Estrazione di news da canali RSS: un approccio basato sull'annotazione semantica

Ilaria Clara Urciuoli

12 aprile 2012

A Maria, Marco e Fabio
alle loro individualità
al nostro esser famiglia

A Daniele
che ha trasformato questa difficoltà in gioia

Riassunto

Questa tesi nasce con l'obiettivo di esplorare l'utilizzo dell'annotazione semantica di testi brevi nella ricerca di news inserite in canali RSS. Alla base dell'annotazione c'è la conoscenza messa a disposizione da Wikipedia, sotto forma di pagine dell'enciclopedia e di link tra esse.

Viene qui proposto un nuovo approccio che utilizza l'annotatore TAGME, sviluppato presso il Dipartimento di Informatica della nostra Università, per individuare e annotare concetti rilevanti contenuti in una news (in questo caso nel suo titolo e nella sua descrizione), e una misura di similarità tra il concetto cercato e quelli individuati nelle news in input.

L'efficacia dell'approccio proposto è stata valutata su un corpus appositamente creato e consistente di 4195 news estratte da quattro feed e pubblicate dal 2 al 14 settembre 2011. I feed sono quelli relativi a due quotidiani (*Corriere della Sera* e *La Repubblica*), un'agenzia di stampa (*ANSA*) e un sito specialistico (*Punto Informatico*).
Sono state poi create 30 query costituite ognuna da un concetto e dalla forma testuale comunemente utilizzata per esprimerlo. Ogni news del corpus è stata poi annotata, specificando per quali delle 30 query essa risulta rilevante.

I risultati ottenuti dal nostro approccio sono stati confrontati con quelli ottenuti da altri due algoritmi che rappresentano lo scenario di riferimento per i sistemi moderni di ricerca/alerting sulle/di news. In generale l'approccio basato su TAGME ottiene una misura F1 pari a 60,9%, circa 2 punti migliore di quella ottenibile dagli altri due approcci sperimentati.

Indice

Capitolo 1

INTRODUZIONE

Nel panorama attuale l'utilizzo della rete rappresenta ormai il principale mezzo per l'acquisizione di informazioni: se fino a qualche anno fa i canali tradizionali attraverso i quali rimanere aggiornati erano principalmente i quotidiani acquistati in edicola e i telegiornali oggi il cittadino ha imparato a sfruttare al meglio i nuovi mezzi a disposizione, assumendo un ruolo attivo nella selezione e anche nella diffusione delle notizie. Questo passaggio ha portato allo sviluppo di nuovi modi di codificare l'informazione, tesi a renderne più semplice la circolazione: sono così nati i feed RSS, in cui sia il contenuto della notizia che i dati relativi all'articolo (come l'autore, il giorno di pubblicazione o il link al testo completo) sono inseriti in file XML che, fornendone una struttura, favoriscono l'elaborazione di questo insieme di informazioni.

Tuttavia, come spesso accade quando si ha a che fare con una novità interessante, si arriva rapidamente ad utilizzarla in maniera massiccia. La quantità di dati pubblicati rende quindi evidente la necessità di creare algoritmi di ricerca efficienti ed efficaci. L'XML rende semplice il recupero di notizie a partire dalle meta-informazioni associate all'articolo: potremmo infatti senza difficoltà recuperare tutti gli scritti pubblicati in un dato arco temporale da un giornalista di un particolare giornale. L'uso di sistemi di alert e parallelamente di filtri e motori di ricerca per RSS conferma l'interesse per un'estrazione dell'informazione basata sul contenuto oltre che sulla struttura.

A oggi, comunque, gli approcci utilizzati per questo task si affidano a ricerche testuali che spesso si rivelano poco efficaci in quanto ancorate a due caratteristiche del linguaggio naturale: l'ambiguità e la possibilità di esprimere un concetto attraverso molteplici espressioni. Tutto ciò rende la formulazione efficace di una query quanto mai difficile e, anche, limitante, in quanto spesso si ha una copertura parziale delle informazioni che interessano a meno di non addentrarsi nella progettazione di espressioni regolari quanto mai complesse, e comunque non alla portata della stragrande maggioranza

di utilizzatori del mezzo.

In questa tesi viene proposto un approccio diverso che si basa sull'utilizzo di TAGME, un software sviluppato presso l'A^3 lab del Dipartimento di Informatica dell'Università di Pisa, che individua ed annota i concetti rilevanti presenti in un testo breve, quale appunto quello di una news. Il modello di ricerca progettato e sperimentato in questa tesi abbandona il piano puramente testuale per spostarsi su quello "semantico": il testo relativo al contenuto delle singole news (che nei file RSS è codificato usando i tag `title` e `description`) viene trasformato mediante TAGME in un insieme di entità e loro sensi, rappresentati da link alle pagine di Wikipedia che li descrivono. In maniera analoga anche la query verrà disambiguata e associata a uno o più link verso pagine di Wikipedia. A questo punto verrà adottata una misura di similarità tra i concetti estratti dalla notizia e i concetti cercati per determinare le news che sono pertinenti con la ricerca effettuata dall'utente. Questa misura di similarità, come anche il modello costruito per la disambiguazione, si appoggerà a una base di conoscenza, che nel caso specifico di TAGME è Wikipedia. La scelta, ben affermata nella letteratura scientifica, offre un buon trade-off tra ampiezza delle informazioni disponibili, buona strutturazione, e ragionevole frequenza di aggiornamento di questi. Questo approccio verrà poi sperimentato su un corpus costruito a partire dai feed pubblicati da *La Repubblica*, il *Corriere della Sera*, l'*ANSA* e *Punto Informatico* nei giorni tra il 2 e il 14 settembre. Ogni news è stata quindi annotata indicando per quale delle 30 query selezionate fosse interessante. I risultati ottenuti mostrano una F1 che sfiora il 61% e una Recall del 51% e una Precision del 75.4% mostrando un miglioramento di 2 punti percentuali sull'F1 del più efficente tra i competitor.

Nei capitoli che seguono verranno mostrati i feed, l'idea che li ha portati ad avere il successo che attualmente hanno, come sono strutturati e i tipi di applicazione sviluppati per essi. Quindi si parlerà di TAGME, mostrandone il funzionamento e commentando le sue prestazioni in efficienza ed efficacia nell'annotazione semantica di testi brevi. Conclusa questa parte si esaminerà il modello di ricerca oggetto di questa tesi, si descriverà il corpus creato e manualmente annotato per valutare l'algoritmo, e si confronterà questo con quelli che basano la ricerca esclusivamente sul match testuale. Si riporteranno quindi i risultati ottenuti fornendone una interpretazione. Infine riportiamo le conclusioni con i possibili scenari di utilizzo.

Capitolo 2

OBIETTIVI E CONTESTO

L'obiettivo di questa tesi è sperimentare e valutare un utilizzo dell'annotazione "semantica" per l'estrazione di news a partire da feed.

L'abbondanza di dati disponibili ha reso auspicabile un nuovo modo di esplorare il web, che superi l'ambiguità del linguaggio naturale e ne esplori invece il contenuto. Allo stesso tempo la diffusione del formato RSS offre nuove possibilità: le informazioni in esso riportate sono infatti organizzate in una struttura facile da processare per un elaboratore. Unire questo aspetto strutturale all'analisi semantica del testo è quindi un modo per tentare nuovi approcci alla ricerca.

Nei paragrafi che seguono si introdurrà il formato RSS e verrà mostrato il funzionamento di TAGME, software qui utilizzato per l'annotazione semantica dei testi.

2.1 Il formato RSS

L'RSS o Really Simple Syndication è un formato utilizzato per la distribuzione dei contenuti Web. Si basa sull'XML (eXtended Markup Language), dal quale eredita la struttura.
Negli ultimi anni si è assistito al diffondersi di questo tipo di formato soprattutto nei siti che frequentemente aggiornano i loro contenuti, come giornali on-line e blog. La ragione di questo successo sta nel fatto che le informazioni così strutturate risultano comode per l'utente che deve solamente inserire, in un cosiddetto aggregatore, i link ai feed di suo interesse per essere continuamente raggiunto dalle informazioni che le fonti selezionate propongono.

La storia dell'RSS è complessa, come dimostra la presenza di ben tre forme per lo stesso acronimo. Questo ha infatti rappresentato le seguenti diciture:

- RDF Site Summary (RSS 0.9 e RSS 1.0)
- Rich Site Summary (RSS 0.91)
- Really Simple Syndication (RSS 2.0)

La sigla RSS nasce inizialmente come acronimo di RDF Site Summary: la prima versione, la 0.9, era utilizzata per mostrare sul portale di My Netscape una breve descrizione (seguita dal link) delle notizie pubblicate su altri siti. Tale versione risale al marzo 1999 e già allora ebbe una vasta diffusione tra coloro che fornivano contenuti, molti dei quali aderirono all'iniziativa.
La complessità derivante dall'essere conforme allo standard RDF (Resource Description Framework) ha portato in pochi mesi allo sviluppo dell'RSS 0.91 da parte di Dave Winer. Questa è una versione semplificata del 0.9, arricchita nei tag ma slegata dal RDF. Resta l'acronimo che però qui indica Rich Site Summary.
L'anno successivo O'Reilly pubblicò la versione 1.0 curata dal RSS-DEV Working Group [13] e nuovamente conforme all'RDF.
Fu ancora Winer a introdurre l'ultima dicitura per RSS: Really Seample Syndication. A questa si fa riferimento quando si parla di RSS 2.0 e ad oggi rappresenta la versione più utilizzata in rete.

Vista la sua diffusione, nel testo si indicherà con RSS la versione 2.0.

Struttura del file

Basandosi sull'XML ogni file RSS sarà costituito da un insieme di tag innestati e senza possibilità di overlap, tutti racchiusi in un unico elemento. L'elemento `rss` è quello più esterno, e contiene un `channel` utilizzato per la descrizione della fonte del feed. Queste informazioni sono racchiuse in tag obbligatori come `title`, `decription` e `link` (in cui vengono riportati

rispettivamente il nome, una descrizione e il collegamento ipertestuale alla fonte) e in alcuni tag opzionali come `language` (che contiene indicazioni sulla lingua utilizzata), `webMaster` (per inserire la mail del responsabile), `pubDate` (per la data di pubblicazione) e altri. L'elemento `channel` dovrà inoltre contenere uno o più tag `item`: è attraverso questo e i tag in esso racchiusi che viene rappresentata l'informazione. I campi obbligatori di `item` sono `title`, che racchiude il titolo, `link`, per indicare l'URL della pagina web, e `description`, per un riassunto del contenuto; ne esistono inoltre alcuni opzionali come `pubDate`, per la data di pubblicazione, `guid` in cui inserire una stringa identificativa dell'item ecc.

