tag{1.2}$$

  dove f è un possibile *hyperedge*, a, b e c sono i suoi vertici e $freq(a,b,c)$ è il numero di paragrafi in p che contengono i vertici a b e c. Il denominatore n è la dimensione totale di p;
- Ciascun *hyperedge* f ha un peso assegnato che rappresenta la media di m confidenze, dove m è la dimensione di f e la $confidence$ di $r_0 = \{a,b\} => c$ è data da

$$confidence(r_0) = \frac{freq(a,b,c)}{freq(a,b)} \tag{1.3}$$

- Gli *hyperedges* con peso sotto una data soglia p_4 vengono rimossi;
- L'ipergrafico rimanente è ridotto in modo da rispettare la definizione di H rimuovendo gli *hyperedges* che coprono più di 4 *words*.

L'estrazione dei *word senses* viene eseguita utilizzando una versione modificata dell'algoritmo *HyperLex* [16] il quale identifica iterativamente gli hub[16] principali nell'*hypergraph*; in ciascuna iterazione il vertice v_i avente il più alto grado viene selezionato in modo che rispetti il criterio del minimo numero di *hyperedges* (parametro p_5) e il peso medio dei primi p_5 *hyperedges* (parametro p_6). Se queste condizioni sono rispettate, gli *hyperedges* contenenti v_i vengono raggruppati in un cluster c_j con distanza 0 da v_i e rimossi dall iper-grafo. Il cluster ottenuto rappresenta un singolo *word sense*.

[16] Vertici con il più alto grado

Nel momento in cui non vi sono più vertici che rispettano le condizioni di cui sopra, ciascun *hyperedge* viene assegnato al *cluster* più vicino basandosi sulla distanza media da tutti i membri del cluster (*Group Average Link*). Il peso assegnato a tali *hyperedges* è inversamente proporzionale alla distanza dai *clusters* a cui vengono assegnati.

1.2.2 Valutazione

Uno dei problemi chiave nella *Word Sense Induction* è quello della valutazione. La WSI è una specifica istanza del problema del *clustering*, quindi è tanto difficile da valutare quanto qualsiasi algoritmo di *clustering*; sfortunatamente valutare l'output di un algoritmo di *clustering* è un compito arduo anche per un essere umano. La difficoltà principale risiede nel fatto che non esiste un singolo output dell'algoritmo di *clustering* che venga accettato da tutti gli annotatori umani, cioè non esiste un *gold-standard* ben definito.

Ciò nonostante sono stati proposti in letteratura differenti approcci per stabilire la qualità di un algoritmo di *Word Sense Induction*:

- **Valutazione supervisionata**: l'output della WSI viene valutato in un task di WSD. Per fare questo i sensi indotti automaticamente vengono mappati ai *gold-standard senses* per la *target word*. Ciascuna frase viene annotata con il *gold-standard sense* e vengono usate delle misure di valutazione tipiche dell'*information retrieval* per determinare la qualità della WSD risultante. Sia $T = (w_1, ..., w_n)$ un test set e A una funzione di risposta che associa a ciascuna parola $w_i \in T$ il senso appropriato da un dizionario D, allora data l'associazione dei sensi $A'(i) \in Sense_D(w_i) \cup \{\epsilon\}$ fornita da un sistema automatico di WSD, possiamo definire la *coverage* C come la percentuale di elementi nel *test set* per cui il sistema fornisce un assegnamento, cioè

$$C = \frac{|i \in \{1, ..., n\} : A'(i) \neq \epsilon|}{n}, \tag{1.4}$$

  dove indichiamo con ϵ il caso in cui il sistema non fornisce alcuna risposta per una specifica parola w_i e con n il numero totale di risposte. La *Precision* P di un sistema è definita come la percentuale di risposte corrette date dal sistema, cioè

$$P = \frac{|i \in \{1, ..., n\} : A'(i) \in A(i)|}{|i \in \{1, ..., n\} : A'(i) \neq \epsilon|}. \tag{1.5}$$

La *Recall* R (detta anche *accuracy* è definita come il numero di risposte corrette date dal sistema, sul totale delle risposte da dare:

$$R = \frac{|i \in \{1, ..., n\} : A'(i) \in A(i)|}{n}. \tag{1.6}$$

Secondo le definizioni precedenti abbiamo che $R \leq P$ e quando la Coverage è del 100% $P = R$. Infine una misura che considera sia *Precision* che *Recall*, facendone la media ponderata armonica è la *F-measure*, definita come:

$$F_1 = \frac{2PR}{P + R}. \tag{1.7}$$

- **Valutazione non-supervisionata**: i *clusters* delle frasi corrispondenti ai sensi indotti vengono valutati in confronto a *clusters gold-standard*, cioè dei cluster generati manualmente da un sottoinsieme di dati rappresentativo dell'intero *dataset*. La qualità della soluzione viene determinata calcolando la similarità tra il *target clustering* e il *gold-standard clustering*, determinata da misure quali la V-Measure [33], il paired F-Score [21], etc. Un metodo alternativo di tipo non supervisionato è quello che ricorre all'uso delle **pseudowords** [34], cioè delle parole ambigue create artificialmente:

  - le parole da disambiguare vengono combinate casualmente in insiemi di due termini;
  - per ciascuna coppia di termini, vengono selezionate dal *testing set* tutte i contesti in cui occorre ciascuna delle due parole;
  - le due parole (ad esempio *banana* e *door*) vengono fuse in un termine composto (es. *banana/door*), che viene usato per rimpiazzare tutte le occorrenze (nei contesti selezionati al passo precedente) di entrambe le parole che lo compongono;
  - in questo modo viene introdotta artificialmente un'ambiguità lessicale che deve essere risolta;
  - a questo scopo si esegue il processo disambiguazione, che porta alla selezione di una delle componenti della parola composita;

  - a questo punto risulta semplice valutare le performance dell'algoritmo di disambiguazione dal momento che basta tornare al testo originale e decidere se per ciascun termine ambiguo è stata presa la decisione corretta.

- **Valutazione all'interno di un'applicazione**: un ulteriore modo di valutare la WSI è quella di utilizzarla all'interno di applicazioni di *Machine Translation* oppure *Web Search* e verificare se le tecniche di WSI riescono a superare le prestazioni dei sistemi non semantici.

Capitolo 2

Clustering

Il *Clustering* o analisi dei gruppi è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e raggruppamento di elementi omogenei in un insieme di dati in maniera non supervisionata. Viene anche chiamato *sorting* dagli psicologi o *segmentation* nel campo del marketing.

Le tecniche di *clustering* si basano su misure relative alla somiglianza tra gli elementi, i quali vengono raggruppati sulla base della loro distanza reciproca, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme stesso. Più rigorosamente, i dati vengono organizzati in classi in modo che vi siano:

- alta similarità intra-classe[1],
- bassa similarità inter-classe[2].

In molti approcci (*distance-based clustering*) la similarità, o meglio, dissimilarità, è concepita in termini di distanza in uno spazio multidimensionale. Un esempio può essere osservato nella figura 2.1 dove viene mostrato l'identificazione di 4 possibili *clusters* a partire da un insieme di dati rappresentati in uno spazio euclideo; il criterio di similarità utilizzato è la distanza: due o più oggetti appartengono allo stesso gruppo se sono vicini secondo una data distanza di tipo geometrico. Esistono comunque altre tipologie di *clustering* dette di *conceptual clustering* dove due o più oggetti sono considerati appartenenti allo stesso *cluster* se quest'ultimo definisce un concetto ad essi comune. In altre parole, gli oggetti

[1]Similarità tra gli elementi dello stesso gruppo
[2]Similarità tra gli elementi appartenenti a gruppi diversi

vengono raggruppati in accordo alla loro capacità di rappresentare determinati concetti descrittivi, invece che ad una semplice misura di similarità.

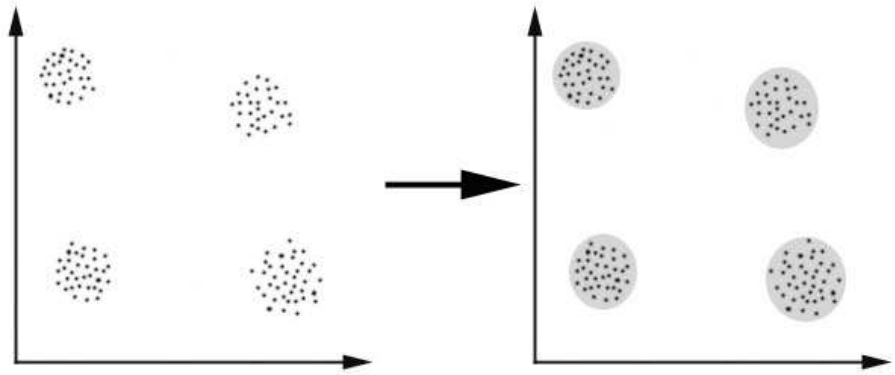

Figura 2.1: Esempio di clustering

2.1 Similarità e Distanza

Tutte le tecniche di *clustering* sono basate su un concetti di *distanza* e similarità per la costruzione di *clusters* omogenei. In seguito vengono presentate in breve le misure più utilizzate.

2.1.1 Misure di distanza

Misurano il livello di diversità tra due punti e variano tra 0 e $+\infty$. Dati due punti x e y una funzione di *distanza* D deve soddisfare le seguenti proprietà:

- $D(x, y) \geq 0$
- $D(x, y) = 0 \iff x \equiv y$
- $D(x, y) = D(y, x)$ (*simmetria*)
- $D(x, y) \leq D(x, z) + D(z, y)$ (*disuguaglianza triangolare*).

Distanza di Minkowski

E' una forma di distanza generale definita come

$$D_p(x,y) = (\sum_{k=1}^{d}(x_k - y_k)^p)^{1/p}. \tag{2.1}$$

Da essa derivano alcune distanze comunemente utilizzate come:

- **Distanza di Manhattan**: ottenuta ponendo $p = 1$

$$D_1(x,y) = \sum_{k=1}^{n}|x_k - y_k| \tag{2.2}$$

- **Distanza euclidea**: ottenuta ponendo $p = 2$

$$D_2(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \tag{2.3}$$

- **Chess-board distance**: ottenuta ponendo $p = \infty$

$$D_\infty(x,y) = \max(x_k - y_k) \tag{2.4}$$

2.1.2 Misure di similarità

Misurano il livello di similarità tra due punti e sono comprese tra 0 e 1.

- **Coefficiente di match**:

$$D_{match}(x,y) = x^T y \tag{2.5}$$

- **Coefficiente di overlap**:

$$D_{overlap}(x,y) = \frac{x^T y}{min(||x||, ||y||)} \tag{2.6}$$

- **Cosine Similarity**

$$D_{cos}(x,y) = \frac{x^T y}{||x||||y||} \tag{2.7}$$

- **Coefficiente di Dice**:

$$D_{dice}(x, y) = \frac{2x^T y}{||x||^2 + ||y||^2} \tag{2.8}$$

- **Similarità esponente**:

$$D_{exp}(x, y) = \exp\left(-|x - y|^{\alpha}\right) \tag{2.9}$$

2.2 Tipologie di Clustering

Le tecniche di clustering si possono basare principalmente su due filosofie:

- **Bottom-Up**[3] o metodi *agglomerativi*: questa filosofia prevede che inizialmente tutti gli elementi siano considerati cluster a sé stanti, e in seguito l'algoritmo provvede ad unire i cluster più vicini. L'algoritmo continua ad unire elementi al cluster fino ad ottenere un numero prefissato di cluster, oppure fino a che la distanza minima tra i cluster non superi un certo valore, o ancora in relazione ad un determinato criterio statistico prefissato.

- **Top-Down**[4] o metodi *divisivi*: all'inizio tutti gli elementi sono un unico *cluster*, e poi l'algoritmo inizia a dividere il *cluster* in tanti gruppi di dimensioni inferiori. Il criterio che guida la divisione è naturalmente quello di ottenere gruppi sempre più omogenei. L'algoritmo procede fino a che non viene soddisfatta una regola di arresto generalmente legata al raggiungimento di un numero prefissato di cluster.

Altre proprietà secondo cui è possibile effettuare una distinzione tra algoritmi di *clustering* sono:

- **Esclusività**: rappresenta la possibilità per un elemento di appartenere a più *clusters* contemporaneamente. E' utile per rappresentare punti di confine o diverse tipologie di classi.

- **Fuzzyness**: in un sistema di *clustering* di tipo *fuzzy* un elemento appartiene a tutti i *clusters* con un livello di appartenenza compreso tra 0 e 1; la somma dei pesi per ciascun elemento deve essere uguale a 1.

[3] Dal basso verso l'alto
[4] Dall'alto verso il basso

- **Completezza**: in un sistema non completo (parziale) alcuni punti potrebbero non appartenere a nessuno dei gruppi.
- **Eterogeneità**: i *clusters* possono avere dimensioni, forma e densità molto diverse tra di loro.

Per valutare la bontà di un algoritmo di clustering occorre prendere in considerazione diversi aspetti quali:

- la *scalabilità*, sia in termidi di tempo che di spazio,
- l'abilità di gestire tipi di dati diversi,
- la richiesta di conoscenza di dominio per poter determinare i parametri in input, che deve essere minima,
- l'abilità di gestire il rumore e gli *outliers*[5],
- l'*indipendenza* dall'ordine in cui vengono forniti i dati di input,
- la capacità di accettare vincoli definiti dall'utente,
- l'*interoperabilità* e l'*usabilità*.

2.2.1 Clustering Gerarchico

Lo *hierarchical clustering* produce un insieme di classi rappresentate come un albero gerarchico. Normalmente per la rappresentazione viene utilizzato un *dendogramma*, cioè un diagramma ad albero nel quale le foglie rappresentano i singoli elementi da raggruppare e ciascun nodo rappresenta un *cluster* (si veda la figura 2.2). Per la creazione di un clustering di tipo gerarchico si può utilizzare sia un approccio agglomerativo, nel quale si parte con un solo elemento in ogni *cluster* e ad ogni passo si uniscono i *cluster* più vicini fino a che non si ottiene un solo *cluster* (oppure k *clusters*), sia un approccio divisivo (*repeated bisections*), dove si parte con un unico gruppo contenente tutti gli elementi e si separa, ad ogni passo, il *cluster* più lontano finchè i *clusters* non contengono un solo elemento (oppure k elementi).