Segue un esempio di file RSS, o flusso, estratto dall'ANSA.it. Nell'esempio per brevità è riportato un solo item.

```
<rss xmlns:atom=‘‘http://www.w3.org/2005/Atom’’ version=‘‘2.0’’>
   <channel>
      <title>ANSA.it</title>
      <link>http://www.ansa.it</link>
      <description>Updated every day</description>
      <language>it</language>
      <copyright> Copyright:  (C) ANSA,
      http://www.ansa.it/web/static/disclaimer.html</copyright>
      <item>
         <title>
            Google, nuove regole privacy Altola’ Ue
         </title>
         <description>
            All’ANSA la spiegazione integrale di Big G
         </description>
         <link>
            http://www.ansa.it/web/notizie/rubriche/tecnologia/
            2012/02/28/visualizza_new.html_105303799.html
         </link>
         <pubDate>28 Feb 2012 20:14:00 +0100</pubDate>
         <guid>
            http://www.ansa.it/web/notizie/rubriche/tecnologia/
            2012/02/28/visualizza_new.html_105303799.html
         </guid>
      </item>
   </channel>
</rss>
```

Aggregatori, filtri e motori di ricerca

Parlando delle origini degli RSS abbiamo accennato all'idea che ha por-

tato alla sua creazione, quella di riunire sul portale My Netscape le notizie pubblicate da altre fonti. Tale idea è stata sviluppata con i feed (o RSS) reader, ossia applicazioni Web o software che riuniscono in un unico spazio gli RSS provenienti dai diversi siti che l'utente decide di seguire. Risulta evidente il vantaggio che ne deriva dall'utilizzo: per essere informati sugli aggiornamenti offerti da un sito di interesse non è più necessario cercare attivamente le novità; basta invece fare una sottoscrizione al feed proveniente da quella fonte e il programma andrà automaticamente e a intervalli regolari alla ricerca degli aggiornamenti. Una volta fatto il download del flusso provvederà a visualizzare le novità sfruttando i tag su indicati.

Quest'abbondanza di informazioni ha portato alla necessità di creare filtri e motori di ricerca operanti sui flussi in modo da lasciare la possibilità all'utente di ricevere feed relativi solo alle informazioni cui è interessato. Avere a disposizione un motore che selezioni in maniera efficace le news assume quindi un'importanza sempre maggiore.

Un esempio di filtro per RSS è dato da FeedRinse (http://feedrinse.com). Funziona previo log in: una volta effetuato l'accesso viene chiesto di inserire un feed da seguire e quindi offre la possibilità di filtrare.

L'immagine 2.2 mostra parte delle opzioni disponibili:

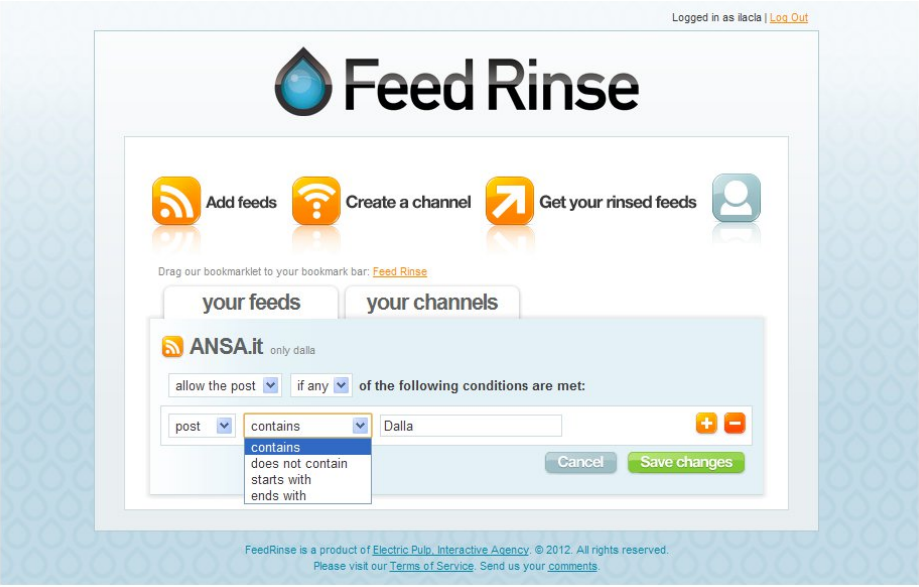

Figura 2.1: Screenshot del filtro fornito da FeedRinse

FeedRinse offre la possibilità di nascondere o mostrare gli item che soddisfano una o tutte le caratteristiche impostate: l'utente può scegliere se il testo che inserisce deve essere contenuto, non contenuto, essere prefisso o suffisso dell'intero item o del titolo, del corpo o dell'autore.

Si noti che il filtro in questione si basa esclusivamente su match di parole come risulta evidente nell'esempio che segue.

Ispirandoci a eventi poco lieti del panorama musicale italiano abbiamo cercato di ottenere informazioni sul cantante Dalla. L'evidente ambiguità di questo cognome crea non poche difficoltà al filtro che valuta come positivi non solo tutti gli item che parlano del cantante ma anche tutti quelli che vedono la presenza della preposizione articolata o del verbo seguito da pronome accusativo atono.

Come si vede nella figura 2.2 infatti le risposte, ottenure filtrando i feed dell'ANSA in modo da ottenere quelli contenenti "Dalla", sono poco pertinenti all'argomento cercato. I risultati sono stati qui visualizzati attraverso Google Reader.

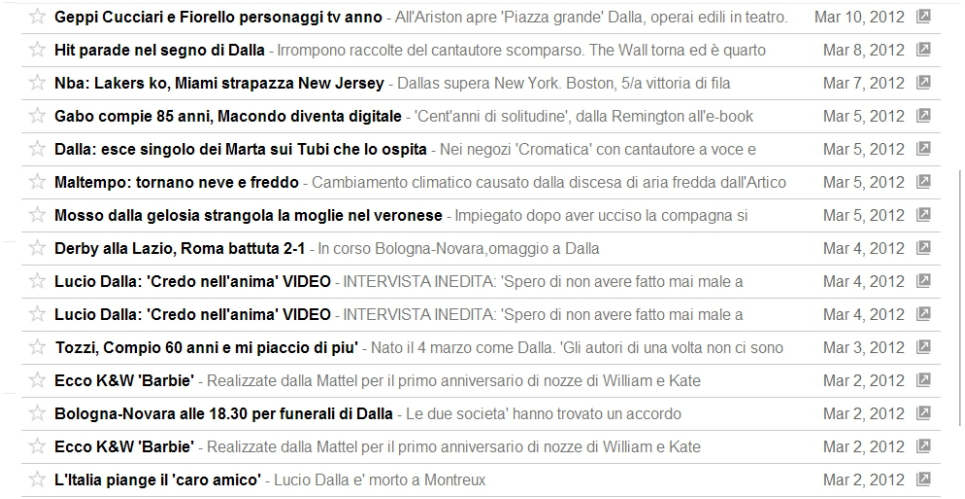

Figura 2.2: Screenshot degli RSS dell'ANSA dopo essere stati filtrati da FeedRinse

Un modo per tentare di superare questo limite evidente è utilizzare un disambiguatore che lavori tanto sulla query quanto sul contenuto degli RSS. Di seguito viene mostrato il funzionamento di uno di questi software.

2.2 TagMe

TAGME è "il primo software che, on-the-fly e con alte performance in termini di precision e recall, annota testi brevi con i link pertinenti alle pagine di Wikipedia."

Questa definizione contenuta in [5] mette in rilievo alcuni aspetti fondamentali che rendono lo strumento largamente utilizzabile nell'ambito dell'esplorazione della semantica dei testi. In questo capitolo fornirò una descrizione della sua architettura concentrandomi su quegli aspetti che sono fondamentali per comprendere il lavoro svolto durante la tesi.

In base a quanto affermato dagli autori, la funzione di TAGME è quella di aggiungere link pertinenti al testo in input. L'inserimento di tale informazione permette il collegamento tra la parola (forma attraverso la quale l'uomo comunica) e il suo contenuto (il concetto che costituisce parte del messaggio per il quale è avvenuta la comunicazione). Ciò è possibile grazie all'utilizzo di Wikipedia come base di conoscenza, le cui pagine con i relativi link, rappresentano l'insieme dei concetti del mondo e il modo in cui essi sono in relazione tra loro.

Vale la pena osservare che TAGME lavora con testi brevi e non specialistici. Questa caratteristica, che lo differenzia da tutti i suoi predecessori, rende possibile l'applicazione del software all'analisi di testi attualmente molto diffusi nel web, ad esempio snippet di motori di ricerca, tweet o, come nel caso qui proposto, feed RSS. I testi contenuti in questo tipo di file infatti sono molto brevi e il più delle volte non settoriali. Come sarà evidente osservando la struttura dell'applicazione, entrambe queste caratteristiche comportano una maggiore difficoltà dovuta alla scarsità e alla generalità dei dati da elaborare.

È stato già accennato che Wikipedia costituisce la base di conoscenza alla quale il software fa riferimento. Le ragioni di tale scelta (largamente condivisa nella comunità scientifica che affronta queste problematiche) risiedono principalmente in una caratteristica dell'enciclopedia, quella di essere un raccoglitore di informazioni inserite dal basso, al quale cioè tutti collaborano rendendone quindi veloce la crescita; ciò le permette di rappresentare un buon compromesso tra quantità e qualità di dati. Se si osservano le altre fonti si nota la disponibilità di una vastissima quantità di dati che tuttavia non presenta alcuna struttura utilizzabile (come l'intero Web) o al contrario di risorse che sono qualitativamente impeccabili ma che hanno una copertura molto limitata (di cui un esempio è la rete dei synset di WordNet).

L'anatomia di TAGME

Il funzionamento di TAGME è legato alla presenza di 4 diverse informazioni:

- un insieme di concetti;

- un dizionario delle ancore (o spot);
- le relazioni tra ancore e concetti;
- le relazioni tra concetti.

Esaminiamo nel dettaglio ognuna di queste componenti.

Con il termine concetto si fa riferimento al senso non ambiguo che costituisce l'elemento base dell'annotazione del testo. Nell'architettura di TAGME queste sono le pagine di Wikipedia.

Le ancore o spot sono tutte quelle sequenze di parole che sono associate a uno o più sensi (concetti o entità o pagine di Wikipedia). Vengono inserite nel dizionario tutte le espressioni usate in Wikipedia come ancora (termine che indica la porzione di testo che rappresenta il link in una pagina) e i titoli (inclusi quelli di reindirizzamento) purché abbiano una frequenza maggiore di 2 e compaiano come link almeno nell'1% dei casi.

Con relazioni tra ancore e concetti si intende un grafo bipartito in cui i vertici sono ancore e concetti. Esiste un arco $a \to p$ se l'ancora a punta alla pagina p. Essendo il linguaggio naturale ambiguo un'ancora a può avere diversi significati e quindi un link verso più pagine $p \in Pg(a)$. Analogamente molto probabilmente una pagina p avrà diverse ancore a, ognuna con una sua probabilità ($Pr(p|a)$) di fare riferimento a p.

L'ultima componente elencata è rappresentata da un digrafo (o grafo diretto) in cui i nodi sono le pagine di Wikipedia (i concetti) ed esiste un arco da p_1 a p_2 se p_1 contiene un link in uscita verso p_2.

Si può dividere il processo per giungere all'annotazione in tre sottotask. Questi sono:

- identificazione delle ancore nel testo (anchor parsing);
- disambiguazione delle ancore (anchor disambiguation);
- selezione delle ancore (anchor pruning).

Anchor parsing

Per risolvere il primo compito vengono cercate tutte le sottosequenze del testo in input che sono presenti nel dizionario delle ancore. Può succedere che due ancore si accavallino (overlap) come nel caso del testo "premio Nobel": qui infatti riconosciamo due possibili spot (a_1=`premio Nobel`, a_2=`Nobel`). Il comportamento di TAGME è il seguete: se a_1 ha una probabilità a priori di essere un link maggiore di quella di a_2 ($lp(a_1) > lp(a_1)$) allora vengono mantenute entrambe le ancore altrimenti, come in questo caso, viene scartata a_2.