Per poter eseguire un algoritmo di *clustering* gerarchico occorre aver definito una misura di similarità tra *clusters*. Sono stati proposti diversi metodi:

[5]Termine utilizzato in statistica per definire, in un insieme di osservazioni, un dato anomalo

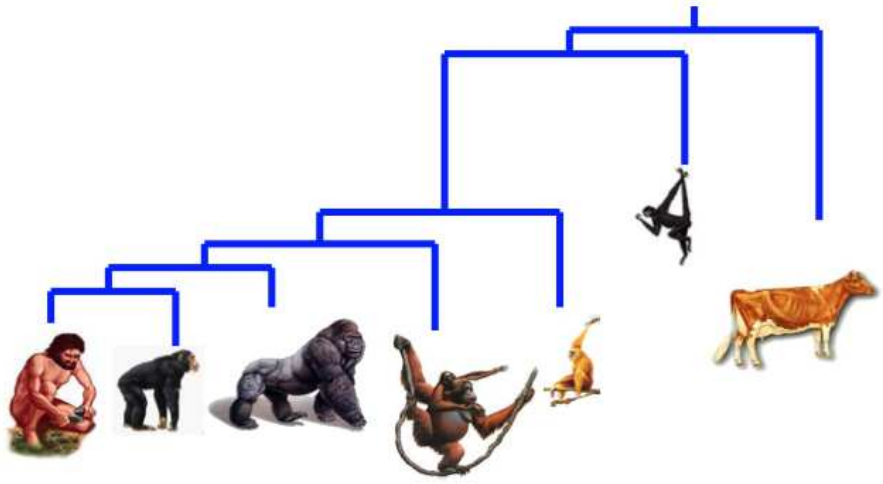

Figura 2.2: Esempio di *clustering* gerarchico rappresentato con un *dendogramma*

- **Single linkage**: In questo metodo la distanza tra due classi è determinata dalla distanza tra i due oggetti più vicini appartenente a due classi diverse (2.3).

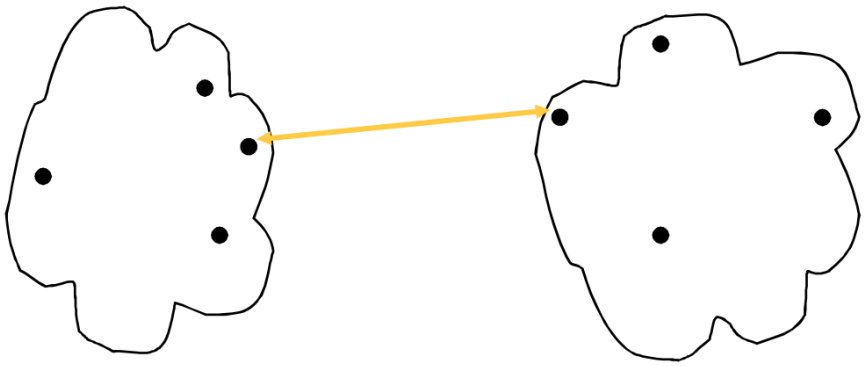

Figura 2.3: Single linkage

- **Complete linkage**: La distanza tra due *clusters* viene determinata dalla più grande distanza tra due oggetti appartenenti ai differenti *clusters* (2.4).
- **Group Average**: In questo metodo si calcola la distanza tra due gruppi come la media delle distanza tra tutte le coppie di oggetti appartenenti ai due diversi *clusters* (2.5).

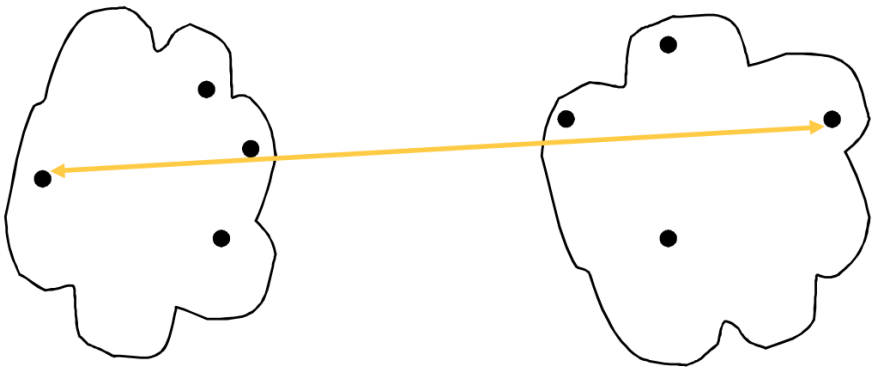

Figura 2.4: Complete linkage

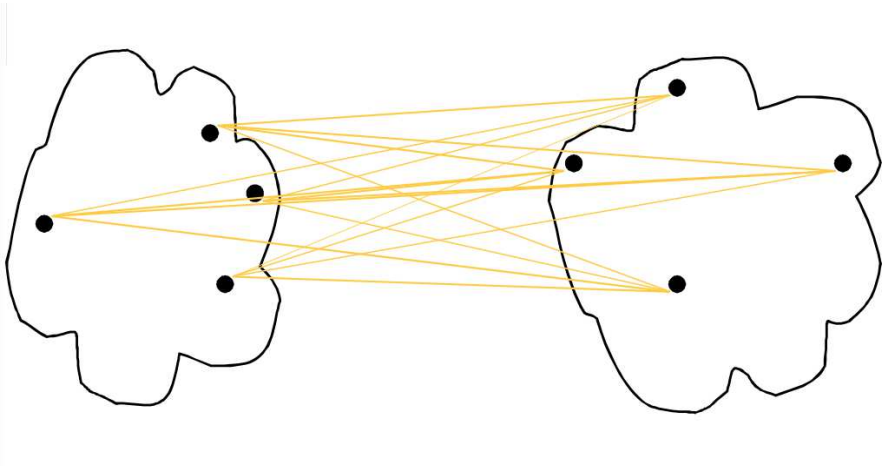

Figura 2.5: Group Average

- **Wards Linkage**: Si tenta di minimizzare la varianze dei *clusters* che vengono fusi.

Pro

Il clustering gerarchico offre numerosi vantaggi:

+ Non c'è alcun bisogno di specificare il numero di *clusters* in anticipo;

+ Può identificare una tassonomia, cioè una classificazione gerarchica di concetti;

+ La natura gerarchica trova corrispondenza nell'intuizione umana per determinati domini.

Contro

Tuttavia sono naturalmente presenti alcune limitazioni:

- L'algoritmo non scala bene: la complessità computazionale è di almeno $O(n^2)$ dove n è il numero totale di oggetti presenti;

- E' presente il problema dei minimi locali;

- Sensibile a rumore e *outliers* in molte configurazioni.

2.2.2 Distribution-based Clustering

Si assume che i dati vengano generati da un modello probabilistico e si tenta di individuare quale sia questo modello. Un *cluster* può quindi essere definito come un insieme di oggetti appartenenti alla medesima distribuzione.

Uno degli algoritmi di tipo *distribution-based* più conosciuti è quello dell'**Expectation-Maximization** (EM). In questo algoritmo il *dataset* è di solito modellato con un fissato numero di distribuzioni Gaussiane inizializzate casualmente, i cui paramentri vengono iterativamente ottimizzati in modo da rappresentare meglio il *dataset*. Esso rappresenta un metodo iterativo per massimizzare la *likelihood*, cioè equivale a calcolare

$$\theta = argmax(L(D, \theta)) \tag{2.10}$$

dove D è la distribuzione da cui vengono generati i dati e θ rappresenta i suoi parametri.

Mentre teoricamente questi metodi sono eccellenti, essi soffrono del problema dell'*overfitting*[6] se non si pongono dei limiti alla complessità del modello. Inoltre bisogna considerare che per i *dataset* reali, potrebbe non esserci un modello matematico che l'algoritmo sia in grado di ottimizzare.

[6]Fenomeno per cui un modello statistico si adatta ai dati osservati usando un numero eccessivo di parametri, risultando in una perdita di generalità del modello

2.2.3 Centroid-based Clustering

Nel *centroid-based clustering* i *clusters* vengono rappresentati da un vettore centrale che non necessariamente fa parte del *dataset*. Uno degli algoritmi più famosi di questo tipo è il **K-Means**, una variante dell'algoritmo di *Expectation-Maximization* (EM). L'assunzione che sta alla base è che gli attributi degli oggetti possano essere rappresentati come vettori, e che quindi formino uno spazio vettoriale.

L'algoritmo si compone dei seguenti passi:

1. Si decide il valore di K.
2. Vengono inizializzati i K centroidi dei *clusters* (casualmente, se necessario).
3. Si decide la classe di appartenenza degli N punti del *dataset* assegnando ciascuno di essi al *cluster* il cui centroide è più vicino.
4. Vengono ricalcolati i K centroidi assumendo che l'assegnamento degli elementi ai *clusters* effettuata al passo precedente sia corretta e si calcola la funzione di costo

$$J = \sum_{k=1}^{K} \sum_{i=1}^{N_k} ||{x_i}^{(k)} - c_k||^2 \tag{2.11}$$

   dove K è il numero totale di *clusters*, N_k è il numero di punti assegnati al *cluster* k e ${x_i}^{(k)}$ e c_k sono rispettivamente un generico punto assegnato al *cluster* k e il centroide del *cluster* k.
5. Si ripete dal punto 3. fino a quano i centroidi restano invariati oppure J è minore di una soglia prefissata ϵ.

Pro

Il K-Means è un algoritmo molto utilizzato per i seguenti motivi:

+ E' un algoritmo relativamente efficiente: la sua complessità computazionale è $O(tkn)$ dove n è il numero di oggetti del *dataset*, k è il numero di *clusters* e t è il numero di iterazioni.

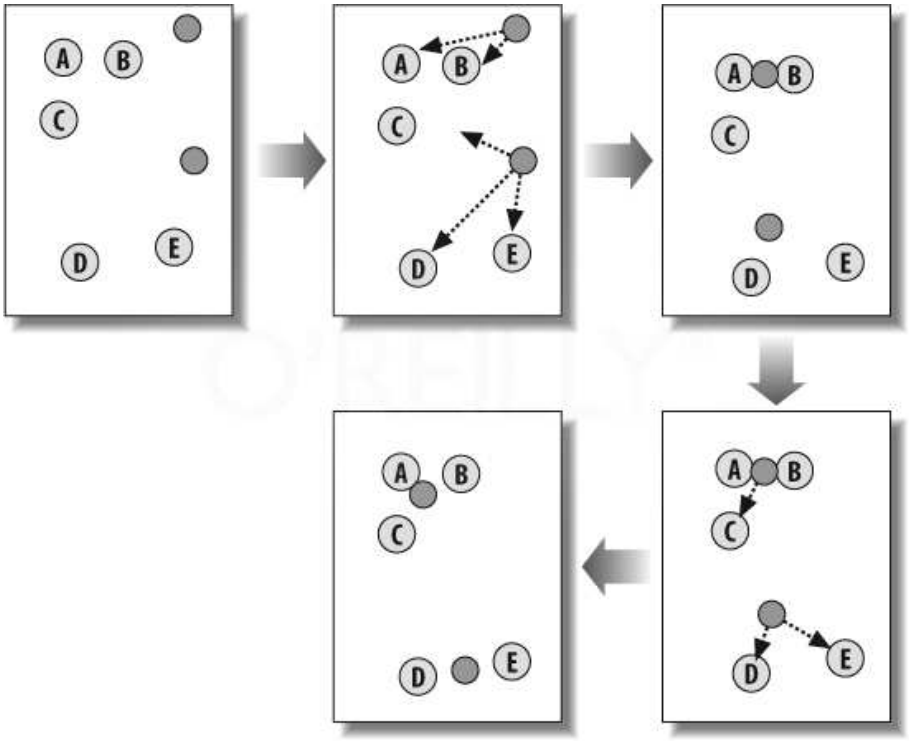

Figura 2.6: Passi dell'algoritmo K-Means

+ La maggior parte delle volte l'algoritmo termina individuando un ottimo locale; tramite l'utlizzo di tecniche come gli *algoritmi genetici* è possibile in seguito trovare un ottimo globale.

+ E' semplice da implementare.

Contro

Numerosi sono però le problematiche che questa metodologia porta con sè:

- Bisogno di specificare a priori il numero di *clusters* desiderato (non sempre lo si conosce a priori).

- Incapacità di gestire rumore e *outliers* che possono tendere a spostare fortemente i centroidi.

- Può essere applicato solo su dati per cui è possibile definire il concetto di media.

- La soluzione finale dipende fortemente dall'inizializzazione (la scelta iniziale della posizione dei centroidi).

- L'algoritmo è basato su una distanza di tipo euclidea, perciò non è in grado di identificare *clusters* di forma non sferica.

2.3 Cluster Validation

Vi sono varie possibilità per la valutazione delle soluzioni fornite dagli algoritmi di *clustering*. Normalmente queste valutazioni vengono utilizzate per confrontare le *performance* di diversi algoritmi sullo stesso insieme di dati. Esistono due tipologie di valutazione:

- Valutazione interna
- Valutazione esterna

2.3.1 Valutazione interna

Quando i risultati dell'applicazione di un algoritmo di *clustering* vengono valutati sugli stessi dati su cui è stato effettuato il *clustering* si parla di valutazione interna. Questi metodi assegnano il punteggio migliore agli algoritmi che producono gruppi omogenei internamente che siano però significativamente differenti tra loro. Il problema principale dei criteri di valutazione interna è che non necessariamente rispecchiano l'efficacia pratica dell'algoritmo valutato.

Alcuni indici di questo tipo sono:

- **Indice di Davies-Boulduin**:

$$DB = \frac{1}{K}\sum_{i=1}^{K}\max_{i\neq j}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{2.12}$$

  dove K è il numero di *clusters*, σ_x è la distanza media tra tutti gli elementi del cluster x e il centroide c_x , e $d(c_i, c_j)$ è la distanza tra i centroidi c_i e c_j. Gli algoritmi che producono bassa similarità *inter-cluster* e alta similarità *intra-cluster* avranno un basso indice di *Davies-Boulduin*.

- **Indice di Dunn**: E' definita come il rapporto tra la minima distanza *inter-cluster* e la massima distanza *intra-cluster*

$$D = \min_{1 \leq i \leq K} (\min_{1 \leq j \leq K, i \neq j} (\frac{d(i,j)}{\max_{i \leq k \leq K} d'(k)})) \tag{2.13}$$

  dove $d(i,j)$ rappresenta la distanza tra i *clusters* i e j e $d'(k)$ misura la distanza *intra-cluster* del *cluster* k.

2.3.2 Valutazione esterna

Nella valutazione esterna, i risultati vengono valutati basandosi su dati che non sono stati usati per il *clustering*, vale a dire degli insiemi di istanze pre-classificate generalmente create da *human experts*. Questi metodi di valutazione misurano quanto la soluzione di *clustering* proposta è vicina ai dati di test forniti. Lo svantaggio di questo criterio è che mira alla riproduzione di conoscenza ben nota, il che potrebbe non essere il risultato desiderato dall'applicazione che si intende valutare. Alcune note misure per la valutazione esterna sono le seguenti:

- **F-Measure**(Si veda la sezione 1.2.2).

- **Rand Measure**:

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.14}$$

  dove TP sono i veri positivi, FP i falsi positivi, FN i falsi negativi e TN i veri negativi. Rappresenta la percentuale di decisioni corrette prese dall'algoritmo.

- **Indice di Jaccard**:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}. \tag{2.15}$$

  Assume un valore che varia tra 0 e 1 dove un valore di 0 indica che i due insiemi non hanno alcun elemento in comune mentre il valore 1 indica che i due insiemi sono identici.

- **Indice di Fowlkes–Mallows**:

$$FM = \sqrt{\frac{TP}{TP + FP} * \frac{TP}{TP + FN}}. \tag{2.16}$$

più alto è il valore dell'indice, maggiore è la similarità tra i *clusters* e i dati di *benchmark*.

Capitolo 3

Il software SenseClusters

SenseClusters è una suite di programmi scritti in linguaggio Perl che consente il *clustering* non supervisionato di contesti simili tra loro. E' un sistema completo che supporta la selezione di *features*, la creazione di vari tipi di rappresentazione del contesto, la riduzione della dimensionalità tramite *Singular Value Decomposition*[1], il *clustering* e l'analisi dei risultati. Al suo interno combina strumenti specializzati come NSP (*Ngram Statistics Package*), SVDPACK, il linguaggio Perl e CLUTO allo scopo di fornire una varietà di scelte e un'elevata efficienza ad ogni passo di elaborazione dei dati.

Il software richiede Perl 5.6.0 o superiore ed è originariamente disponibile per ambiente Linux, Windows e Solaris. Una versione dello script di installazione per Mac OS X (che verrà probabilmente resa disponibile a breve per la comunità), è stata invece realizzata nella fase iniziale di questo lavoro di tesi; essa risolve dei problemi di incompatibilità di alcuni dei moduli utilizzati da *SenseClusters* con il sistema operativo della Apple.

E' anche disponibile una versione web del software, fornita in due versioni, una principale detta di *production* e una di riserva, detta di *backup*, nel caso la versione principale dovesse essere temporaneamente non disponibile.

[1] Tipologia particolare di fattorizzazione basata sull'uso di autovalori e autovettori, utilizzata per produrre un'approssimazione della matrice originaria con il minor rango

3.1 Descrizione generale

SenseClusters è fortemente basato su *features* lessicali e non fa affidamento su insiemi di dati di *training* creati manualmente o sorgenti di conoscenza esterne; perciò è **indipendente dalla lingua**. L'unico prerequisito è che la lingua utilizzata deve essere suddivisibile in *token*[2] tramite l'uso di espressioni regolari Perl (definibili dall'utente).

Può essere utilizzato per risolvere problemi di *sense discrimination* di parole ambigue, attraverso l'utilizzo di una rappresentazione di tipo *headed*, dove ciascun contesto è centrato intorno al termine di cui si vuole identificare il significato. In questo caso i contesti contenenti la data *target word* vengono raggruppati e ciascun gruppo si assume corrisponda ad un differente significato di tale parola. Inoltre può essere utilizzato in problemi di raggruppamento di brevi porzioni di testo che non contengono alcuna *target word* (rappresentazione *headless*); in questo caso si effettua il *clustering* dell'intero contesto al fine di determinare il significato dell'intera porzione di testo. Un esempio di problemi che possono essere affrontati utilizzando una rappresentazione *headless* può essere quella della categorizzazione di email o di articoli di giornale. *SenseClusters* può anche essere applicato al problema del raggruppamento di parole o *features* lessicali con l'obiettivo di scoprire sinonimi, antonimi[3] o altre classi di parole. Più in generale può essere utilizzato per ogni attività che richieda il riconoscimento di unità contestualmente simili o parole che occorrono in contesti simili.

Il programma è organizzato in una gerarchia di *directory* strutturata in questo modo:

- **preprocess**/: contiene tutti gli strumenti per il processamento preliminare del testo. SenseClusters è in grado di gestire testo sia in formato *plain*, sia in formato Senseval-2. Per quanto riguarda il testo semplice, *SenseClusters* fornisce il programma *text2sval.pl* che si occupa di convertire il testo in formato Senseval-2, in modo che possa poi essere processato utilizzando una delle tante *utility* offerte, come ad esempio:
  - *balance.pl* - Bilancia la distribuzione dei sensi rimuovendo alcune istanze;

[2]Elemento di base su cui opera un analizzatore sintattico
[3]Parole con significato opposto

- *filter.pl* - Rimuove le istanze che sono associate a *sense tags* che occorrono poco di frequente;
- *frequency.pl* - Consente di visualizzare la distribuzione di frequenza dei sensi;
- *maketarget.pl* - Crea una espressione regolare Perl per la *target word*, identificando tutte le forme che essa assume nel file Senseval-2 di input;
- *prepare_ sval2.pl* - Effettua delle verifiche sui dati di test per assicurarsi che siano adatti agli esperimenti;
- *preprocess.pl* - Si occupa dei processi di tokenizzazione e formattazione e consente di dividere il testo in una parte di *training* e una parte di *test*;
- *sval2plain.pl* - Converte dal formato *plain text* al formato Senseval-2;
- *windower.pl* - Diminuisce la dimensione di un contesto tagliando via le parole fuori da una data finestra di W *words* centrata sulla *target word*.

- **count**/: contiene *reduce-count.pl*, utile per ridurre la dimensione del *feature space*, rimuovendo i termini che non compaiono nei dati di valutazione.
- **matrix**/: al suo interno vi sono gli strumenti per costruire la matrice di similarità
  - *bitsimat.pl* - crea la matrice da vettori binari;
  - *simat.pl* - crea la matrice da vettori non binari (interi o reali);
- **vector**/: si occupa della rappresentazione dei contesti come vettori da *clusterizzare*. Al suo interno troviamo:
  - *nsp2regex.pl* - Crea espressioni regolari che rappresentano le *features* a partire dall'output di NSP (cioè il software che creai gli n-grammi);
  - *order1vec.pl* - Crea *context vectors* del primo ordine;
  - *order2vec.pl* - Crea *context vectors* del secondo ordine;
  - *wordvec.pl* - Crea *word vectors* dall'output di NSP.

- **svd**/: contiene gli strumenti per l'interfacciamento con SVDPACKC come ad esempio:

  - *mat2hardbo.pl* - Converte le matrici dal formato proprio di *SenseClusters* al formato Harwell-Boeing;
  - *svdpackout.pl* - Ricostruisce una matrice dai suoi *singular vectors* così come vengono trovati da SVDPACKC.

- **clusterstopping**/: ha al suo interno *clusterstopping.pl* il quale si occupa di predire il numero di *clusters* in cui un dato insieme di dati dovrebbe essere diviso. Fornisce tre differenti misure di stop.

- **evaluate**/: si compone di strumenti per la valutazione dei risultati di *SenseClusters* confrontati con il *gold standard*:

  - *cluto2label.pl* - Converte l'output del *clustering* di CLUTO in una matrice di confusione di tipo *cluster by sense*[4] per la valutazione;
  - *format_ clusters.pl* - Mappa un file di output contenente la soluzione fornita da CLUTO in un file di tipo Senseval-2 per fornire una forma di output più leggibile;
  - *label.pl* - Assegna dei *sense tags* ai *clusters* individuati per la valutazione di *report.pl*;
  - *report.pl* - Valuta le prestazioni in termini di Precision, Recall e F-Measure e fornisce una matrice di confusione.

- **clusterlabel**/: contiene al suo interno *clusterlabeling.pl* il quale si occupa di selezionare coppie di parole significative dalle istanze dei contesti e di assegnarle come etichetta dei *clusters*. Queste etichette possono essere descrittive (le top n coppie di termini più significative) o discriminative (cioè quelle coppie di parole appartenenti solo al cluster che si vuole etichettare).

Per facilitare l'utilizzo del software è stato sviluppato un programma *wrapper* chiamato **discriminate.pl** che consente di eseguire *SenseClusters* con un singolo comando. Tramite *discriminate.pl* si possono impostare tutti i parametri desiderati per l'esecuzione di un intero esperimento, a partire dall'algoritmo di

[4] Avente *clusters* su una dimensione e sensi sull'altra

clustering, passando per la rappresentazione del contesto che si vuole utilizzare, fino ad arrivare al metodo di valutazione delle *performance* desiderato.

3.2 Tipologie di features

SenseClusters distingue tra differenti contesti in cui una *target word* occorre basandosi su un insieme di *features* identificate da un *raw corpora*. Esso utilizza l'*Ngram Statistics Package* [28], il quale è in grado di estrarre *features* lessicali da grandi *corpora* utilizzando tagli di frequenza e varie misure di associazione. SenseClusters attualmente supporta l'uso di varie tipologie di *features*:

- **Unigrams**: sono termini individuali che occorrono sopra un determinata frequenza di taglio. Possono essere un tipo di *features* efficace se sono condivise da un minimo di due contesti, ma diventano inutili se sono comuni a tutti i contesti. Normalmente per evitare questo problema si fa uso di una *stop-list*[5] per escludere le *non-content words*[6] molto comuni.

- **Bigrams**: sono coppie di parole che occorrono al di sopra di una data frequenza di taglio e che hanno un punteggio statisticamente significativo in un test di associazione.

- **Co-occurrence features**: sono dei *bigrams* che includono la *target word*. In effetti le *co-occurrences* servono a dare un parametro di localizzazione agli *unigrams*, dal momento che si selezionano solo quelle parole che occorrono entro un determinato numero di posizioni dalla *target word*.

I criteri di selezione delle *features* possono essere modificati utilizzando delle opzioni specifiche:

- *remove* F: rimuove quelle *features* che occorrono meno di F volte, in modo da filtrare eventuale rumore;

- *window* W: specifica la dimensione della finestra per i bigrammi o le co-occorrenze. Le coppie di *words* che co-occorrono all'interno della finestra specificata (W indica che possono esserci al massimo $W - 2$ *words* intermedie) andranno a formare le *features*;

[5] Liste di parole da non considerare
[6] Parole a cui non può essere associato un significato

- *stat STAT*: con questa opzione attiva la matrice di associazione tra parole userà lo *score* calcolato da uno specifico test statistico, piuttosto che il semplice conteggio delle co-occorrenze tra coppie di parole. Sono supportati i seguenti test:

  - *Dice Coefficient*
  - *Log Likelihood Ratio*
  - *Odds Ratio*
  - *Phi Coefficient*
  - *Point-Wise Mutual Information*
  - *True Mutual Information*
  - *Chi-Squared Test*
  - *T-Score*
  - *Left Fisher's Test*
  - *Right Fisher's Test.*

- *stat_ score S*: se si è indicato di voler utilizzare una misura di associazione statistica, è possibile eliminare quelle *features* il cui punteggio è inferiore al punteggio S specificato.

3.3 Rappresentazione del contesto

Data la quantità limitata di informazione disponibile nei metodi non supervisionati, la scelta della rappresentazione del contesto che deve essere sottoposto a *clustering* è di vitale importanza. Sono state sviluppate in effetti numerose tecniche [10]. Il più conosciuto e probabilmente più ovvio metodo è quello di utilizzare un approccio di tipo *bag of words* facendo sì che ciascuna parola nei contesti da clusterizzare rappresenti una *feature*. Questo tipo di rappresentazione è conosciuta come rappresentazione del **primo ordine** dal momento che i contesti sono rappresentati direttamente dalle parole che occorrono in essi. E' possibile anche annotare *part of speech* o processare il testo in altro modo ed usare le informazioni sintattiche o linguistiche risultanti come *feature* ma il tipo di rappresentazione

sottostante resterà invariato. Più semplicemente i metodi del primo ordine stabiliscono la similarità tra contesti trovando quei contesti che condividono il maggior numero di parole tra loro. Questo può essere ragionevole se si ha a disposizione una grande quantità di dati di *test* che includono una terminologia specializzata, ma può portare delle difficoltà nel caso di piccole quantità di dati, soprattutto se questi dati sono affetti da rumore.

Un'alternativa è data dai metodi del **secondo ordine**; in essi i termini presenti nei contesti da clusterizzare vengono rappresentati basandosi su un'informazione indiretta. Molti di questi metodi trovano le loro origini nel *Latent Semantic Indexing* (LSI), applicato originalmente a problemi di *Information Retrieval*. L'idea generale dell'LSI è di rappresentare le *words* presenti in una collezione di contesti con una matrice *word by context*[7], dove le parole che occorrono approssimativamente nello stesso insieme di documenti vengono identificate come simili tra loro. In seguito questa matrice viene ridotta, di solito attraverso l'uso della *Singular Value Decomposition* (SVD), in modo da diminuire la quantità di rumore presente nei dati e rendere le relazioni tra i concetti sottostanti più chiare. Schütze [34] ha sviluppato una estensione della LSI che rende possibile il suo utilizzo all'interno di problemi di *Word Sense Discrimination*. Invece di creare una matrice di tipo parola per documento, ha introdotto l'idea di usare una matrice di co-occorrenze *word by word*[8] per rappresentare ciascuna *word*. In questo caso i termini che occorrono con lo stesso sotto-insieme di parole avranno dei *word vectors*[9] simili in questa matrice di co-occorrenza e saranno identificate come simili tra loro. Nell'approccio di Schütze i dati da cui la matrice è costruita possono provenire da una sorgente esterna oppure dai contesti di cui si sta effettuando il *clustering*. Indipendentemente dalla provenienza dei dati, ciascun termine in un contesto viene rimpiazzato da un vettore che rappresenta le parole che co-occorrono con esso. Effettuando la media dei vettori di tutte le parole all'interno di un contesto viene calcolato un singolo vettore che diventa la rappresentazione *word by word* di secondo ordine di quel contesto. Una volta che tutti i *context* sono stati rappresentati in questo modo l'algoritmo di *clustering* identifica quanti differenti significati esistono all'interno del *dataset*, nonché il senso associato a ciascun contesto.