Anchor disambiguation

Una volta ottenuto l'elenco di tutte le possibili ancore del testo, TAGME procede alla loro disambiguazione utilizzando uno schema di voti per cui all'ancora a viene associato il senso p_a tra tutti quelli in $Pg(a)$ perché è quello che ha una relazione più stretta (o è semanticamente più vicino) a tutti i possibili sensi non ancora disambiguati delle restanti ancore del testo. Per definire il grado di relazione tra due concetti viene utilizza la formula proposta in [2] e ispirata alla Google Similarity Distance ([3]).

$$dist(p_a, p_b) = \frac{\log(\max(|\ in(p_a)\ |, |\ in(p_b)\ |)) - \log(|\ in(p_a) \cap in(p_b))}{\log(W) - \log(\min(|\ in(p_a)\ |, |\ in(p_b)\ |))} \quad (2.1)$$

Qui con *in(p)* si indica il numero di link in entrata della pagina p e con W il numero di pagine di Wikipedia. A partire da questa distanza calcoliamo la relatedness nel seguente modo:

$$rel(p_a, p_b) = \begin{cases} 1 - dist(p_a, p_b) & \text{se } dist(p_a, p_b) > 1 \\ 0 & \text{altrimenti} \end{cases} \quad (2.2)$$

Nell'esempio riportato in figura 2.3 le pagine che puntano a P_a sono 4, due delle quali puntano anche a P_b che ha però 11 link in ingresso. In questo caso dunque $dist(p_a, p_b) = \log(11) - \log(2)/\log(W) - \log(4)$

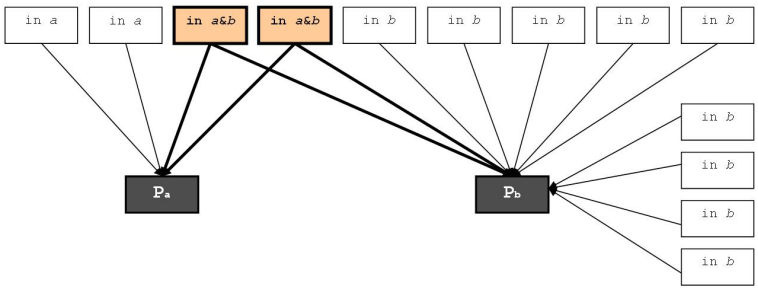

Figura 2.3: Ogni rettangolo rappresenta una pagina in Wikipedia: i rettangoli colorati in grigio fanno riferimento ai sensi di cui si vuole calcolare la relatedness. Gli altri sono le pagine contenenti un link a una delle 2 (rettangoli bianchi) o a entrambi (rettangoli arancione).

Da quanto riportato emerge che due concetti sono tanto più in relazione tra loro quanto maggiore è il numero di pagine che puntano ad entrambi.

Una volta calcolata la relatedness tra i vari sensi si può calcolare il voto che ogni ancora del testo da ai vari sensi dell'ancora che si sta esaminando. Supponendo di voler disambiguare l'ancora a calcoliamo per ogni senso $p_a \in Pg(a)$ e per ogni ancora $b \in A_T - \{a\}$, la media pesata della relatedness tra p_a e $p_b \in Pg(b)$.

$$vote_b(p_a) = \frac{\sum_{p_b \in Pg(b)} rel(p_b, p_a) \cdot Pr(p_b \mid b)}{\mid Pg(b) \mid} \tag{2.3}$$

Come si vede in 2.3 il peso è dato dalla probabilità a priori che l'ancora b sia associata al senso $p_b \in Pg(b)$, quindi:

$$Pr(p_b \mid b) = \frac{\mid b \to P_b \mid}{\mid link(b) \mid} \tag{2.4}$$

Infine il punteggio dell'annotazione $a \to p_a$ è dato dalla somma di tutti i voti delle altre ancore, quindi

$$rel_a(p_a) = \sum_{b \in A_T - \{a\}} vote_b(p_a) \tag{2.5}$$

Anchor pruning

L'ultima fase, quella relativa all'anchor pruning, ha lo scopo di selezionare tra i sensi (le annotazioni ormai disambiguate) quelli che sono più significative per il testo in esame. Questo avviene associando ad ogni annotazione due features: la probabilità dell'ancora a di rappresentare un link ($lp(a)$) e la coerenza tra il senso associato all'ancora e tutti i sensi delle altre ancore.

$$lp(a) = \frac{link(a)}{freq(a)} \tag{2.6}$$

$$coherence(a \to p_a) = \frac{1}{|S| - 1} \sum_{p(a) \in S - \{p_a\}} rel(p_a, p_b) \tag{2.7}$$

in cui S indica il numero di sensi ottenuti in fase di disambiguazione. Come si vede esaminando la formula la coerenza è una media delle relatedness tra un senso e tutti gli altri trovati nel testo. Link probability e coherence sono quindi combinati per ottenere un valore di pruning (indicato con $\rho(a \to p_a)$) che se inferiore al pruning score (ρ_{NA}) determinerà l'eliminazione del senso tra quelli significativi del testo.

$$\rho(a \to p_a) = \frac{lp(a) + coherence(a \to p_a)}{2} \tag{2.8}$$

La scelta della soglia di pruning è fondamentale nell'equilibrio tra precision e recall poiché un valore troppo alto di ρ determinerebbe un'alta recall a danno della precision. Insieme ad alcuni sensi interessanti infatti non

verrebbero scartati nemmeno altri non rilevanti, mentre avremmo l'effetto opposto (ugualmente spiacevole) se alzassimo troppo la soglia.

TAGME **in un esempio**

Viene qui proposto un esempio per mostrare il funzionamento di TAGME. Il testo che segue rappresenta il titolo di un articolo pubblicato su *La Stampa* il 29 marzo 2012.

`Svolta Mediaset: cacciato Fede. Toti è il nuovo direttore del Tg4`

Alcune delle parole in esso presenti sono, se estratte dal contesto, associabili ad aree semantiche differenti tra loro.

Si procede mostrando il funzionamento di ognuno dei tre moduli su presentati.

Anchor parser. Il testo in input, in questo caso il titolo della notizia viene letto e confrontato con l'insieme di ancore inserite nel dizionario delle ancore. Ciò equivale a individuare tutte le porzioni di testo che sono collegate a uno o più concetti.
Nell'indice che segue le lettere identificano le ancore (per le quali è utilizzato `questo font`) mentre il numeri rappresentano i sensi che questa può avere (riportati in corsivo):

a) `Svolta`:

   1. *Svolta della Bolognina (1989)*
   2. *Svolta di Fiuggi*

   ⋮

b) `Mediaset`:

   1. *Mediaset S.p.A*
   2. *Gruppo Mediaset*
   3. *Mediaset Italia*
   4. *Mediaset Premium*

   ⋮

c) `Cacciato`:

   1. *Caccia*

   ⋮

d) `Fede`:

1. *Fede - simbolo araldico*
2. *Fede - scultura di Donatello*
3. *Fede - dipinto di Raffaello*
4. *Fede - dipinto di Piero del Pollaiolo*
5. *37 Fede - asteroide scoperto nel 1855*
6. *Fede - prenome italiano femminile*
7. *Emilio Fede - giornalista italiano*
8. *Fede - termine usato prevalentemente in ambito religioso come sinonimo di credenza o fiducia in diversi gradi*
9. *Fede - dea romana della lealtà*
10. *Fede nuziale - tipo di anello che gli sposi si scambiano durante la cerimonia di matrimonio*
11. *Fede (Faith) - episodio della quarta stagione di Battlestar Galactica*

⋮

e) `Toti`:

1. *Claudio Toti - imprenditore romano e presidente della Pallacanestro Virtus Roma*
2. *Enrico Toti - Medaglia d'Oro al Valore Militare della prima guerra mondiale*
3. *Enrico Toti - sommergibile classe Balilla varato nel 1928*
4. *Enrico Toti (S 506) - sottomarino classe Toti varato nel 1968*
5. *Giovanni Toti, giornalista italiano*

⋮

f) `Direttore`:

1. *Direttore d'orchestra*
2. *Direttore artistico nell'ambito dello spettacolo*
3. *Direttore del doppiaggio*
4. *Direttore finanziario in ambito aziendale*
5. *Direttore della fotografia in ambito cinematografico*
6. *Direttore generale di organizzazioni pubbliche o private*
7. *Direttore dei lavori di un cantiere*
8. *Direttore responsabile di un giornale*

9. *Direttore di scena in ambito teatrale*
10. *Direttore sportivo*

⋮

g) `Tg4`:

1. *TG4 - testata informativa della rete televisiva italiana Rete 4*
2. *TG4 - canale televisivo irlandese*

⋮

Anchor disambiguation. Una volta individuate le ancore presenti nel testo occorre disambiguarle, ossia trovare per ognuna di esse il senso più appropriato nel contesto.
Per fare questo il software calcola la relatedness tra i vari concetti presenti. Si prendano in esame le ancore d, `Fede`, e g, `Tg4`: il senso d_1, quindi *Fede - simbolo araldico*, avrà con entrambi i concetti di g una relatedness bassissima (vedi formula 2.2): saranno infatti poche (se non nessuna) le pagine di Wikipedia ad avere un link in uscita sia verso d_1 che verso g_1 o g_2.
Se invece si calcola la relatedness tra concetto d_7 (*Emilio Fede*) e i due dell'ancora g (`TG4`) risulta evidente la presenza di un forte legame con il senso *TG4 - testata informativa*, mentre quello con il canale televisivo irlandese sarà anche in questo caso molto debole o inesistente.
Dunque il voto che l'ancora `TG4` da al singolo concetto possibile per `Fede` viene attribuito calcolando la media pesata della relatedness tra questo e i vari sensi g_n. Il peso utilizzato è dato dalla probabilità a priori che l'ancora g faccia riferimento g_n (formuna 2.3). Il concetto tra quelli possibili per `Fede` che massimizza la somma dei voti ottenuti da ogni altra ancora sarà quello risultante dalla disambiguazione. Il procedimento fin qui descritto per l'ancora d viene effettuato per ogni altro spot ottenendo così questo risultato:

a `Svolta`: *Svolta di Fiuggi*

b `Mediaset`: *Mediaset S.p.A*

c `Cacciato`: *Caccia*

d `Fede`: *Emilio Fede - giornalista italiano*

e `Toti`: *Giovanni Toti, giornalista italiano*

f `Direttore`: *Direttore responsabile di un giornale*

g `Tg4`: *TG4 - testata informativa della rete televisiva italiana Rete 4*

Anchor pruning. In questa fase le annotazioni ottenute attraverso i moduli precedenti vengono elaborate con lo scopo di eliminare quelle errate o non rilevanti. A questo scopo si calcolano i valori espressi nelle formule 2.6 e 2.7 indicanti rispettivamente la link probability e la coherence.

La coherence è la media della relatedness che un concetto ha con tutti gli altri annotati nel testo. È evidente che la media ottenuta per il senso *Caccia* sarà inferiore a quella ottenuta da *Mediaset S.p.A.*. Combinando questa misura con la link probability si ottiene ρ, responsabile della cancellazione dell'annotazione in esame. Se questo valore non supera la soglia ρ_{NA} (pari a 0.2) allora il senso verrà eliminato.