[7] avente le parole su una dimensione e i contesti sull'altra
[8] Avente parole su entrambe le dimensioni
[9] Vettori composti da parole

Un altro metodo del secondo ordine che è strettamente legato al LSI è la Latent Semantic Analysis [17]. Come nel LSI, la LSA utilizza una rappresentazione *word by context* che viene ridotta tramite SVD in modo da individuare termini e concetti simili tra loro. La metodologia utilizzata è molto simile; la principale differenza tra LSI e LSA sta nel dominio di utilizzo per cui sono state ideate: il LSI è spesso visto come una tecnica di *Information Retrieval* mentre la LSA è spesso usata in applicazioni che riguardano la psicologia, le scienze cognitive o l'educazione.

SenseClusters consente di specificare il tipo di rappresentazione che si vuole utilizzare tramite il parametro *–context ORD* dove *ORD* può assumere il valore *o1* nel caso si intenda utilizzare una rappresentazione del primo ordine o *o2* per utilizzare una rappresentazione del secondo ordine.

Di default gli elementi di un *feature vectors* rappresentano il numero di co-occorrenze per ciascuna coppia di parole, mentre i *context vectors* mostrano la media dei *feature vectors* delle parole occorrenti nel contesto. Con l'opzione *–binary* è possibile indicare al sistema di utilizzare una rappresentazione binaria in cui i *feature vectors* mostrano solo lo stato di presenza o assenza della particolare coppia di parole nel *training set* e i *context vectors* mostrano l'OR binario dei *feature vectors* delle parole considerate.

3.4 Misure di similarità

SenseClusters supporta la costruzione di una *matrice si similarità* cioè una matrice che contiene per ogni coppia di elementi, la loro mutua similarità; nello specifico ciascun elemento è dato da un vettore e ogni cella della matrice contiene la similarità tra coppie di vettori calcolata secondo una delle seguenti misure:

- Match Coefficient
- Overlap Coefficient
- Dice Coefficient
- Jaccard Coefficient
- Cosine Coefficient

3.5 Algoritmi di clustering

Sono supportati diversi algoritmi di *clustering*:

- *rb* - Repeated Bisections: è un algoritmo partizionale, basato sulla divisione dei dati in due sottoinsiemi ad ogni passo, fino a raggiungere il numero di gruppi desiderato.
- *rbr* - Repeated Bisection attraverso rifinitura k-way: è simile al Repeated Bisections, ma la soluzione finale è ottimizzata globalmente.
- *direct* - *clustering* diretto k-way, in cui i k *clusters* vengono ricercati simultaneamente.
- *agglo* - *Clustering* agglomerativo.
- *graph* - *Clustering* partizionale basato su Graph, costruito secondo un criterio di distanza e diviso in k gruppi tramite algoritmo *min-cut*[10].

3.6 Automatic Cluster stopping

Come si è visto, alcuni algoritmi di *clustering* richiedono che venga specificato in anticipo il numero K di *clusters*; questo può rappresentare un problema in applicazioni reali, dal momento che K è solitamente sconosciuto all'utente. Sono stati perciò sviluppati dei metodi di predizione del numero di *cluster*, detti di *Automatic Cluster Stopping*. Il *Cluster stopping* può essere visto come un problema di selezione del modello, dal momento che un numero di modelli differenti (cioè soluzioni di *clustering*) vengono creati utilizzando diversi valori di K, e il modello che meglio si adatta ai dati osservati viene selezionato basandosi su una *loss function*[11]. L'approccio utilizzato in *SenseClusters* è quello di partire con soluzioni basate su un solo *cluster* e incrementare costantemente il numero di *clusters* finché non viene individuata la soluzione migliore [29]. Attualmente sono quattro le differenti misure supportate: PK1, PK2, PK3 e *Gap Statistic*.

[10] Algoritmo che prevede delle divisioni del grafo in parti tramite tagli, dove a ciascun taglio è associato un costo direttamente proporzionale alla similarità dei sotto-grafi che verrebbero creati dopo la divisione. La soluzione è data dal taglio con costo minore

[11] Funzione di costo utilizzata tipicamente per la stima di parametri

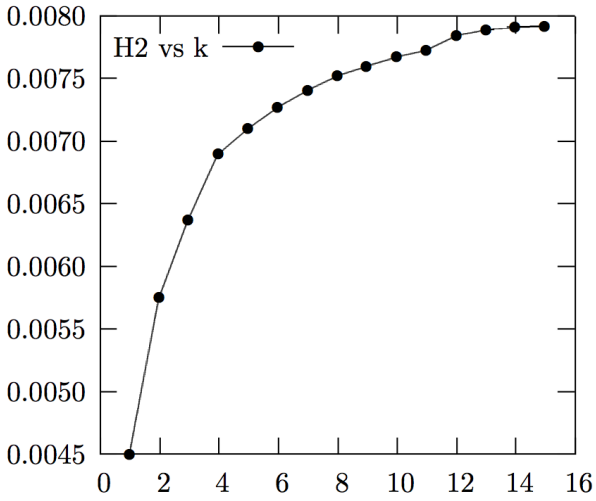

Figura 3.1: Esempio di predizione numero di clusters con misura H2. Il valore predetto di K è 2

3.6.1 PK1

La misura PK1 è basata su [**?**]: si trova soluzione per tutti i valori di k da 1 a N e si determina la media e la deviazione standard per la *loss function*; a questo punto viene calcolato un punteggio per ciascun valore di k sottraendo il valore medio dalla *loss function* e dividendo per la deviazione standard:

$$PK1(k) = \frac{H2(k) - mean(H2[1..deltaK])}{std(H2[1..deltaK])} \tag{3.1}$$

dove $H2$ rappresenta il rapporto tra *intra-cluster similarity* e *inter-cluster similarity* e $deltaK$ è il massimo numero di *clusters* che si intende accettare come risultato. Per selezionare un valore di k viene impostata una soglia, definita empiricamente, in modo tale che nel momento in cui $PK1(k)$ supera tale soglia, $k-1$ viene selezionato come numero appropriato di *clusters*.

3.6.2 PK2

Questa misura è definita come:

$$PK2(k) = \frac{H2(k)}{H2(k-1)} \tag{3.2}$$

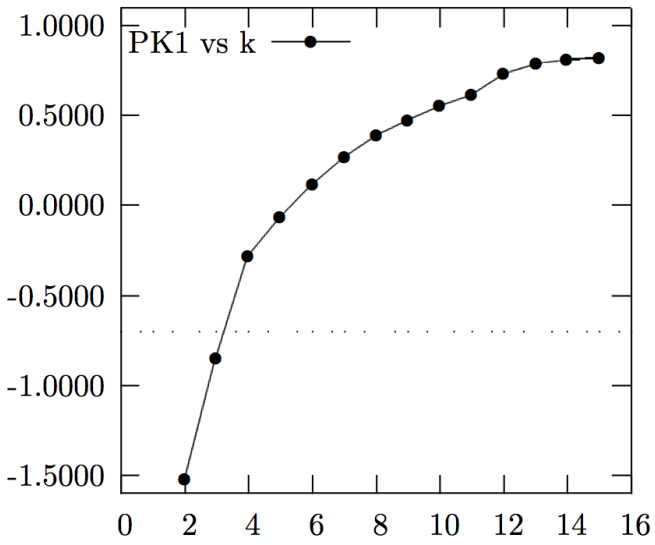

Figura 3.2: Esempio di predizione numero di clusters con misura PK1. Il valore predetto di K è 2

Quando questo rapporto si avvicina ad 1, aumentare k non porta più alcun beneficio. Il k desiderato è quel numero per cui $PK2(k)$ assume il valore più vicino a

$$1 + std_dev(PK2([1..deltaK]) \tag{3.3}$$

che allo stesso tempo sia anche maggiore di

$$1 + std_dev(PK2([1..deltaK]). \tag{3.4}$$

3.6.3 PK3

PK3 utilizza tre valori per individuare il punto in cui la *loss function* aumenta e diminuisce improvvisamente; per un dato valore di k, viene confrontata la *loss function* con il precedente e il successivo valore di k:

$$PK2(k) = \frac{2 * H2(k)}{H2(k-1) + H2(k+1)} \tag{3.5}$$

La misura PK3 assume valore prossimo a 1 se i valori di $H2$ sono crescenti, cioè se vi è un costante miglioramento della qualità del *clustering*. Quando il

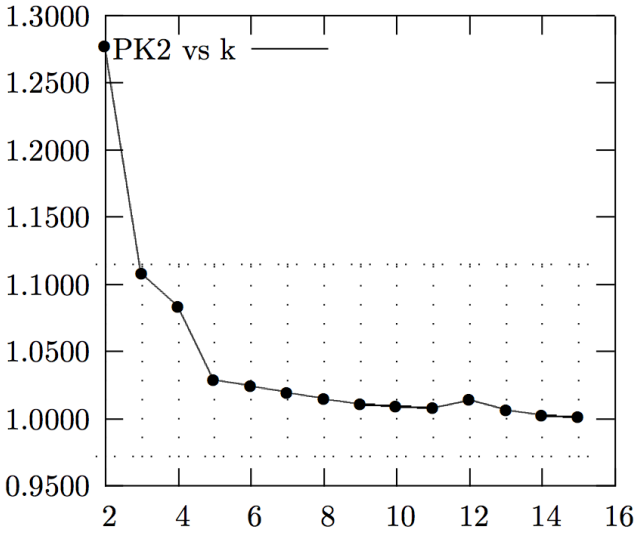

Figura 3.3: Esempio di predizione numero di clusters con misura PK2. Il valore predetto di K è 2

valore assunto è molto maggiore di 1, non si ha più un incremento netto della qualità della soluzione. Per selezionare k si utilizza il valore di PK3(k) che è più vicino possibile alla regione critica definita dalla deviazione standard di PK3.

3.6.4 Gap Statistic

SenseClusters supporta anche una versione adattata della *Gap Statistic*[39]. La *Gap Statistic* si distingue dalle precedenti misure dal momento che non cerca di trovare direttamente un ginocchio nel grafico della *loss function* ma crea, invece, un campione di dati di riferimento che rappresentano i dati osservati come se non avessero *clusters* significativi al loro interno, ma fossero composti semplicemente da rumore. La *loss function* dei dati di riferimento è allora confrontata con quella dei dati osservati, allo scopo di identificare il valore di k nei dati osservati che sia meno simile al rumore e che di conseguenza rappresenti il miglior *clustering* dei dati.

3.7 Strumenti di valutazione delle performance

SenseClusters supporta due differenti tipologie di valutazione: una valutazione interna e una valutazione esterna [32].

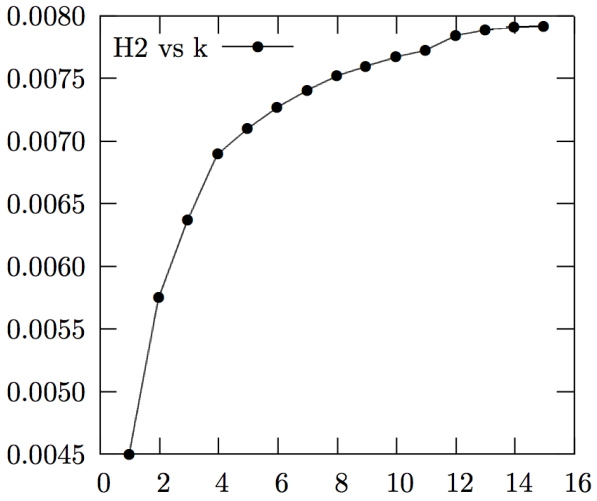

Figura 3.4: Esempio di predizione numero di clusters con misura PK3. Il valore predetto di K è 2

3.7.1 Valutazione esterna

Quando è disponibile una soluzione di *clustering gold-standard* delle istanze, *SenseClusters* costruisce una matrice di confusione che mostra la distribuzione dei sensi conosciuti in ciascuno dei *clusters* individuati nell'esperimento. Un *gold-standard* deve esistere sotto forma di testo *sense-tagged*[12], dove ciascun *sense tag* può essere considerato come la rappresentazione di un *cluster* potenzialmente individuabile.

Nella figura 3.6, le righe C0-C9 rappresentano dieci *clusters* individuati mentre le colonne rappresentano sei *gold-standard senses*. Il valore della cella (i, j) mostra il numero di istanze nell'i-esimo *cluster* che attualmente appartiene al *gold-standard sense* rappresentato dalla j-esima colonna. Si noti che la riga inferiore mostra la distribuzione delle istanze attraverso i sensi, mentre la colonna a destra mostra la distribuzione dei *clusters* individuati.

Per condurre la valutazione dei *clusters* individuati, *SenseClusters* trova il mapping tra *gold-standard senses* e *clusters* che produce la più accurata discriminazione. Il problema dell'assegnamento dei significati ai *clusters* diviene quello del riordinamento delle colonne della matrice di confusione allo scopo di massimizzare la somma degli elementi sulla diagonale; questo corrisponde a numerosi

[12] Annotato con i significati corretti

ben noti problemi, primi fra tutti il Problema di Assegnamento in Ricerca Operativa e la ricerca della massima corrispondenza di un grafo bipartito. Ciascun possibile ordinamento mostra uno schema di assegnamento e la somma degli elementi sulla diagonale indica il numero totale di istanze nei *clusters* individuati a cui viene assegnato il senso corretto.

La figura 3.7 mostra che il *cluster* C1 viene mappato con più precisione con il senso S3 mentre il *cluster* C2 trova corrispondenza con il senso S5 e così via. Ai *clusters* indicati con * non viene assegnato a nessun significato. L'*accuracy* può essere semplicemente calcolata come la somma degli elementi sulla diagonale della matrice di confusione riordinata diviso il numero totale di istanze, mentre la *Precision* può essere calcolata dividendo il numero totale delle istanze correttamente discriminate per il numero delle istanze nei *clusters* mappati ai *gold-standard senses*.