Nell'esempio fornito le annotazioni che avrebbero motivo di superare tale fase sono:

b `Mediaset`: *Mediaset S.p.A*

d `Fede`: *Emilio Fede - giornalista italiano*

e `Toti`: *Giovanni Toti, giornalista italiano*

f `Direttore`: *Direttore responsabile di un giornale*

g `Tg4`: *TG4 - testata informativa della rete televisiva italiana Rete 4*

Risultati

I risultati riportati da TAGME hanno messo in luce l'affidabilità di questo strumento. Per quanto riguarda la disambiguazione, quindi l'attribuzione di un senso a una porzione di testo inserita in un contesto limitato a poche parole, TAGME ha raggiunto una F1 del 91.2% con una Precision del 91.5% e una Recall del 90.9%. Un grosso miglioramento sullo stato dell'arte è stato apportato anche per quanto riguarda l'intero task di annotazione per il quale il valore di F1 è superiore al 76%.

Da quanto descritto risulta evidente che sia nella fase di anchor disambiguation che in quella finale di pruning il funzionamento di TAGME è legato alla presenza di diverse ancore nel testo. Non potremmo dunque utilizzarlo per disambiguare parole isolate. Un'ancora ambigua come Battisti non portrebbe rimandare ad alcun concetto se non inserita in un contesto appropriato che mostri qualcosa del mondo cui quell'entità appartiene.

Capitolo 3

RICERCA NEI FEED

Il compito che ci siamo posti è quello di recuperare, data una query, le news interessanti contenute in un feed.

Le difficoltà da affrontare in questo tipo di ricerca sono due: da un lato infatti abbiamo query potenzialmente ambigue, dall'altra la brevità del contenuto testuale di una notizia riportata in un file RSS che rende maggiormente complesso il comprenderne la rilevanza per l'argomento.

L'idea sviluppata in questa tesi è quella di sfruttare TAGME sia per disambiguare la query che per annotare semanticamente ogni news con i concetti più rilevanti in essa contenuti. Il processo di ricerca termina con l'inserimento in un file XML di tutte le news in cui esiste almeno un senso annotato che è strettamente legato a quello cercato.
Quest'idea è stata implementata e quindi valutata su un corpus manualmente annotato.

Nei paragrafi che seguono verranno spiegate in dettaglio le scelte implementate nella ricerca, la creazione del corpus e i risultati ottenuti che saranno paragonati a quelli ottenibili attraverso l'utilizzo di altri metodi. Verrà quindi fornita una loro interpretazione.

3.1 Descrizione del processo di ricerca

Osservando il diagramma di flusso presentato in figura 3.1 è possibile notare che il processo di ricerca è diviso in due parti principali:

- parsing dei feed, annotazione semantica e memorizzazione delle news;
- disambiguazione della query e ricerca.

A queste segue la stampa dei risultati presentati in un file XML.

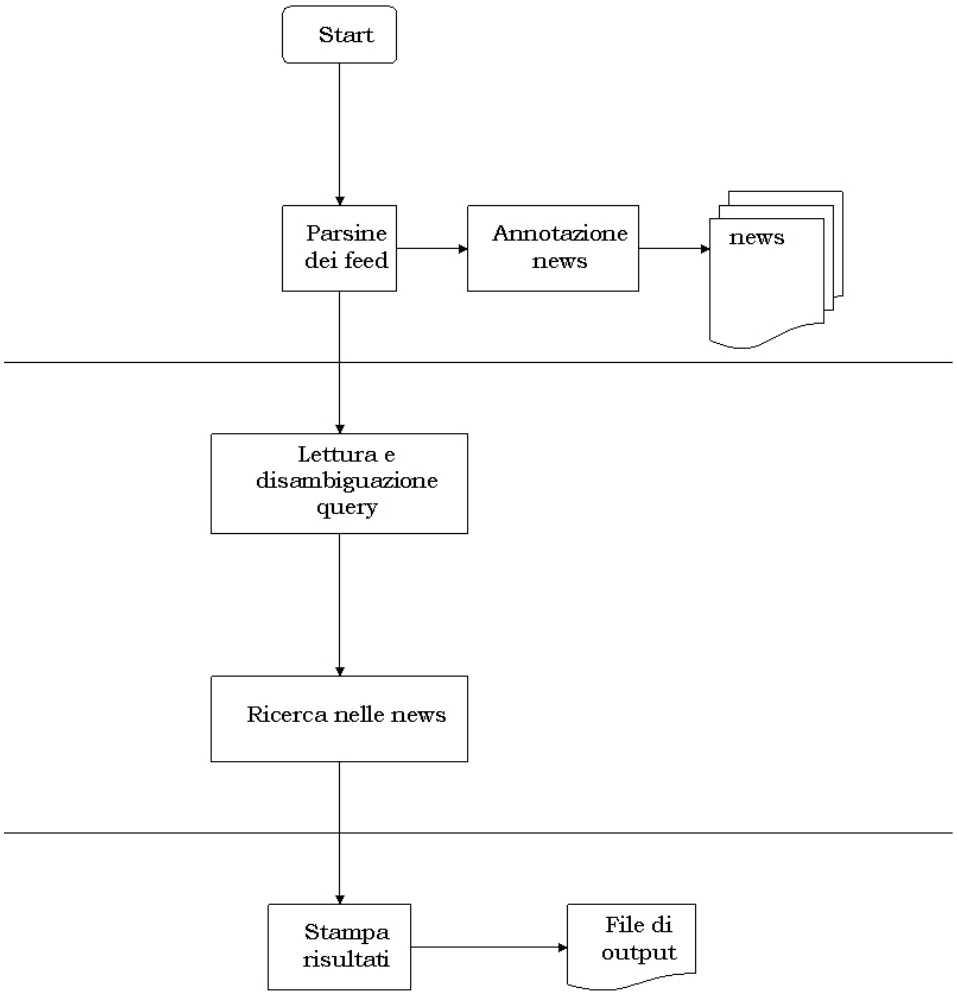

Figura 3.1: Diagramma di flusso

3.1.1 Parsing, annotazione e memorizzazione delle news

Il primo compito che il programma è chiamato a svolgere è il parsing del feed al fine di estrarre i dati circa le fonti e le notizie: sempre in questa fase viene svolta l'annotazione semantica che si conclude con la selezione di un insieme di concetti rilevanti per ogni notizia: questa informazione, unita a quelle estratte dall'RSS, è quindi inserita in memoria così da potervi accedere rapidamente in fase di ricerca, quando bisognerà quindi confrontare il concetto cercato con quelli associati alle news.

Esaminando il feed troviamo due tipi di informazione: quelle relative alla fonte (racchiuse nel tag `channel`) e quelle inerenti le notizie (che sono comprese nei tag `item`).
I dati che vengono memorizzati per il canale sono:

- `title`;
- `description`;
- `link`.

Per quanto riguarda le singole news vengono estratte le informazioni racchiuse nei tag omonimi a quelli della fonte. I tag `title` e `description` riportano il contenuto testuale della notizia che, unito a formare un'unica stringa, costituirà l'oggetto dell'annotazione di TAGME. Il campo `link` contiene invece l'URL per raggiungere il testo completo della notizia. Questi dati (che devono necessariamente essere stati inseriti per rispettare le specifiche in [12]) costituiscono la base dell'annotazione di ogni notizia, alla quale viene aggiunta poi l'informazione semantica estratta automaticamente.

Una volta terminato il parsing di una news se ne processa il contenuto. La stringa composta da titolo e descrizione viene inviata all'anchor parser che ne estrae tutte le possibili ancore scartando quelle che non superano la fase di prepruning così come mostrato a pagina 9.
In questa fase dunque il testo è rappresentato come un insieme di ancore $a_n \in A_{news}$, ognuna delle quali con un proprio insieme di sensi $Pg(a)$.

Conclusa questa fase si procede con la disambiguazione, al termine della quale ogni ancora avrà un solo possibile senso. Come si è visto parlando di TAGME, questo processo si basa sull'idea che due concetti sono tanto più in relazione tra loro quanto maggiore è il numero di volte in cui si trovano insieme in uno stesso contesto. Valutando la relazione che i sensi di p_a dell'ancora a hanno con gli altri possibili concetti del testo si disambigua scegliendo il senso che ha le relazioni più strette, ossia quello per cui è massima la relatedness media.

Le annotazioni in questo insieme sono quindi filtrate dall'anchor pruning. In questa fase a ogni ancora ormai disambiguata viene attribuito un

valore di coherence (formula 2.7) che costituisce la media della similarità che ha con tutti gli altri sensi della frase. La combinazione tra questa e la link probability dell'ancora (formula 2.6) rappresenta il valore responsabile dell'eliminazione dell'annotazione determinando così una diminuzione della cardinalità di A_{news}.

Il valore di soglia (ρ_{NA}) in TAGME corrisponde a 0.2; si è visto tuttavia che per questo tipo di applicazione risulta più conveniente adottare una soglia inferiore, in particolare, come riportato in 3.3, un valore pari a 0.05.

Il risutato di questo processo di annotazione è dunque un insieme di concetti P_{news} (ognuno dei quali associato a un'ancora in A_{news}) che costituisce l'informazione semantica della notizia in esame.

Rappresentazione della news

Ogni news viene dunque rappresentata attraverso la seguente ennupla di elementi:

- testata
- sezioni (potenzialmente più di una)
- link alla fonte
- link all'articolo completo
- titolo
- descrizione
- insieme di concetti rilevanti (P_{news})

3.1.2 Risoluzione delle query

Una volta organizzata la conoscenza sul contenuto delle news nella maniera più conosona al nostro scopo, si procede alla disambiguazione della query e quindi alla selezione delle notizie che presentano nel loro bagaglio semantico concetti affini a quello di interesse. La disambiguazione prevede nuovamente l'utilizzo di TAGME, in particolare dei moduli di anchor parsing e disambiguation.
La ricerca delle news invece si fonda sul valore di similarità tra il concetto cercato e almeno uno di quelli aggiunti da TAGME alla notizia.

Si può quindi riassumere che data una query verranno recuperate tutte quelle news contenenti almeno un senso $p_n \in P_{news}$ che è strettamente in relazione con uno dei sensi risultanti dalla disambiguazione della query.

Volendo affrontare una ricerca che provi a muoversi su un livello semantico si ha la necessità di avere una query che non sia più costituita da parole ma da concetti: anche per questo compito si è fatto uso di TAGME.
Dei suoi tre moduli (presentati a pagina 9) se ne utilizzano qui esclusivamente due:

- anchor parser, per trovare le ancore nella query;
- anchor disambiguator, per associare ognuna di esse a un solo senso.

Si è preferito non utilizzate l'anchor pruning poiché si ritiene che ogni parola inserita dall'utente abbia un valore.

L'anchor parser genera l'insieme delle ancore possibili A_{query}. Anche in questo caso ogni ancora $a \in A_{query}$ è collegata a un insieme $Pg(a)$ contenente i diversi sensi cui l'ancora può riferirsi. Attraverso l'anchor disambiguator per ogni a viene selezionato un solo $p_a \in Pg(a)$ che viene quindi inserito nell'insieme dei concetti cercati P_{query}.

Il processo di disambiguazione non può però essere avviato nel caso in cui venga restituita una sola ancora per la query poiché non si avrebbe a disposizione il contesto necessario per il calcolo della relatedness.
Per ovviare a questo problema nel caso in cui si abbia a che fare con un'ancora isolata si può decidere di selezionare il concetto con la commonness maggiore. Verrebbe così scelto il significato al quale in Wikipedia si fa più spesso riferimento utilizzando quella forma.

Una volta ottenuto l'insieme dei sensi P_{query} si procede alla ricerca delle notizie annotate con almeno un concetto avente con quello cercato una relatedness superiore alla soglia δ. Si è deciso di introdurre questo valore di similarità perché l'annotazione di un concetto potrebbe in alcuni casi implicare l'annotazione di un'altro: ad esempio se un testo in cui viene annotato il concetto "Windows 7" risulterà interessante anche per il concetto "Microsoft Windows". Grazie a questa soglia di relatedness dunque possono essere recuperati risultati anche se non annotati. Inoltre questo valore permette di attutire piccoli errori presenti nell'annotazione fatta da TAGME. Il processo di ricerca consiste dunque nell'estrazione di tutte le news per cui è vero che esiste almeno una coppia p_q, p_n per cui

$$rel(p_q \in P_{query}, p_n \in P_{news}) > \delta \tag{3.1}$$

Dagli esperimenti fatti (riportati in 3.4) è emerso che il valore di δ che massimizza la F1 è 0.95.

3.1.3 Formato dell'output

Le news recuperate vengono inserite in un file RSS che contiene le seguenti informazioni: racchiusi nell'elemento `rss` si trovano tanti `channel` quante sono le diverse fonti dalle quali le news sono state recuperate. Per ogni

`channel` sono riportati i campi obbligatori: `title`, `description`, `link` e quindi gli `item` che contengono a loro volta i tag `title`, `description` e `link` estratti dal feed originale. A questi sono aggiunti i campi `category` per riportare i concetti della query per i quali la news è stata recuperata.

Segue un esempio di file di output.

```
<rss xmlns:atom=‘‘http://www.w3.org/2005/Atom’’ version=‘‘2.0’’>
 <channel>
    <title>ANSA.it</title>
    <link>http://www.ansa.it</link>
    <description>Updated every day</description>
    <item>
       <title>
          Google, nuove regole privacy Altola’ Ue
       </title>
       <description>
          All’ANSA la spiegazione integrale di Big G
       </description>
       <link>
          http://www.ansa.it/web/notizie/rubriche/tecnologia/
          2012/02/28/visualizza_new.html_105303799.html
       </link>
       <category>Google</category>
       <category>Unione Europea</category>
    </item>
    <item>...</item>
 </channel>

 <channel>...</channel>
</rss>
```

Mantenere l’informazione in questo formato permetterà sia di visualizzare questo file in una pagina web che di inviarlo a un feed reader (utilizzandolo quindi come output di un filtro).

3.2 Creazione del corpus

Per comprendere la validità del modello proposto si è resa necessaria la creazione di un corpus annotato. Questo dunque costituisce il gold standard da utilizzare come termine di paragone per la valutazione dei risultati delle ricerche effettuate.

Il corpus è stato creato a partire dai file RSS di due tra i più diffusi quotidiani italiani (il *Corriere della Sera* e *La Repubblica*), di una agenzia di stampa molto conosciuta (*ANSA*) e di un sito specialistico (*Punto Informatico*): si presenta come un file XML in cui sono state inserite alcune delle informazioni presenti nei feed scaricati. Queste sono: testata, sezioni (0, 1 o più), titolo, descrizione, link e data di pubblicazione delle news.
Segue un frammento di file allo scopo di mostrarne la struttura:

```
<xml>
   <rss>
      <newspaper>nome fonte</newspaper>
      <section>sezione</section>
      <title>titolo dell'articolo</title>
      <description>sottotitolo dell'articolo</description>
      <link>link all'articolo</link>
      <pubDate>data nel formato rss</pubDate>
   </rss>
   <rss>
      ...
   </rss>
</xml>
```

La scelta delle prime tre fonti indicate (*Corriere della Sera*, *La Repubblica* e *ANSA*) deriva dalla volontà di valutare un reale processo di ricerca così come potrebbe essere effettuato da un utente interessato a rimanere aggiornato sull'attualità. La quarta fonte è stata aggiunta per valutare il comportamento del modello su testi tecnici e settoriali come quelli pubblicati da *Punto Informatico*.

In totale le news raccolte sono 4195, scaricate dal 2 al 14 settembre ma relative a un periodo più ampio: di queste infatti 3813 sono pubblicate nei giorni del download, le altre fanno riferimento a date precedenti.

Nella tabella 3.1 è riportato il numero di articoli per testata; per ognuna di esse inoltre sono indicate le sezioni cui gli articoli appartengono con la relativa frequenza.

Il grafico in figura 3.2 mostra invece la distribuzione delle notizie per sezione ignorando la testata. Sono state ad esempio riunite tutte le notizie di politica. In un'unica sezione sono raggruppate anche le news provenienti

Corriere della Sera		407
	Politica	14
	Sport	21
	Cronaca	35
	Motori	56
	Cinema	16
	Cultura	20
	Scienza	31
	Spettacoli	16
	Editoriali	9
	Homepage	158
	Animali	31
La Repubblica		1277
	Politica	27
	Solidarietà	39
	Sport	107
	Cronaca	32
	Persone	22
	Arte	9
	Gallerie	15
	Spettacoli e cultura	35
	Scienza	24
	Tecnologia	31
	Viaggi	16
	Tv	3
	24ore AGI	612
	Homepage	134
	Salute	32
	Economia	42
	Ambiente	13
	Mondo	74
	Scuola e Università	10
Ansa		2395
	Politica	109
	Calcio	240
	Cronaca	191
	Cinema	71
	Topnews	479
	Altrisport	184
	Cultura	122
	Scienza	44
	Tecnologia	22
	Spettacolo	135
	Homepage	189
	Economia	281
	Mondo	328
Punto Informatico		116

Tabella 3.1: Numero di articoli per testata cui segue il dettaglio per sezioni.

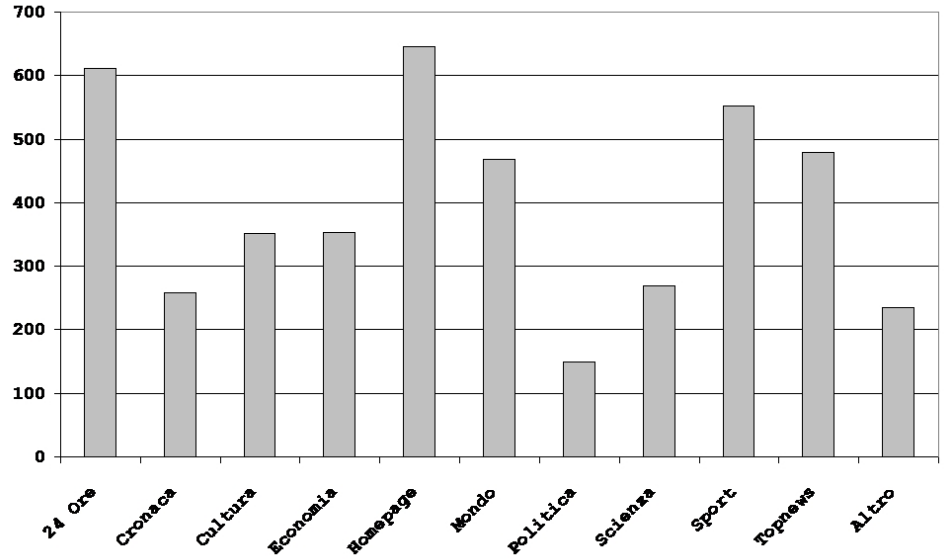

Figura 3.2: Numero di news raggruppate per sezione

da Tecnologia e Scienza e tutte quelle scaricate da Punto Informatico, quelle provenienti da Sport, Calcio e Altri Sport, quelle in Cultura, Spettacoli/Spettacolo, Cinema, Arte e Gallerie e infine quelle in Animali, Editoriali, Solidarietà, Persone, Viaggi, Motori, Tv, Scuola e Università, Ambiente e Salute (classificati come Altro).

Sono state quindi create 30 query, consistenti ognuna di un singolo concetto già disambiguato e di una forma attraverso la quale esprimerlo. La scelta dei concetti è stata fatta seguendo due criteri:

- ottenere un numero non troppo esiguo di news rilevanti;
- differenziare le aree tematiche.

Nella tabella 3.2 sono riportati i concetti selezionati.

Ogni news del corpus è stata quindi manualmente annotata, specificando per quale delle query essa risulta rilevante. L'annotazione è stata fatta attribuendo un valore (0 o 1) per ogni coppia <*news-query*> in base alla rilevanza della news per la query.

Il numero complessivo delle coppie è 124850: di queste 1528 sono risultate essere positive.

Questa informazione è inserita nel file XML subito dopo quelle sopra riportate. Segue un esempio estratto dal corpus:

	QUERY	concetto	news inerenti
1	Fini	Gianfranco Fini	22
2	Berlusconi	Silvio Berlusconi	237
3	Bersani	Pier Luigi Bersani	57
4	Di Pietro	Antonio Di Pietro	20
5	Grecia	Grecia	51
6	Libia	Libia	191
7	Obama	Barack Obama	54
8	Camorra	Camorra	19
9	FIAT	FIAT	46
10	Chiesa	Chiesa cattolica	62
11	Carceri	Prigioni	36
12	Scuola	Scuola	60
13	Sciopero	Sciopero	34
14	Corruzione	Corruzione	38
15	Terremoto	Terremoto	12
16	Sindacato	Sindacato	48
17	Ciclone	Ciclone tropicale	32
18	Servizio Sanitario	Servizio Sanitario Nazionale (Italia)	31
19	Tubercolosi	Tubercolosi	14
20	Attentati dell'11 settembre	Attentati dell'11 settembre 2011	78
21	Nazionale di calcio	Nazionale di calcio italiana	56
22	Valentino Rossi	Valentino Rossi	19
23	Formula 1	Formula Uno	42
24	Mostra del cinema di Venezia	Mostra Internazionale d'Arte Cinematografica	147
25	Rai	RAI - Radiotelevisione Italiana	20
26	Amy Winehouse	Amy Winehouse	8
27	Vasco Rossi	Vasco Rossi	32
28	Android	Android	7
29	Sistemi operativi	Sistemi operativi	25
30	Facebook	Facebook	30
totale			1527

Tabella 3.2: La seconda colonna contiene la forma utilizzata per la ricerca testuale. Per rappresentare il concetto (riportato nella terza colonna) si è utilizzata la pagina di Wikipedia che è oggetto della ricerca. L'ultima colonna mostra il numero di articoli rilevanti per la query.

```
<xml>
 <rss>
    <newspaper>corriere</newspaper>
    <section>Politica.xml</section>
    <title>Bersani: ‘‘La manovra resta iniqua’’ I sindacati in
coro: novità negative</title>
    <description>Il segretario del Pd: ‘‘La fiducia? Questo
governo sa solo mentire’’. Di Pietro: ‘‘Napolitano sciolga le
Camere’’</description>
    <link>http://www.corriere.it/politica/11_settembre_06/
bersani_manovra_iniqua_5fe0770e-d89d-11e0-b038-3e67ea432e86.shtml</link>
    <pubDate>Tue, 6 Sep 2011 21:45:46 +0200</pubDate>
    <annotation title=Fini id=334345> 0 </annotation>
    <annotation title=Berlusconi id=601725> 1 </annotation>
    ...
 </rss>
 <rss>
    ...
 </rss>
```

La figura 3.3 riporta la distribuzione della news rilevanti per ogni concetto, dalla quale risultano evidenti tre picchi di interesse che corrispondono a “Silvio Berlusconi”, “Libia” e “Mostra Internazionale d’Arte Cinematografica” (o più comunemente “Mostra del cinema di Venezia”).