3.7.2 Valutazione interna

Quando i tag derivanti da *gold-standard senses* delle istanze di test non sono disponibili, *SenseClusters* si affida alle metriche interne di CLUTO per riportare la similarità *intra-cluster* e *inter-cluster*. E' disponibile anche un componente (gCLUTO), che consente di visualizzare graficamente i *clusters* individuati, in modo da fornire un supporto visivo per l'impostazione manuale dei parametri dell'algoritmo di *clustering* da utilizzare. Un esempio di output di gCLUTO può essere osservato nella figura 3.8.

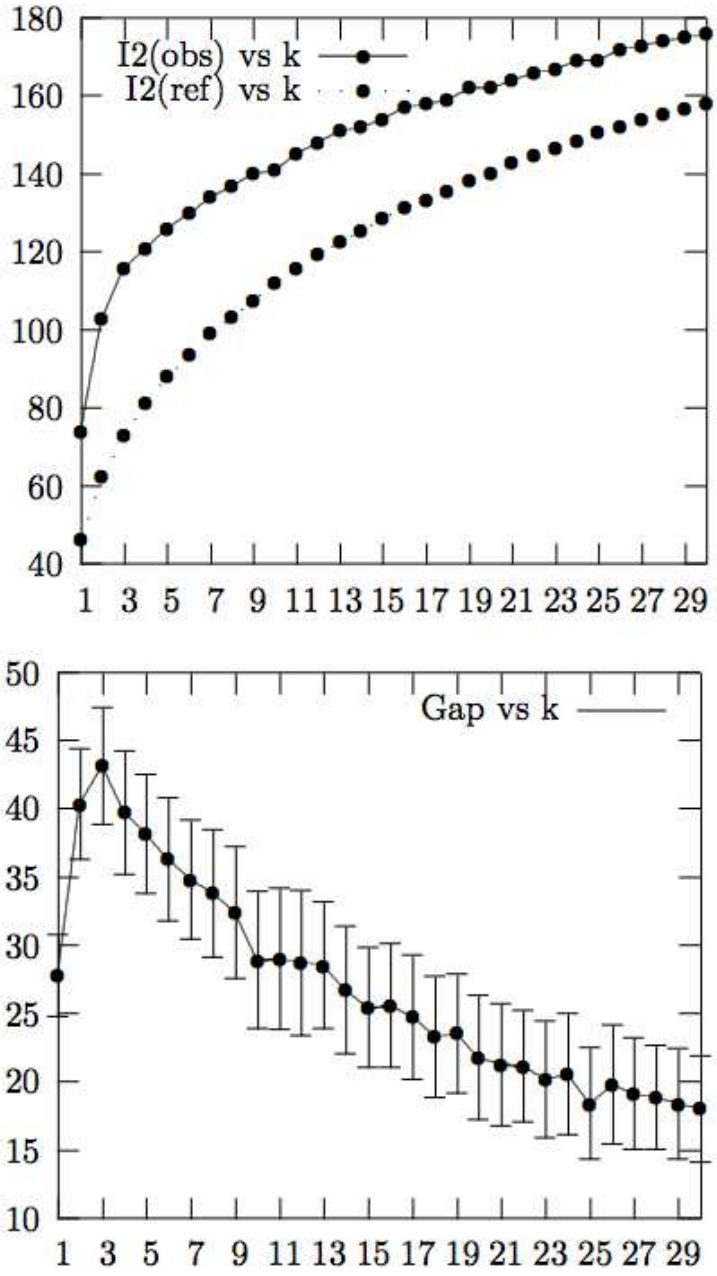

Figura 3.5: Esempio di similarità intra-cluster per dati di riferimento e dati osservati (superiore) e Gap tra loro. Il mumero di clusters predetto è 3

	S1	S2	S3	S4	S5	S6	
C0:	2	3	3	1	99	3	111
C1:	11	5	43	11	11	8	89
C2:	1	19	7	19	208	7	261
C3:	3	15	13	7	37	12	87
C4:	6	5	8	16	143	8	186
C5:	37	18	8	18	186	20	287
C6:	17	7	11	59	14	13	121
C7:	4	9	13	14	163	12	215
C8:	54	20	15	6	16	35	146
C9:	29	51	12	18	11	35	156
	164	152	133	169	888	153	1659

Figura 3.6: Matrice di confusione prima del mapping

	S3	S5	S6	S4	S1	S2	
C1:	**43**	11	8	11	11	5	89
C2:	7	**208**	7	19	1	19	261
C5:	8	186	**20**	18	37	18	287
C6:	11	14	13	**59**	17	7	121
C8:	15	16	35	6	**54**	20	146
C9:	12	11	35	18	29	**51**	156
C0:*	3	99	3	1	2	3	111
C3:*	13	37	12	7	3	15	87
C4:*	8	143	8	16	6	5	186
C7:*	13	163	12	14	4	9	215
	133	888	153	169	164	152	1659

Figura 3.7: Matrice di confusione dopo il mapping

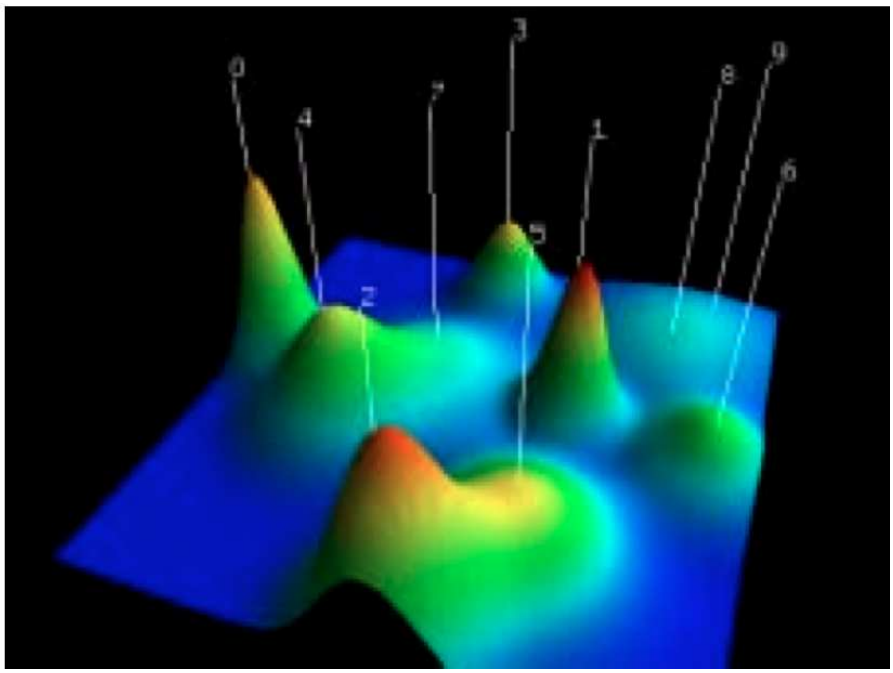

Figura 3.8: Vista MountainView fornita da gCLUTO

Capitolo 4

JRC-Acquis

JRC-Acquis[38] è un *corpus* multi-lingua allineato a livello di *sentence* che può essere visto come un'approssimazione del *corpus* **Acquis Communautaire** creata dal **JRC**.

Il JRC, acronimo per Joint Research Center (Centro Comune di Ricerca, acronimo italiano: CCR), è una direzione generale della Commissione europea che dispone di sette istituti di ricerca dislocati in cinque paesi membri dell'Unione europea (Belgio, Germania, Italia, Olanda e Spagna). Il JRC fornisce un sostegno scientifico e tecnico alla progettazione, allo sviluppo, all'attuazione e al controllo delle politiche dell'Unione europea. A differenza delle università europee, è direttamente finanziato dall'Unione Europea (è un servizio della Commissione Europea), allo scopo di garantire l'indipendenza delle attività di ricerca da interessi privati o dalle singole politiche nazionali, come condizione essenziale per perseguire la sua missione internazionale. Il JRC svolge un ruolo di coordinamento e ricerca in numerose reti comunitarie di enti nazionali di ricerca, università e industria avanzata degli stati membri dell'Unione Europea, oltre ad effettuare un vasto insieme di ricerche indipendenti che si avvalgono delle competenze dei migliori scienziati europei che lavorano direttamente nel centro o vi svolgono periodi di ricerca. Nei suoi laboratori si svolgono complessi studi ed esperimenti per conto delle Istituzioni Europee. Il JRC collabora con enti e reti extra-europee e mondiali nel campo scientifico e della normativa.

L'*Acquis Communautaire* (AC) è l'intero corpo delle leggi dell'Unione Europea (UE) applicabili negli Stati Membri dell'UE. Questa collezione di testi legislativi cambia continuamente e comprende testi selezionati, scritti dal 1950 ad oggi.

Attualmente (2013) l'Unione Europea ha 27 Stati Membri per un totale di 23 Lingue ufficiali. I testi legislativi dell'*Acquis Communautaire* esistono in tutte le 23 lingue ufficiali, sebbene la traduzione in Irlandese non sia ancora completa.

Il JRC ha selezionato i documenti dell'*Acquis Communautaire* disponibili in almeno dieci delle lingue ufficiali dei paesi appartenenti all'EU-25[1] e che in più esistono in almeno tre delle nove lingue che sono diventate ufficiali con l'ampliamento dell'Unione Europea nel 2004[2]. Il suo scopo non è quello di essere un *corpus* di riferimento legale, ma quello di fornire un *corpus* ampio e parallelo per attività di ricerca nel campo della linguistica computazionale.

4.1 Descrizione

JRC-Acquis è disponibile in Bulgaro, Ceco, Danese, Tedesco, Greco, Inglese, Spagnolo, Estone, Finlandese, Francese, Ungherese, Italiano, Lituano, Lettone, Maltese, Olandese, Polacco, Portoghese, Rumeno, Slovacco, Sloveno e Svedese.

Il gruppo di ricerca Optima (Open Source Text Information Mining and Analysis) del JRC di Ispra, Italia ha identificato i documenti che fanno parte di AC, e li ha convertiti in formato XML. Successivamente i testi sono stati ripuliti da note e annessi e sono stati allineati a livello di frase (*sentence-alignment*) utilizzando due strumenti diversi:

- **Vanilla**: è un allineatore statistico che basa i tentativi di allineamento esclusivamente sulla lunghezza delle frasi [11]. Come input prende testi segmentati da *hard link* (affidabili) e *soft link* (probabili). Sebbene i documenti di JRC-Acquis contengono sezioni e paragrafi numerati, molto adatti per essere utilizzati come *hard link*, gli autori hanno deciso per questioni di tempistiche di utilizzare solo *soft link* come le terminazioni di paragrafo.

- **HunAlign** [41]: è un *sentence aligner* indipendente dalla lingua. L'allineamento viene costruito utilizzando una semplice misura di similarità. Questa misura è basata sulla lunghezza delle frasi e la percentuale di parole identiche. I numeri vengono trattati in modo specifico, cosa molto utile per i testi

[1] I paesi dell'UE prima dell'ingresso di Bulgaria e Romania

[2] Repubblica Ceca, Estonia, Ungheria, Lituania, Lettonia, Malta, Polonia, Slovacchia, Slovenia

legali dove la quantità di numeri è elevata (ad esempio nel *corpus* Acquis la percentuale di numeri presenti raggiunge il 6.5%).

Invece di utilizzare una singola lingua di riferimento, tutte le possibili (231) coppie di combinazioni sono state allineate individualmente.

Il *corpus* allineato non viene distribuito direttamente, per questioni di *storage* limitato, ma vengono messi a disposizione i file di allineamento per ciascuna coppia di lingue, sia per il *Vanilla Sentence Aligner* che per *HunAlign*, e viene fornito un programma scritto in Perl (*getAlignmentWithText.pl*) che a partire dai *corpora* mono-lingua delle lingue desiderate, genera il *corpus* bi-lingua utilizzando i file di allineamento. Il file generato è in XML e utilizza la codifica UTF-8 per poter gestire tutti gli insiemi di caratteri.

Language	N° of Texts	Text body			Signature	Annex	Total N° Words (Text + Signature + Annex)
		Total N° Words	Total N° Characters	Average N° Words	Total N° Words	Total N° Words	
cs	**7983**	6000751	38625616	751	715895	1972356	8689002
da	7939	6556131	44497890	825	778125	1505554	8839810
de	7913	6481949	46628367	819	822797	1349791	8654537
el	7782	7267113	47260657	933	991962	1306164	9565239
en	7972	7547154	45372451	946	817085	1568297	9932536
es	7809	**8006579**	48547661	**1025**	792355	1707348	**10506282**
et	7943	4998334	39077676	**629**	431570	1752216	7182120
fi	7735	5141742	43771107	664	618042	1120613	6880397
fr	7862	7814912	46526758	994	865693	1531754	10212359
hu	7489	5403934	40697500	721	594176	1821141	7819251
it	7872	7305910	47068517	928	787131	1582773	9675814
lt	7965	5395807	40011655	677	693712	1869631	7959150
lv	7980	5513265	38544761	690	669890	1946340	8129495
mt	7639	7273072	44168004	952	588103	2162699	10023874
nl	7882	7362017	47846520	934	806123	1593619	9761759
pl	7968	6002780	43373027	753	757757	1953001	8713538
pt	7848	7900690	47529143	1006	746070	1692161	10338921
ro	5792	5125673	33701702	884	474159	3972847	9572679
sk	**5278**	**3914177**	26091927	741	510522	1282176	**5706875**
sl	**7983**	5954252	37646679	745	743699	2017049	8715000
sv	7731	6797710	44589319	879	303527	1356547	8457784
Average	**7,636**	**6.369,712**	**42,456,045**	**833**	**690,876**	**1,764,956**	**8,825,544**

Figura 4.1: Distribuzione dei testi per ciascuna lingua in JRC-Acquis

Nella figura 4.1 è possibile osservare la dimensione del *corpus* in ciascuna delle lingue che lo compongono.

JRC-Acquis è attualmente l'unico *corpus* della sua grandezza disponibile in così tante lingue e molti dei suoi testi sono anche classificati secondo gli *EROVOC subject domains* [1].

4.2 Limiti

JRC-Acquis è un corpus molto ampio che però soffre di alcune limitazioni importanti:

- Eccessiva specificità: essendo composto da una raccolta di testi legislativi, la terminologia utilizzata è specifica del settore. Nella pratica questo rende il *corpus* adatto solo a scenari che non prevedano applicazioni reali in contesti diversi da quello legislativo. Infatti un sistema allenato su *training set* specifico di un dato dominio potrebbe facilmente risultare poco performante su dati di *test* di dominio differente o generale.