L’ultimo grafico riportato (3.4) mostra invece la distribuzione nelle sezioni delle news che sono risultate rilevanti per almeno una query.

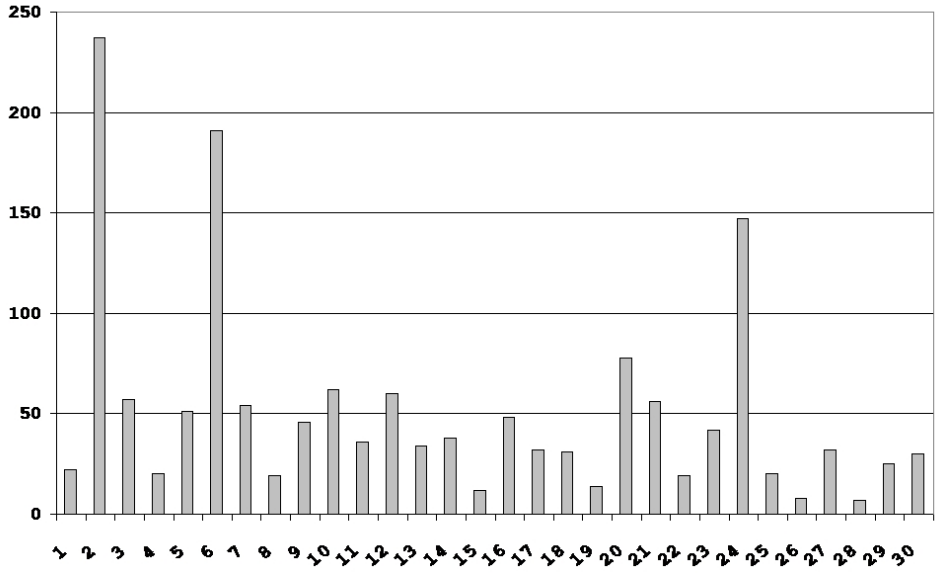

Figura 3.3: Grafico della distribuzione della frequenza delle news rilevanti per concetto

3.3 Valutazione sperimentale

Sul corpus costruito così come mostrato precedentemente è stato testato il modello descritto nel paragrafo 1 di questo capitolo.

Il sistema utilizza una fotografia di Wikipedia risalente a maggio 2011: ciò implica che sono ignorate le informazioni (che qui si identificano con le pagine dell'enciclopedia e le relazioni tra esse) su ciò che è accaduto nei 4 mesi precedenti al download delle news presenti nel ground truth.

Nel paragrafo 2.2 si è parlato di un valore ρ che TAGME utilizza come soglia al di sotto della quale le annotazioni individuate per il testo vengono eliminate. Da questa variabile dipende quindi il bilanciamento tra Precision e Recall, ossia l'equilibrio tra il numero di concetti annotati e la qualità dell'annotazione: a un valore di soglia più alto corrisponde un numero di concetti rilevanti più esiguo; viceversa una soglia molto bassa aumenta il numero di annotazioni (alcune delle quali potenzialmente poco o affatto rilevanti per il testo).

L'estrazione di notizie interessanti per una query richiede l'identificazione del valore di un'altra soglia, quella di relatedness (δ) tra senso cercato e senso annotato nel testo. Come descritto in 3.1.2 il task viene risolto estraendo tutte le news per le quali TAGME ha annotato un concetto che ha un valore di relatedness con quello nella query superiore alla soglia δ.

Per trovare ρ e δ più adatti al compito è stato quindi necessario fare delle prove. Nella tabella 3.3 si riportano i valori delle soglie che hanno dato i

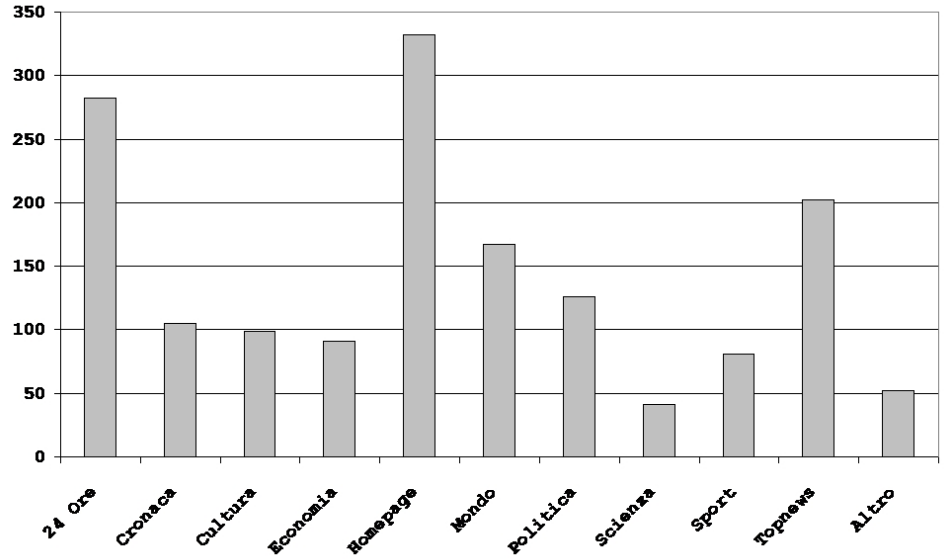

Figura 3.4: Grafico della distribuzione nelle sezioni delle news annotate come rilavanti per una query

migliori risultati in termini di Precision, Recall e F1.

ρ	δ	**Accuracy**	**Precision**	**Recall**	**F1**
0.05	0.95	99.2	75.4	51	**60.9**
0.05	0.9	99.2	73.3	**51.7**	60,6
0.1	0.95	99.2	**79.9**	48.5	60,4

Tabella 3.3: Tabella riportante le variazioni di Precision, Recall e F1 in base alla sezione di appartenenza

Utilizzando i valori di rho e relatedness per cui la F1 risulta massima si è valutato il funzionamente del sistema sulle singole sezioni. I risultati così ottenuti sono indicati nella tabella 3.4.

Nel grafico 3.5 è riportato per ogni sezione il numero di coppie query-news annotate come rilevanti nel ground truth e recuperate (TP) o non recuperate (FN) dal sistema mentre in 3.6 sono mostrate le news recuperate dal software dividendole in rilevanti (TP) e non rilevanti (FP) nel ground truth.

Come si evince dalla tablella alcune sezioni presentano dei valori anomali rispetto alla media complessiva. Evidente è il caso della sezione `24 ore` che ha una F1 di 15 punti percentuali superiore a quella dell'intero corpus. Nel paragrafo 3.4 si esamineranno questi dati fornendo una motivazione per i

SEZIONI	Precision	Recall	F1
24ore	79.7	72.3	75.8
Cronaca	86,4	36,2	51
Cultura e Spettacolo	76,5	49,5	60,1
Economia	82,8	63,7	72
Home	71,3	36,7	48,5
Mondo	74,8	55,1	63,4
Politica	95	45,2	61,3
Scienza e Tecnologia	71,8	56,1	63
Sport	46,3	61,7	52,9
Topnews	80.8	50.5	61.8
Altro	69.2	34.6	46.1
Tutte le sezioni	75.4	51	60.9

Tabella 3.4: Tabella riportante i valori di Precision, Recall e F1 in base alle sezioni dei giornali. Punto Informatico è stato inserito nella sezione Scienza e Tecnologia ρ e δ

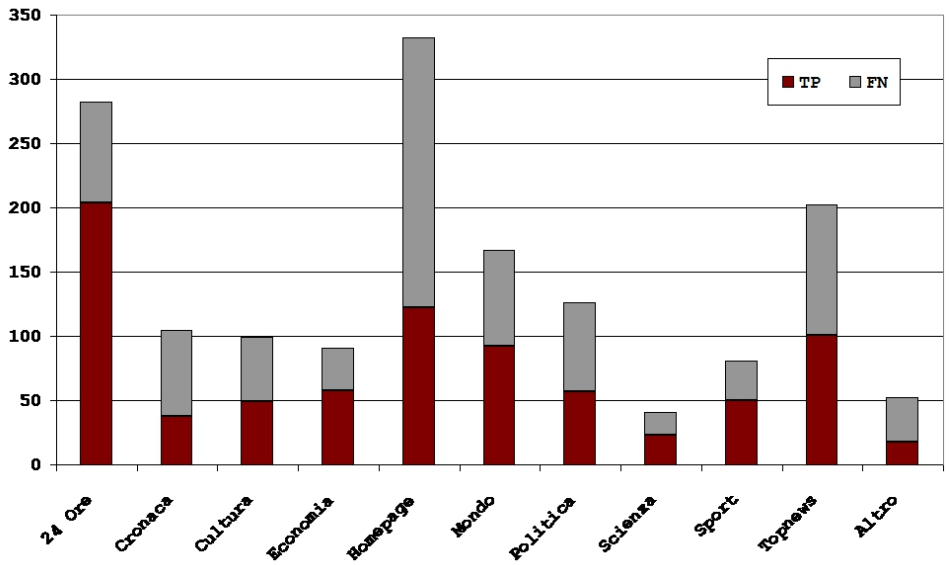

Figura 3.5: Grafico in cui viene riportato per ogni sezione del corpus il numero di articoli identificati come rilevanti per almeno una query nel ground truth. I colori indicano la suddivisione in True Positive (TP) e False Negative (FN)

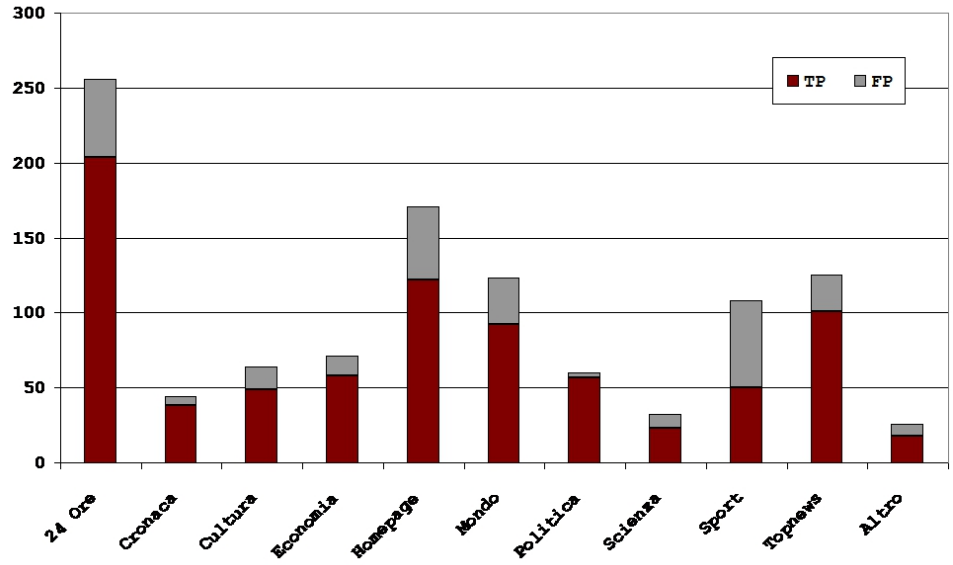

Figura 3.6: Grafico in cui viene riportato per ogni sezione del corpus il numero di articoli recuperati come rilevanti per almeno una query dal sistema. I colori indicano la suddivisione in True Positive (TP) e False Positive (FP)

risultati.