- Presenza di *stop-words* specifiche del dominio: sono presenti termini che hanno contenuto di informazione nulla, in quanto presenti nella quasi totalità delle istanze, che sono specifiche del settore legislativo (es. Art., article, Commission, UE, ecc.). Questo può portare, in scenari di *Word Sense Induction*, a risultati falsati dal peso eccessivo attribuito a tali termini, data la loro frequenza, a meno che non vengano create delle *stop-lists* personalizzate per ogni lingua di riferimento.

- Disallineamento a livello di *corpus*: i testi generati con coppie di lingue differenti non sono allineati tra loro. Nello specifico ad ogni istanza di contesto è assegnato un ID; questo ID è differente per ogni coppia di lingua utilizzata e ciò rende molto difficile riuscire a creare *corpora* multi-lingua che utilizzino più di due lingue contemporaneamente.

- Formato del corpus: è necessario manipolare il testo fornito in XML per poterlo trasformare in *plain text*, se necessario, o nei formati più utilizzati per applicazioni di Natural Language Processing (come ad esempio il formato Senseval-2).

- Presenza di testo non puro: il testo è composto da sequenze di caratteri che non sono lettere dell'alfabeto (come sequenze di trattini, di simbolo uguale, *underscore*, ecc.) che rendono il software per la manipolazione del testo più complicato o maggiormente soggetto ad errori.

Capitolo 5

Multilingual Word Sense Induction

Definiamo come *multilingual Word Sense Induction* qualsiasi tecnica di *clustering* effettuato su testo multi-lingua allo scopo di individuare gruppi di parole che condividono lo stesso significato.

In questo lavoro di tesi vengono applicate delle tecniche di *Word Sense Induction* su testo parallelo, cioè allineato a livello di frase tra lingue diverse. In particolare, viene proposto un algoritmo di disambiguazione non supervisionata di tipo *targeted* dove per ciascuna istanza della *target word* si ha a disposizione un unico contesto formato dall'unione della frase in cui è presente l'istanza specifica e le diverse traduzioni di tale frase in altre lingue. La disambiguazione viene eseguita per le *words* in una sola delle lingue e le altre lingue vengono utilizzate come supporto.

5.1 Ipotesi

L'ipotesi fondamentale alla base di questo approccio alla *Word Sense Induction* è che l'estensione del contesto in cui una parola occorre possa avere un effetto positivo sulle performance di un algoritmo di disambiguazione. E' stato dimostrato che tale ipotesi risulta verificata nel caso della *Word Sense Disambiguation*, dove l'incremento di prestazioni che si ha passando da una rappresentazione mono-lingua ad una rappresentazione multi-lingua raggiunge il 26% [2] [27].

Un incremento delle *performance* dovuto all'utilizzo di testo multi-lingua troverebbe una doppia giustificazione:

- In primo luogo, grazie all'ampliamento del contesto ci si trova a disposizione un numero maggiore di *features* e quindi di caratteristiche distintive di ciascuna istanza della *target word* da poter utilizzare per il discernimento del senso ad essa associato. Naturalmente l'incremento del numero di parole all'interno di un contesto può portare ad un aumento del rumore e ad una conseguente degradazione delle prestazioni. Per poter gestire questo problema è necessario che venga utilizzato un qualche tipo di filtro *passa-alto* che consenta la selezione delle sole *features* più significative.

- In secondo luogo vi è un'introduzione implicita della conoscenza propria del processo di traduzione all'interno del sistema di disambiguazione. Si pensi ad esempio alla parola *plant* in inglese, che in italiano può essere tradotta come *pianta* o *impianto* a seconda del significato specifico; la conoscenza del termine corretto, in lingua italiana, corrispondente al significato di una specifica istanza della parola *plant* fa già parte del bagaglio di informazione posseduto dal sistema di traduzione (nel nostro caso *human experts*). Nel momento della traduzione tale conoscenza viene trasferita all'interno del contesto dell'istanza di *target word* sotto forma di termini tradotti. Tali termini diventano delle *features* molto significative nel processo di *clustering* dal momento che, se a due parole legate da una relazione di omonimia, corrispondono differenti traduzioni, tali termini tradotti saranno presenti in tutti i contesti associati al rispettivo significato e in nessuno dei contesti associati a significati differenti.

5.2 Impostazioni sperimentali

Si è scelto di utilizzare come lingua di riferimento su cui effettuare la *Word Sense Disambiguation* la lingua inglese e come lingue di supporto sono state introdotte l'italiano, lo spagnolo, il francese e il portoghese.

Sono stati effettuati numerosi esperimenti utilizzando sia la sola lingua inglese, sia testi *bi-lingua* nonchè testi *tri-lingua*[1] e a cinque lingue. I testi bi-lingua sono composti dalla lingua inglese combinata con una delle altre lingue selezionate, mentre i testi tri-lingua sono creati utilizzando la lingua inglese come punto fisso e tutte le combinazioni possibili di coppie delle altre lingue considerate. Il test

[1] Composti da tre lingue

con cinque lingue utilizza tutte le lingue considerate contemporaneamente. Il risultato è un totale di dodici esperimenti con le lingue:

- Inglese
- Inglese - Italiano
- Inglese - Francese
- Inglese - Spagnolo
- Inglese - Portoghese
- Inglese - Francese - Portoghese
- Inglese - Italiano - Francese
- Inglese - Italiano - Portoghese
- Inglese - Spagnolo - Portoghese
- Inglese - Spagnolo - Francese
- Inglese - Spagnolo - Italiano
- Inglese - Italiano - Francese - Spagnolo - Portoghese

Si noti che sono state scartate le combinazioni tri-lingua che differiscono dalle altre solo per l'ordine in cui le lingue vengono utilizzate, dal momento che con le impostazioni utilizzate, l'ordine delle parole non influisce sulla soluzione fornita dall'algoritmo di *clustering*.

Ogni singolo esperimento utilizza gli stessi parametri dell'algoritmo di *clustering*; tali parametri sono stati selezionati empiricamente tramite l'esecuzione di test con tutte le loro possibili combinazioni ed andando a scegliere le impostazioni in grado di offrire le migliori prestazioni sulle parole considerate:

- **Features**: vengono utlizzate *features* di tipo **bi-gramma** occorrenti però entro una finestra di due parole; più semplicemente vengono considerate solo coppie di parole consecutive per la creazione di un bi-gramma. Inoltre vengono selezionate solo quei bi-grammi che occorrono almeno N volte all'interno del *training set*, le cui due componenti raggiungano un punteggio

sul test statistico di associazione basato su **log-likelihood ratio** superiore ad un dato P. Per scartare quelle parole molto comuni ma che non contengono nessuna informazione utile per la *Word Sense Induction* vengono utilizzate delle *stop-list* multi-lingua adattate ad essere utilizzate con *SenseClusters* dallo script *bash* **stopListAdapt.sh**.

- **Contesto**: la rappresentazione del contesto utilizzata è del **secondo ordine**.
- **Algoritmo di Clustering**: viene utilizzato un algoritmo di **clustering diretto** (*k-means*).
- **Space**: viene utilizzato uno spazio di tipo **vettoriale**, cioè vengono *clusterizzati* i *context vector* direttamente.
- **Cluster stop measure**: per individuare automaticamente il numero corretto di *clusters* viene utilizzata la misura **PK2**.

5.3 Estensione del corpus JRC-Acquis

Come è già stato detto, JRC-Acquis è composto da una raccolta di strumenti che consentono la generazione di *corpora* paralleli bi-lingua. Non è stato pensato per un utilizzo che contempli più di due lingue per volta e di conseguenza il suo adattamento agli scopi di questo lavoro non è stato immediato.

E' stato creato un programma denominato **createMLwordsDataset.pl** che per ciascuna delle parole utilizzate negli esperimenti genera un *dataset* tri-lingua a partire da due *dataset* bi-lingua (entrambi aventi l'inglese come una delle lingue). Questo programma può essere in realtà utilizzato per generare *corpora* contenenti un numero teoricamente illimitato di lingue.

- *LANG1*: la prima (o il primo gruppo) delle lingue che si intende unire all'inglese per comporre il *dataset* tri-lingua;
- *LANG2*: la seconda (o il secondo gruppo) delle lingue che si intende unire all'inglese per comporre il *dataset* tri-lingua;
- *WORDSFILE*: il file contenente tutte le *target word* su cui eseguire i test.

A causa del disallineamento dei testi in lingue diverse, la generazione del *dataset* tri-lingua risulta abbastanza onerosa dal momento che richiede necessariamente un'operazione di ricerca. In particolare, dato un contesto presente nel *corpus* in lingua inglese associato ad una *target word*:

- si ricerca tale contesto nel primo dei *dataset* bi-lingua e si seleziona la traduzione corrispondente;
- si ricerca tale contesto nel secondo dei *dataset* bi-lingua e si seleziona la traduzione corrispondente;
- si effettua un'operazione di *merge* tra i due contesti selezionati e il contesto in lingua inglese;
- si salva il contesto generato al punto precedente in un file *plain text* [2].

Una volta terminata la ricerca si converte il file in formato *Senseval-2*.

E' stato inoltre creato un programma **prepareData.sh** che provvede alla generazione dei dati di *training* per ciascuna *target word*, andando a selezionare dall'intero *corpus* JRC-Acquis solo i contesti in cui tale parola occorre e generando così un *dataset* per parola in formato *Senseval-2*, in modo da rendere le successive operazioni più veloci, dal momento che ci si trova in questo modo ad operare su *dataset* di dimensioni contenute aventi al loro interno solo i dati necessari all'esecuzione degli esperimenti.

5.4 Annotazione dei dati di Test

Dal momento che si è scelto di utilizzare una valutazione delle performance di tipo esterno, è stato necessario annotare manualmente i dati di test con un *sense id* corrispondente a dei sensi di tipo *coarse grained* generati utilizzando un dizionario come riferimento.

Tale processo richiede l'annotazione del senso di circa mille frasi per ciascuna combinazione di lingue utilizzata negli esperimenti. Dal momento che il totale degli esperimenti previsti era molto ampio è stato realizzato uno script *bash* di annotazione automatica e generazione dei dati di test **annotate.sh** che a partire

[2] Testo semplice

dalle annotazioni manuali, in lingua inglese, è in grado di generare le istanze di test in una delle restanti combinazioni di lingue e di annotarle con il *correct sense*[3].

Per ciascuna delle *target words*, il processo di annotazione si avvale dei seguenti passi:

- viene eseguita l'annotazione manuale su istanze di test tratte da JRC-Acquis sotto forma di testo bi-lingua (italiano-inglese) in modo da facilitare il compito dell'annotatore (dal momento che si hanno a disposizione due lingue contemporaneamente);
- le annotazioni vengono propagate al testo in sola lingua inglese;
- per ciascuna combinazione di lingue, viene ricercata l'istanza di test in lingua inglese all'interno del *corpus* multi-lingua desiderato, si seleziona l'intero contesto multi-lingua e lo si annota con il senso tratto dalla frase in lingua inglese;
- le istanze di test così generate vengono salvate in un file in formato Senseval-2.

5.5 Esecuzione degli esperimenti

Per l'esecuzione degli esperimenti è stato realizzato un programma **startAllExp.sh** che accetta due parametri:

- *LANGUAGES*: cioè la combinazione di lingue da utilizzare per l'esperimento;
- *WORDSFILE*: il file di testo contenente tutte le *target word* su cui eseguire gli esperimenti.

Esso è in grado di gestire l'intero test e si occupa di tutte le operazioni, come la preparazione dei dati, la divisione del *corpus* in parte di *training* e parte di *test*, la creazione di espressioni regolari per la *target word*, l'esecuzione dei test, la presentazione dei risultati e la pulizia dei file non più utili.

Il processo di esecuzione dei test si compone dei seguenti passi:

[3]Senso associato alla parola dagli esperti

- viene controllato se sono stati effettuati dei test, e in caso affermativo si ripulisce la *directory* corrente dai residui degli esperimenti precedenti;
- vengono recuperati i file contenenti le espressioni regolari utili per il riconoscimento della *target word*, delle parole da ignorare (presenti nella *stoplist*), dei *token*[4] e dei *non-token*;
- se non fatto in precedenza viene generato il *corpus* multi-lingua per ogni *target word* per la data combinazione di lingua, utilizzando il programma **createMLwordDataset** e il programma **prepareData.sh**;
- vengono rimosse le stringhe che non sono considerate *token* e viene diviso il *corpus* in parte di *training* e parte di *test*;
- vengono annotati i dati di test manualmente, se è il primo esperimento che si esegue, oppure automaticamente grazie al programma **annotate.sh** se sono presenti già dati di test annotati;
- si impostano i parametri dell'algoritmo di *clustering*;
- viene effettuato il *training* dell'algoritmo, cioè vengono individuati *n clusters* all'interno dei dati di *training*, ciascuno descritto da un *context vector*;
- per ciascuna istanza di test si individua il vettore di bi-grammi del secondo ordine che la rappresenta;
- viene fatta la valutazione delle performance *esterna* utilizzando i *gold-senses* forniti con i dati di *test*.

Le parole utilizzate per i testi sono composte sia da verbi che da nomi e sono state scelte da una precedente competizione SemEval[5] e sono 21 in totale, scelte in base al numero di istanze di test presenti all'interno di JRC-Acquis:

- *side*: 3 significati
- *education*: 4 significati
- *mission*: 3 significati

[4]Elemento base su cui opera un analizzatore sintattico
[5]Semantic Evaluation, è una serie di competizioni di analisi semantica computazionale

- *paper*: 2 significati
- *plant*: 3 significati
- *post*: 3 significati
- *range*: 3 significati
- *rest*: 4 significati
- *ask*: 3 significati
- *effect*: 4 significati
- *exchange*: 4 significati
- *explain*: 3 significati
- *grant*: 2 significati
- *hold*: 5 significati
- *part*: 3 significati
- *position*: 3 significati
- *power*: 4 significati
- *rate*: 3 significati
- *see*: 3 significati
- *state*: 4 significati
- *work*: 3 significati

Nella figura 5.1 viene mostrato lo schema che descrive l'intero processo di esecuzione degli esperimenti.