Confronto con altri due metodi di ricerca

I risultati ottenuti da questo tipo di ricerca sono stati confrontati con quelli ottenuti utilizzando due metodi differenti: nella tabella 3.5 si riportano i risultati.

METODO	Accuracy	Precision	Recall	F1
Testuale	99	**85.3**	25.8	39.7
Anchor Parser	99.2	75.8	48.1	58.9
tagme	99.2	75.4	**51**	**60.9**

Tabella 3.5: Valori di precision recall e F1 per i tre tipi di ricerca

Il primo metodo, quello chiamato testuale, è stato scelto tra 3 metodi sperimentati che lavorano esclusivamente sul match tra query e testo della news. Si è scelto di indicare quello con il valore di F1 maggiore: con questo metodo una news viene recuperata se contiene tutte le parole presenti nella query ad eccezione delle stop words che sono state eliminate.
Le altre 2 tecniche provate prevedono il recupero della news se

1. contiene almeno una delle parole della query (stop words escluse) (Accuracy = 97.2; Precision = 17.2; Recall = 33; F1 = 22.6);

2. contiene il match esatto della query (stop words incluse), in cui viene rispettato anche l'ordine di inserimento delle parole (Accuracy = 99; Precision = 85,6; Recall = 24,9; F1 = 38,6).

Il valore basso per la Recall nell'ultimo metodo è chiaramente dovuto alla difficoltà di trovare una news che contenga la query esattamente così come inserite, soprattutto per quelle fortemente caratterizzate come "Mostra del Cinema di Venezia". A una bassa Recall corrisponde un valore alto della Precision.

Il secondo tipo di ricerca utilizzato per il confronto, quello indicato con il nome "Anchor Parser", è stato implementato per comprendere l'utilità della fase di disambiguazione nel processo proposto in 3.1. Anche in questo caso la query deve essere costituita da entità semantiche: partendo da queste si memorizzano tutti i modi che si utilizzano per esprimere il/i concetto/i da cercare e si procede dunque con il match. Questo tipo di ricerca è assimilabile al cercare sinonimi in un testo: è dunque una ricerca testuale in cui la query immessa è estesa fino a comprendere tutte le forme utilizzabili per esprimere il concetto da cercare.

Questo tipo di approccio è quello che alcuni motori adottano basandosi su co-occorrenze di parole. Il metodo qui utilizzato per estrarre tutte le formule si basa invece su Wikipedia: sono state prese tutte le ancore che hanno tra i possibili sensi quello cercato.

Una volta ottenuto questo insieme di espressioni è stato cercato un criterio che potesse rendere più efficiente la ricerca. Poiché ad ogni ancora è associata una link probability (cioè la probabilità che l'ancora sia un link e non solo testo) si è cercato di comprendere se le performance del sistema di ricerca erano in relazione con questo valore.

In maniera analoga si è studiata la variazione della qualità del processo di ricerca al variare di un'altra soglia, quella della commonness tra l'ancora e il senso. La commonness corrisponde alla probabilità che quell'ancora faccia riferimento al senso in esame; si è cercato quindi di capire se esiste una soglia per questo valore al di sotto della quale, anche in presenza di match tra query e testo, la news avrebbe rappresentato più probabilmente un falso positivo.

Sono stati inoltre combinati questi due valori attraverso il prodotto ottenendone uno che rappresenta la probabilità che il testo individuato rappresenti un'ancora per il senso cercato. Data l'ancora a e il senso cercato p_a si riportano le formule per una più chiara comprensione delle tre soglie utilizzate.

$$lp(a) = \frac{\mid link(a) \mid}{\mid freq(a) \mid} \tag{3.2}$$

$$commonness(a \rightarrow p_a) = \frac{\mid (a \rightarrow p_a) \mid}{\mid link(a) \mid} \tag{3.3}$$

$$lp(a) \times commonness(a \rightarrow p_a) = \frac{\mid link(a) \mid}{\mid freq(a) \mid} \times \frac{\mid (a \rightarrow p_a) \mid}{\mid link(a) \mid} = \frac{\mid a \rightarrow p_a \mid}{\mid freq(a) \mid} \tag{3.4}$$

Si riportano inoltre i grafici con l'andamento di Precision, Recall e F1 per le tre sogli considerate. Nelle ascisse sono riportati i valori delle soglie, nelle ordinate la percentuale per ogni misura. Il primo grafico (3.7) è quello che utilizza la link probability. Come si vede la F1 rimane pressappoco costante al variare di lp rendendo inutile il suo utilizzo. La legenda qui inserita è comune per i 3 grafici.

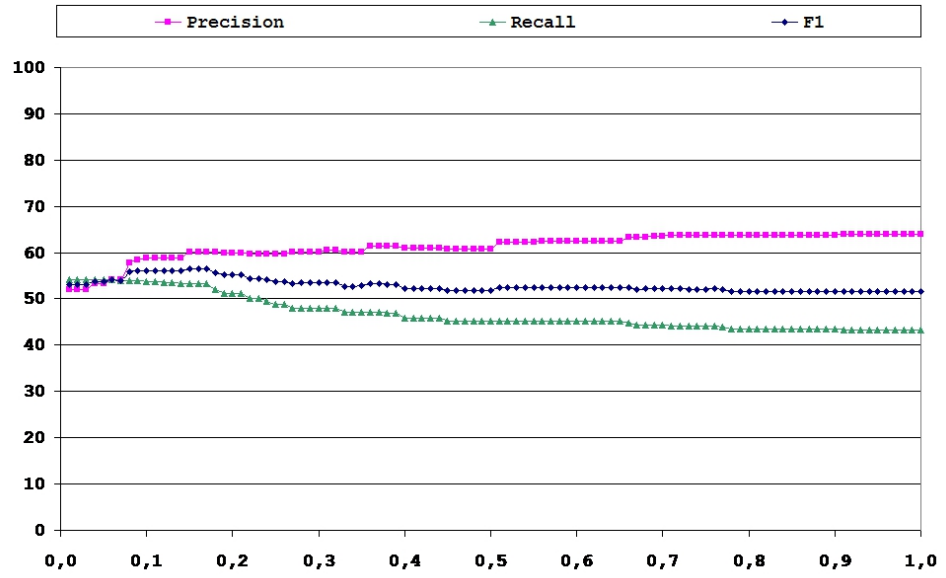

Figura 3.7: Variazioni di Precision, Recall e F1 al variare della link probability (in ascissa).

Il grafico successivo (3.8) mostra i tre valori al variare della commonness. Queste tre curve, sebbene abbiano una parte centrale che mostra valori all'incirca costanti per ogni misura, sono costituite da una parte iniziale e una finale in cui si riscontrano variazioni considerevoli.

L'ultimo grafico (3.9) è quello che vede come soglia il prodotto tra link probability e commonness.
Dato l'andamento discendente dell'F1 si è preferito non prendere in considerazione questa misura.

Visti i tre andamenti dunque il valore di soglia più adatto al task è quello dato da commonness = 0.05.

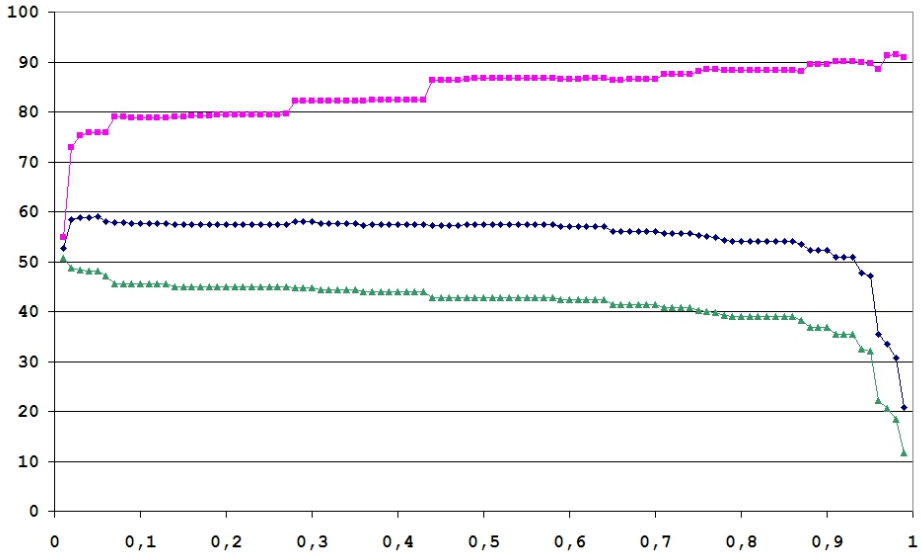

Figura 3.8: Variazioni di Precision, Recall e F1 al variare della commonness (in ascissa).

Per motivi di chiarezza si è preferito descrivere questo processo di ricerca nella maniera fino a qui indicata.
Per comodità implementativa però la ricerca si è sviluppata selezionando i possibili sensi cui le varie ancore presenti nel testo rimandavano per poi cercare tra questi quello inserito nella query.

In dettaglio: le news vengono lette da un parser XML che seleziona il testo contenuto nei tag `title` e `description` di ogni elemento `item` e lo invia all'anchor parser di TAGME (vedi pag. 9) che ne estrae tutti i possibili spot. Il risultato di questo processo è un elenco di ancore, ognuna delle quali fa riferimento a un insieme di concetti (tipicamente più di uno in quanto non ancora disambiguata). Si procede dunque a salvare i sensi che hanno con l'ancora una commonness maggiore di 0.05.
Il task di ricerca si conclude con la restituzione di tutte le news per le quali è stato recuperato uno dei concetti indicati nella query. Utilizzando la dicitura fin qui adoperata sia $Pg(news)$ l'insieme derivante dall'unione di tutti gli insiemi ($Pg(a)$) di sensi possibili per ogni ancora $a \in A_{news}$; vanno quindi eliminati da questo insieme tutti i sensi $p(a)$ per cui $commonness(a, p(a)) < 0.05$.
Se $Pg(news) \cap P_{query}$ (dove P_{query} è l'insieme dei concetti della query) genera un insieme non vuoto (quindi se i due insiemi hanno almeno un elemento in comune) allora la news viene considerata interessante.