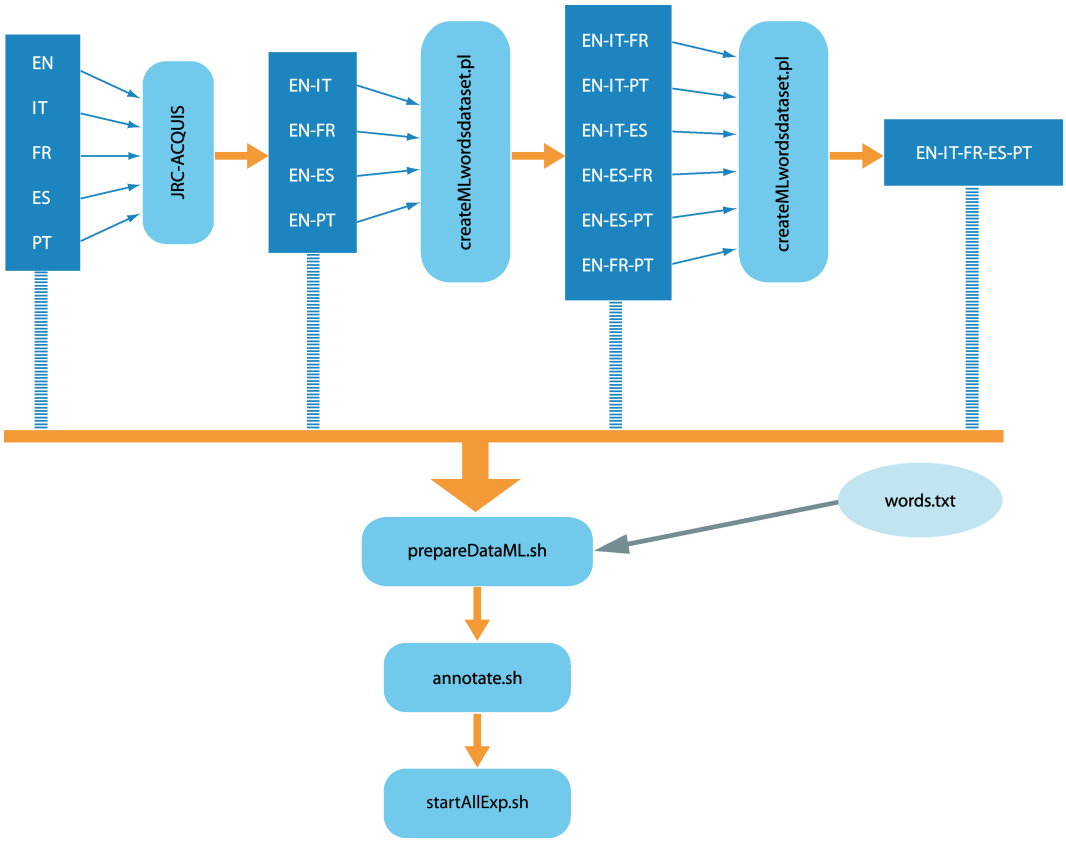

Figura 5.1: Processo di esecuzione degli esperimenti

5.6 Risultati

Nella tabella 5.6 e nella tabella 5.6 sono mostrati i risultati ottenuti per tutte le parole considerate, per ciascuna combinazione di lingue, sotto forma di F-Measure:

word	EN	EN-ES	EN-FR	EN-IT	EN-PT	EN-ES-FR
side	39.34	36.36	53.85	46.15	46.15	57.69
education	40	43.43	30.84	42.2	36.7	38.46
mission	59.62	46.15	51.28	45.95	39.44	53.85
paper	68.63	46.81	81.25	65.57	55.74	88.24
plant	51.54	44.02	57	57.75	53.77	58.25
post	57.63	47.37	35.44	41.98	44.87	35.9
range	50.93	39.42	40	54.55	51.66	44.3
rest	53.19	42.55	56.76	59.46	55.56	63.89
ask	44.12	38.89	44.74	43.24	37.33	60.53
effect	55.62	58.59	44.88	54.37	53.6	54.76
exchange	47.42	45.81	45.28	47.94	46.1	45.8
explain	50	51.61	56.67	50	41.94	45.16
grant	52.76	60.32	52.03	50.3	54.03	52.03
hold	38.46	37.21	39.53	38.46	40.48	39.53
part	44.54	45.09	38.6	40	42.29	40.94
position	52.08	54.29	51.43	45.09	49.21	50
power	42.24	36.96	46.67	44.83	57.14	45.05
rate	70.59	70.71	47.52	70.59	46.53	61.11
see	36.56	50	61.54	54	62.69	58.82
state	44	39.13	44.93	44	43.48	39.13
work	73.79	65.15	68.66	72.82	76.71	79.45

word	EN-ES-IT	EN-ES-PT	EN-FR-PT	EN-IT-FR	EN-IT-PT	EN-IT-FR-ES-PT
side	50	46.15	53.85	53.85	57.69	53.85
education	33.96	39.62	41.35	32.69	39.62	38.46
mission	55.56	53.85	42.42	27.45	44.44	58.97
paper	65.38	85.71	82.54	88.24	82.54	76.19
plant	56.73	53.85	56.37	58.17	57.62	56.52
post	45.57	42.31	39.74	32.91	40.51	42.31
range	44.44	50.62	44.3	44.16	45.75	55.35
rest	52.78	52.78	61.11	86.11	47.22	61.11
ask	65.79	63.16	60.53	36.84	63.16	81.58
effect	53.97	54.84	54.03	54.33	54.76	71.2
exchange	45.81	45.75	46.1	46.54	46.45	46.45
explain	45.16	45.16	46.67	45.16	45.16	45.16
grant	53.17	52.8	52.03	52.42	53.17	52.52
hold	37.21	37.21	44.19	44.19	44.19	39.53
part	41.62	50	43.93	40.94	43.18	43.93
position	51.43	57.14	52.86	52.86	49.3	57.14
power	45.65	47.25	43.82	45.05	45.05	44.44
rate	61.11	61.11	55.45	55.45	54.9	56
see	58.82	61.76	57.58	54.29	54.29	57.14
state	39.13	41.18	39.71	40.58	39.13	39.71
work	78.38	78.08	79.17	79.45	78.08	81

L'incremento di prestazioni per alcune parole raggiunge anche il 25% nei test eseguiti con determinate combinazioni di lingue, mantenendosi in linea con quanto scoperto in [2] [27] sebbene il miglioramento medio sia inferiore a questa cifra. I dati non sono semplici da interpretare e soprattutto le prestazioni dipendono in larga misura dalla specifica parola e dalle lingue utilizzate; inoltre in alcuni dei test multi-lingua si ottiene un punteggio di *F-Measure* inferiore a quello ottenuto con il testo mono-lingua. Tuttavia vi è una chiara tendenza positiva dell'andamento delle performance all'aumentare del numero di lingue utilizzate, come si può meglio notare osservando i grafici seguenti.

Nella figura 5.2 è mostrata la rappresentazione grafica dei risultati presenti in tabella: sull'asse delle ascisse ci sono le parole scelte per i test, mentre sull'asse delle ordinate troviamo il valore di *F-Measure* ottenuto nella valutazione. Ciascuna linea rappresenta un esperimento distinto.

Una visione più chiara dei risultati è mostrata in figura 5.4 dove si può notare come l'area in rosso, corrispondente all'esperimento eseguito con la sola lingua

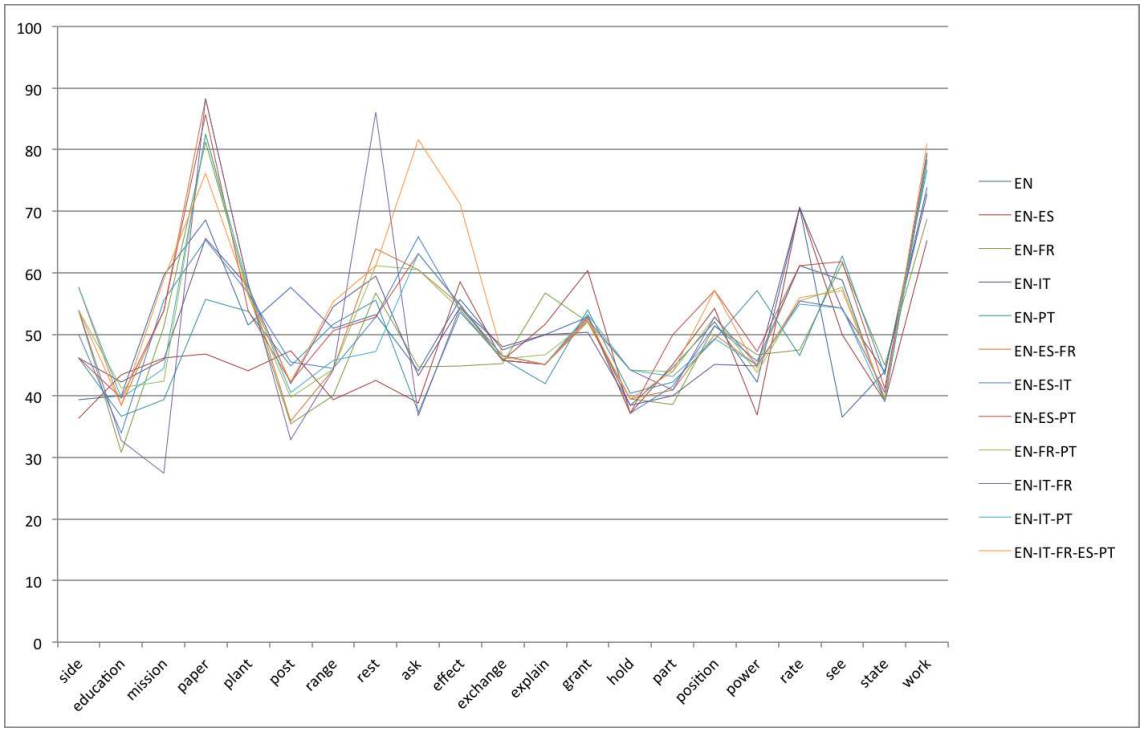

Figura 5.2: Rappresentazione grafica dei risultati dei test. Ciascuna linea mostra le performance in termini di *F-Measure* di un singolo esperimento eseguito con una data combinazione di lingue

inglese, riesca a superare le prestazioni degli esperimenti multi-lingua per solo due delle parole considerate.

Nella figura 5.5 è invece rappresentata la *F-Measure* media degli esperimenti eseguiti con *features* bi-lingua e degli esperimenti eseguiti con *features* tri-lingua confrontata con le performance dei test eseguiti con la sola lingua inglese: l'approccio mono-lingua ottiene performance inferiori in 16 delle 21 parole.

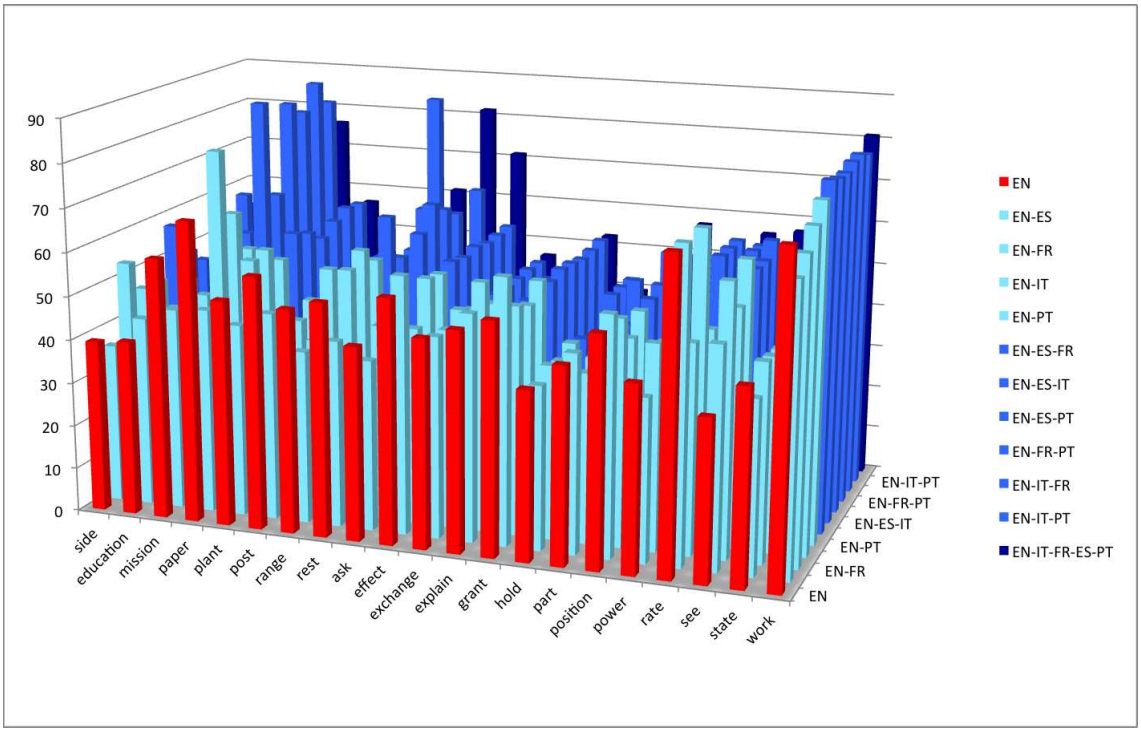

Figura 5.3: Rappresentazione 3D dei risultati degli esperimenti. In grigio è rappresentato il test mono-lingua, in celeste i test bi-lingua, in blu chiaro i test tri-lingua e in blu scuro il test a cinque lingue

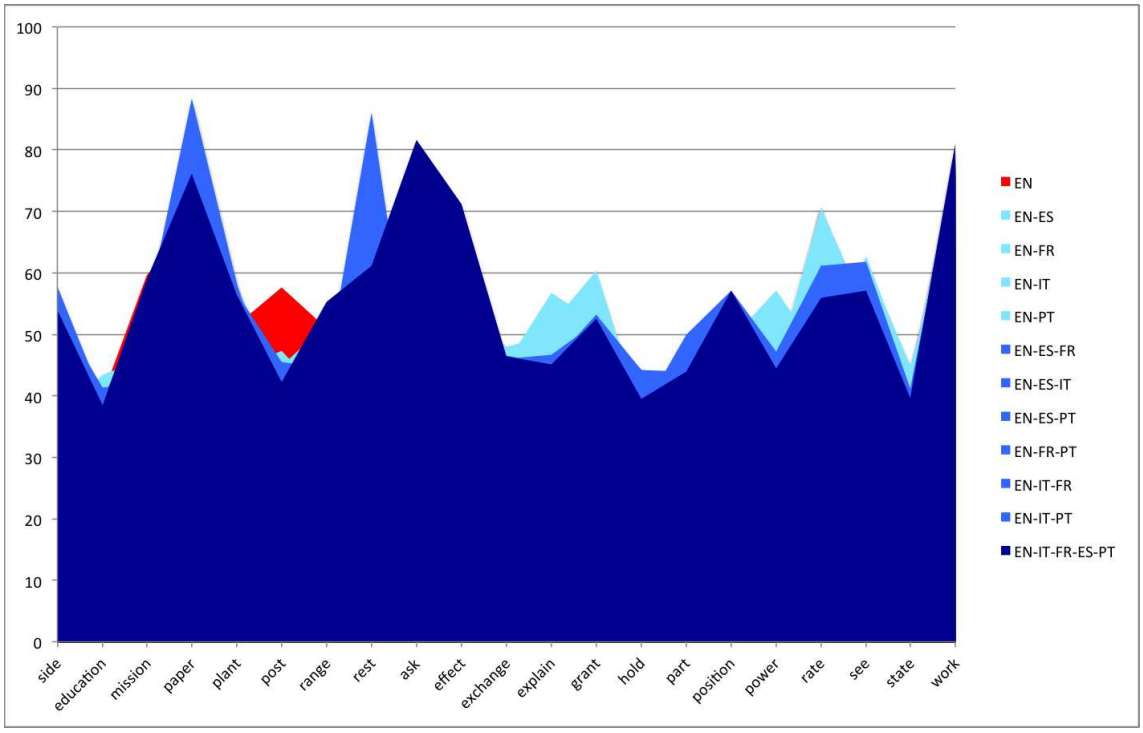

Figura 5.4: Rappresentazione filled-area dei risultati. In rosso è rappresentato il test mono-lingua, in celeste i test bi-lingua, in blu chiaro i test tri-lingua e in blu scuro il test a cinque lingue

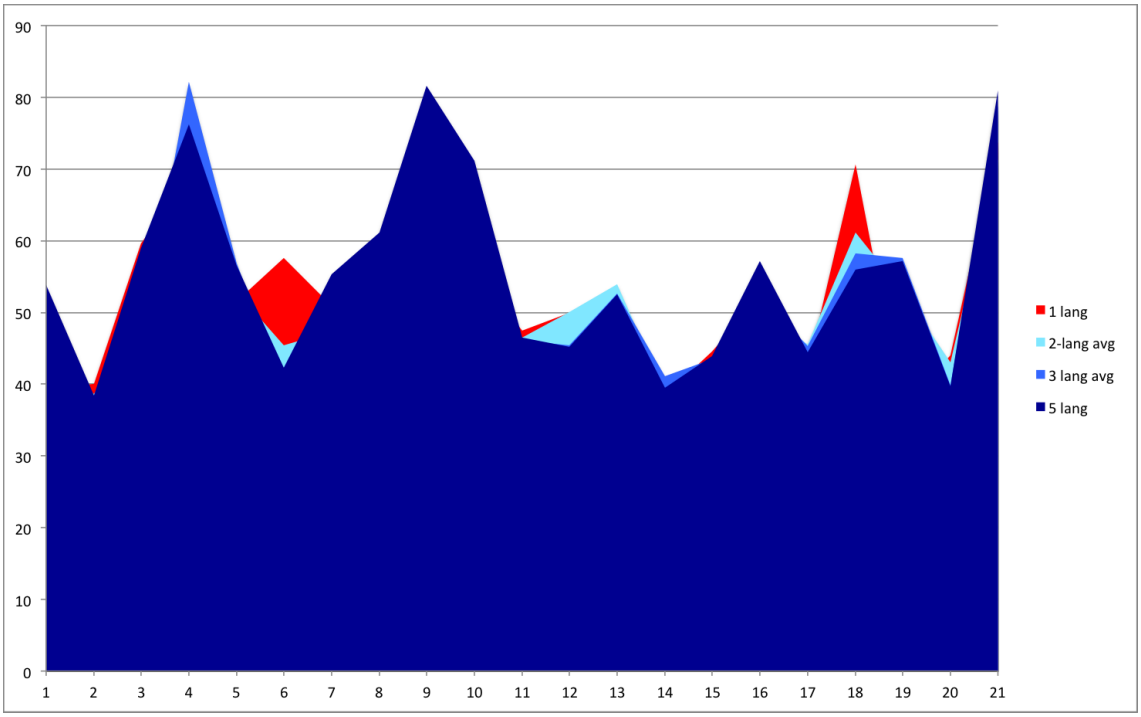

Figura 5.5: Rappresentazione della media delle performance per numero di lingue. In rosso è rappresentato il test mono-lingua, in celeste i test bi-lingua, in blu chiaro i test tri-lingua e in blu scuro il test a cinque lingue

Conclusioni

Si è osservato come l'uso di *features* multi-lingua all'interno di *task* di *Word Sense Induction* comporti dei vantaggi reali in termini di prestazioni.

In alcuni test è stato raggiunto addirittura un miglioramento del 25% nella F-Measure rispetto al caso mono-lingua. Tuttavia l'incremento medio di performance si attesta intorno al 5-10%. In alcuni casi, con alcune combinazioni multi-lingua si è comunque osservato un calo di prestazioni dell'algoritmo, sebbene questi siano dei casi isolati, e siano per lo più relegati alle parole per cui il sistema offre prestazioni peggiori in tutti gli esperimenti eseguiti.

I risultati sono molto vari ma è comunque possibile scorgere una tendenza precisa: le prestazioni dell'algoritmo di *Word Sense Induction* sono direttamente proporzionali al numero di lingue utilizzate per la rappresentazione delle *features*.

Si deve tenere presente che il *corpus* utilizzato è di tipo *domain specific* e i risultati ottenuti potrebbero non rispecchiare i risultati ottenibili con un *dataset* generale. JRC-Acquis è stato scelto per la grande quantità di documenti forniti e la varietà di lingue supportate, nonchè per la sua disponibilità *offline* e il suo costo (gratuito), ma si potrebbe pensare di utilizzare un sistema di *Machine Translation* (come ad esempio Google Translate) applicato a testo proveniente da sorgenti diverse, in modo da produrre testo parallelo multi-lingua generale. Tale possibilità non è stata considerata per lo svolgimento di questo lavoro di Tesi a causa del numero limitato di traduzioni effettuabili gratuitamente che questi sistemi di *Machine Translation* consentono.

Sebbene sia possibile affermare che per le 20 parole considerate, l'approccio multi-lingua funzioni meglio dell'approccio mono-lingua, è difficile stabilire quali combinazioni di lingue offrano le migliori performance. Infatti dall'analisi dei dati si può notare come i risultati varino moltissimo a seconda della *target word* e delle lingue utilizzate. Negli esperimenti sono state considerate le quattro lingue con il

maggior numero di documenti disponibile, ma si potrebbe pensare di introdurre ulteriori lingue e di conseguenza ulteriori esperimenti ed incrementare il numero di parole considerate, al fine di verificare se il *trend* positivo delle performance al crescere del numero di lingue utilizzate viene confermato.

Alcune combinazioni di lingue, per una data parola, offrono prestazioni di molto inferiori alla media delle altre sulla stessa parola. Questo fa pensare ad un possibile sviluppo dell'algoritmo che vada ad introdurre la stessa idea utilizzata negli *ensemble methods* del *machine learning*: si potrebbe impostare un sistema di votazione, dove ciascun algoritmo applicato con una data combinazione di lingue ha diritto a votare per il senso corretto. In questo modo si potrebbe verificare se si è in grado di eliminare gli errori isolati con conseguente guadagno in termini di prestazioni.

Bibliografia

[1] *Thesaurus EUROVOC - Volume 2: Sub- ject-Oriented Version. Ed. 3/English Language*. 1995.

[2] C. Banea and R. Mihalcea. Word sense disambiguation with multilingual features. *International Conference on Semantic Computing (IWCS 2011), Oxford, UK*, January 2011.

[3] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Building an integrated ontology within sewasie system. *proceedings of the First International Workshop on Semantic Web and Databases (SWDB), Co-located with VLDB 2003 Berlin, Germany*, September 2003.

[4] S. Bergamaschi, D. Beneventano, L. Po, and S. Sorrentino. Automatic normalization and annotation for discovering semantic mappings. *SeCO Workshop on Search Computing, Como/Milan, Italy*, May 2011.

[5] S. Bergamaschi and et al. A semantic approach to information integration: the momis project. *Sesto Convegno della Associazione Italiana per l'Intelligenza Artificiale, AI*IA 98, Padova IT*, September 1998.

[6] S. Bergamaschi, L. Po, S. Sorrentino, and A. Corni. Dealing with uncertainty in lexical annotation. *28th International Conference on Conceptual Modeling, (Demo Session) ERPD 2009, Gramado, Brasil*, November 2009.

[7] S. Bergamaschi and M. Vincini. A semantic approach to access heterogeneous data sources: the sewasie project. *Invited talk at TELEBALT Conference, Teleworking for Business, Education, Research and e-Commerce, Vilnius, Lithuania*, October 2002.

[8] J. Chai and A. Biermann. The use of word sense disambiguation in an information extraction system. *Proceedings of Sixteenth National Conference in Artificial Intelligence and Eleventh Annual Conference on Innovative Applications of Artificial Intelligence*, July 1999.

[9] Y.S. Chan, H.T. Ng, and Z. Zhong. Nus-pt: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval2007), Prague, Czech Republic*, 2007.

[10] T. Cohen and D. Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of Biomedical Informatics*, 2009.

[11] W. Gale and K. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics, 19(1):75-102*, 1993.

[12] Schütze H. Automatic word sense discrimination. *Computational Linguistics*, 1998.

[13] Z. Harris. Distributional structure. 1954.

[14] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: The 90% solution. *Companion Volume to the Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, New York*, 2006.

[15] N. Ide, T. Erjavec, and D. Tufis. Sense discrimination with parallel corpora. *Proceedings of ACL-02 Workshop on WSD: Recent Success and Future Directions, Philadelphia, USA*, 2002.

[16] I. Klapaftis and M. Suresh. Uoy: A hypergraph model for word sense induction and disambiguation. *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007.

[17] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 1997.

[18] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceedings of the 5th annual international conference on Systems documentation, New York, USA*, 1986.

[19] D. Lin. Automatic retrieval and clustering of similar words. *Proceedings of the 17th International Conference on Computational linguistics (COLING, Montreal, P.Q., Canada)*, 1998.

[20] D. Lin and P. Pantel. Discovering word senses from text. *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2002.

[21] S. Manandhar, I.P. Klapaftis, D. Dligach, and S.S. Pradhan. Semeval-2010 task 14: Word sense induction & disambiguation. *Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden*, 2010.

[22] D. Martinez. Supervised word sense disambiguation: Facing current challenges. *Ph.D. Thesis. University of the Basque Country, Spain*, 2004.

[23] Miller, Beckwith, and et al. Wordnet: an online lexical database. *International Journal of Lexicography*, 1999.

[24] R. Navigli. Meaningful clustering of senses helps boost word sense disambiguation performance. *Proceedings of 21st Annual Meeting of the Association for Computational Linguistics (COLING-ACL), Sidney, Australia*, 2009.

[25] R. Navigli and M. Lapata. An experimental study on graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[26] H.T. NG, B. Wang, and Y.S. Chan. Exploiting parallel texts for word sense disambiguation: an empirical study. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan*, 2003.

[27] E. Ordonez, R. Mihalcea, and S. Hassan. Unsupervised word sense disambiguation with multilingual representations. *Proceedings of the Conference on Language Resources and Evaluations (LREC 2012), Istanbul, Turkey*, May 2012.

[28] S. Patwardhan, S. Banerjee, and T. Pedersem. Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, 2003.

[29] T. Pedersen and A. Kulkarni. Automatic cluster stopping with criterion functions and the gap statistic. *Proceedings of the Demonstration Session of the Human Language Technology Conference and the Sixth Annual Meeting of the North American Chapter of the Association for Computational Linguistics, pp. 276-279, New York City*, 2006.

[30] L. Po and S. Sorrentino. Automatic generation of probabilistic relationships for improving schema matching. *Information Systems, Volume 36, Issue 2 (2011), pp. 192-208*, 2011.

[31] S.P. Ponzetto and R. Navigli. Knowledge-rich word sense disambiguation rivaling supervised system. *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics (ACL), Uppsala, Sweden*, 2010.

[32] Amruta Purandare and Ted Pedersen. Senseclusters - finding clusters that represent word senses. *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, May 2004.

[33] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic*, 2007.

[34] H. Schütze. Dimensions of meaning. *Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing. IEEE Computer Society Press, Los Alamitos, CA. 787–796*, 1992.

[35] S. Sorrentino, S. Bergamaschi, and M. Gawinecki. Norms: An automatic tool to perform schema label normalization. *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011. Hannover, Germany*, April 2011.

[36] S. Sorrentino, S. Bergamaschi, M. Gawinecki, and L. Po. Schema label normalization for improving schema matching, data & knowledge engineering.

in Data & Knowledge Engineering, Vol. 69, Issue 12 (2010), pp. 1254-1273, 2010.

[37] S. Sorrentino, S. Bergamaschi, and E. Parmiggiani. A supervised method for lexical annotation of schema labels based on wikipedia. *ER 2012 - 31st International Conference on Conceptual Modeling (ER 2012) - Florence, Italy*, October 2012.

[38] R. Steinberger and et al. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006). Genoa, Italy*, May 2006.

[39] R Tibshirani, G. Walther, and T. Hastie. Esti- mating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistics Society (Series B), pages 411–423*, 2001.

[40] O. Uzuner and B. Katz. Word sense disambiguation for information retrieval. *Proceedings of AAAI/IAAI1999*, July 1999.

[41] D. Varga, L. Németh, and et al. Parallel corpora for medium density languages. *Proceedings of the RANLP 2005, pages 590-596*, 2005.

[42] Vickrey, Biewald, Teyssier, and Koller. Word-sense disambiguation for machine translation. *HLT/EMNLP*, 2005.

Elenco delle figure