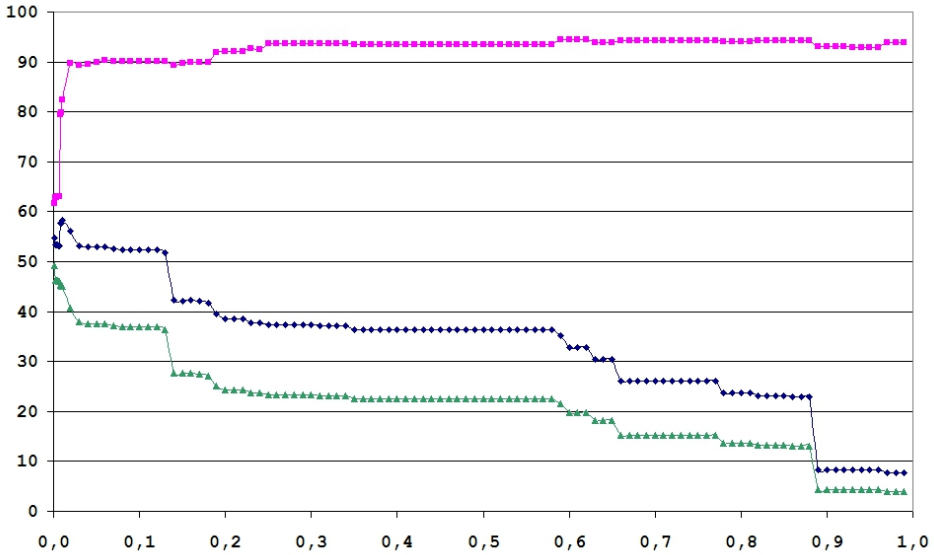

Figura 3.9: Variazioni di Precision, Recall e F1 al variare del prodotto tra link probability e commonness (in ascissa).

Le tre tipologie di ricerca in un esempio

Viene qui simulato il processo di estrazione di news da un canale RSS utilizzando le tre tipologie di ricerca in modo da chiarirne meglio le differenze.

Si utilizza `questo font` per indicare le ancore, mentre il corsivo indica i *concetti*.

Si riporta un breve frammento di feed composto esclusivamente da due news, sufficiente però a mostrare il funzionamento dei tre modelli.

```
<rss xmlns:atom=http://www.w3.org/2005/Atom version=2.0>
<channel>
   <atom:link rel=self type=application/rss+xml href=http://www.ansa.it/>
   <title>ANSA.it</title>
   <link>http://www.ansa.it</link>
   <description>Updated every day - FOR PERSONAL USE ONLY</description>
   <language>it</language>
   <copyright>
      Copyright:  (C) ANSA, http://www.ansa.it/web/static/disclaimer.html
   </copyright>
   <item>
      <title>
         <![CDATA[ Cinema:  e' morto Tonino Guerra ]]>
      </title>
      <description>
```

```
         <![CDATA[
            Aveva compiuto da poco 92 anni
         ]]>
         </description>
         <link>
            http://www.ansa.it/web/notizie/regioni/marche/2012/03/21/
visualizza_new.html_134853679.html
         </link>
         <pubDate>21 Mar 2012 11:43:00 +0100</pubDate>
         <guid>
            http://www.ansa.it/web/notizie/regioni/marche/2012/03/21/
visualizza_new.html_134853679.html
         </guid>
      </item>
      <item>
         <title>
            <![CDATA[ Afghanistan, in Usa 69% contro la guerra ]]>
         </title>
         <description>
         <![CDATA[
            Sondaggio Nyt/Cbs:  contrario 60% repubblicani e 68%
democratici
            ]]>
         </description>
         <link>
            http://www.ansa.it/web/notizie/rubriche/mondo/2012/03/27/
visualizza_new.html_157735061.html
         </link>
         <pubDate>27 Mar 2012 17:14:00 +0100</pubDate>
         <guid>
            http://www.ansa.it/web/notizie/rubriche/mondo/2012/03/27/
visualizza_new.html_157735061.html
         </guid>
      </item>
   </channel>
</rss>
```

Il desiderio dell'utente è quello di acquisire informazioni sulle guerre in corso. Il concetto viene rappresentato dalla pagina di Wikipedia che si trova all'indirizzo http://it.wikipedia.org/wiki/Guerra e corrisponde alla descrizione di *Guerra, evento sociale e politico generalmente di vaste dimensioni*. La parola chiave inserita è "Guerra".

Ricerca testuale: match della query. Questo tipo di ricerca prevede una fase di preprocessing in cui si eliminano dal testo cercato le stop words (o parole grammaticali). Essendone in questo caso privo, tale fase non produce modifiche. Si procede dunque cercando eventuali match della query all'interno del testo racchiuso tra i tag `title` e `description` di ogni `item` del file RSS.

La parola guerra compare nel titolo sia della prima che della seconda notizia: verranno quindi recuperate entrambe anche se la prima risulterà un falso positivo in quanto viene affrontato il tema della morte dello sceneggiatore Tonino Guerra.

Ricerca basata sull'anchor parser: match della query allargata. Sapendo di voler recuperare tutte le informazioni inerenti il tema descritto nella pagina http://it.wikipedia.org/wiki/Guerra di Wikipedia si selezionano tutti gli articoli che contengono un'ancora che almeno lo 0.05 delle volte sia servita a indicare il concetto cercato.

Il procedimento attuato è il seguente: si estrae per ogni `item` del file il contenuto dei tag `title` e `description`. Il testo derivante dalla loro unione viene processato dall'anchor parser che ne estrae tutte le possibili ancore, quindi per ogni ancora tutti i concetti cui può fare riferimento. Segue una porzione dei sensi possibili per la prima news:

a) `Cinema`

1. *Cinema, forma d'arte moderna, nonché uno dei più grandi fenomeni culturali, nata alla fine del XIX secolo, nota anche come la "settima arte".*
2. *Cinema, nome con cui comunemente viene definita la sala cinematografica.*
3. *Cinema - rivista italiana di critica cinematografica fondata nel 1935 da Luigi Freddi.*
4. *Cinema - album del 1980 della cantante pop italiana Viola Valentino.*

⋮

b) `Tonino Guerra`

1. *Tonino Guerra, Antonio Guerra (1920), poeta, scrittore e sceneggiatore italiano*

c) `Guerra`

1. ***Guerra, evento sociale e politico generalmente di vaste dimensioni.***

2. *Guerra: la guerra nell'analisi filosofica.*
3. *Guerra - EP dei Litfiba.*
4. *Alfonso Guerra (1845 - 1920), ingegnere e architetto italiano*

⋮

⋮

I puntini indicano che si sono riportati solo alcuni tra i possibili sensi per l'ancora. Si nota che tra i sensi compare quello che stavamo cercando (messo in risalto attraverso l'uso del grassetto) che avendo un'alta commonness con l'ancora `guerra` non viene scartato permettendo così alla news di essere recuperata. Anche in questo caso tuttavia ci troviamo di fronte a un false positive. Si ripete lo stesso procedimento per la seconda notizia. Anche qui la presenza dell'ancora `guerra` permetterà il recupero (correttamente questa volta) della news.

Ricerca basata su TAGME**.** Questo tipo di ricerca prevede la disambiguazione del testo relativo alla news. Viene quindi estratto e unito in un'unica stringa il contenuto dei tag `title` e `description` di ogni `item`, inviato all'anchor parser per l'estrazione di tutte le possibili ancore che poi vengono processate dall'anchor disambiguator. Alla fine di questi passaggi avremo il seguente set di annotazioni

a) `Tonino Guerra`: *Antonio Guerra (1920), poeta, scrittore e sceneggiatore italiano*

b) `Cinema`: *Cinema, forma d'arte moderna, nonché uno dei più grandi fenomeni culturali, nata alla fine del XIX secolo, nota anche come la "settima arte".*

c) `Morte`: *Morte, cessazione delle funzioni biologiche che definiscono gli organismi viventi*

d) `Anno`: *Anno, periodo di tempo pari a quello impiegato dalla Terra per completare la sua orbita attorno al Sole*

Si può subito vedere che tra i sensi individuati manca quello cercato. Nonostante ciò la ricerca non si interrompe ma continua come segue: per ognuna delle annotazioni si calcola il valore di rho (vedi formula 2.8) in modo da eliminare quelli che non superano la soglia di 0.05. Per le restanti annotazioni si provvede a calcolarne la relatedness con il concetto cercato.

In questo caso tutte le annotazioni sono molto lontane per cui la news non viene recuperata.

L'uso di TAGME dunque è riuscito ad evitare che venisse restituita una news non rilevante per la query.

Si osservi ora il comportamento del modello di ricerca sulla seconda delle news riportate. Le annotazioni prodotte dall'anchor disambiguator in questo caso sono le seguenti:

a) `Afghanistan`: *Afghanistan, stato di 647.500 km^2 con capitale Kabul*

b) `CBS`: *Columbia Broadcasting System - nome originario della CBS, network televisivo statunitense*

c) `USA`: *Stati Uniti d'America*

d) `Repubblicani`: *Partito Repubblicano degli Stati Uniti d'America*

e) `Democrazia`: *Democrazia, etimologicamente governo del popolo.*

f) `Guerra`: *Guerra del Vietnam*

g) `Sondaggio`: *Sondaggio d'opinione*

Si nota subito un errore nell'annotazione: TAGME infatti attribuisce all'ancora `guerra` il significato di *Guerra del Vietnam*. La motivazione è evidente conoscendo il funzionamento del software: le ancore vicine sono le stesse che potremmo trovare parlando della guerra degli anni '60. Una volta terminata la fase di disambiguazione si procede al calcolo del rho (ρ) e quindi della relatedness (δ) tra queste e il concetto cercato. Grazie alla flessibilità derivante dal δ la news, sebbene mancante dell'annotazione al concetto esatto, verrà recuperata e visulizzata tra i risultati.

Osserviamo ora cosa accade se si cerca di recuperare informazioni sullo scrittore Antonio Guerra (concetto rappresentato dalla pagina http://it.wikipedia.org/wiki/Tonino_Guerra di Wikipedia). La query utilizzata per cercarlo è "Antonio Guerra".

Ricerca testuale: match della query. In questo caso la ricerca testuale non produrrebbe alcun risultato mancando il match relativo al nome.

Decidendo invece di restituire una news se ha un match con almeno una delle parole della query, verrebbero restituite entrambe le notizie poiché entrambe contenenti la parola "Guerra".

Scegliere l'uno o l'altro metodo significa quindi spostare il problema da una bassa Recall a una bassa Precision.

Ricerca basata sull'anchor parser: match della query allargata. L'anchor parser in questo caso ci permette di comprendere che Antonio Guerra corrisponde a Tonino Guerra per cui la prima news verrà restituita. Dal momento che l'ancora `Guerra` non ha un collegamento (nella versione di Wikipedia attualmente a disposizione) con il concetto *Tonino Guerra* il

secondo articolo non verrà restituito. Nel momento però in cui serrà inserito questo senso per l'ancora e la commonness sarà maggiore di 0.05, questa news verrà erroneamente recuperata dal sistema.

Ricerca basata su TAGME**.** Anche in questo caso il tipo di ricerca in esame restituice i risultati migliori: si è visto in precedenza che nella disambiguazione della prima news compare il concetto *Tonino Guerra* e con un rho maggiore di 0.05 per cui la notizia viene recuperata.

Nel secondo caso invece questo concetto non è presente tra le annotazioni, e quelle identificate hanno un valore di relatedness con il concetto cercato molto basso per cui questa notizia non viene restituita.

3.4 Discussione dei risultati

Valori per ρ e δ

Nei paragrafi precedenti si è accennato alla necessità di trovare i valori per le soglie ρ e δ. A questo scopo sono state fatte alcune prove i cui risultati in termini di F1 sono riportati nel grafico in fig. 3.10.

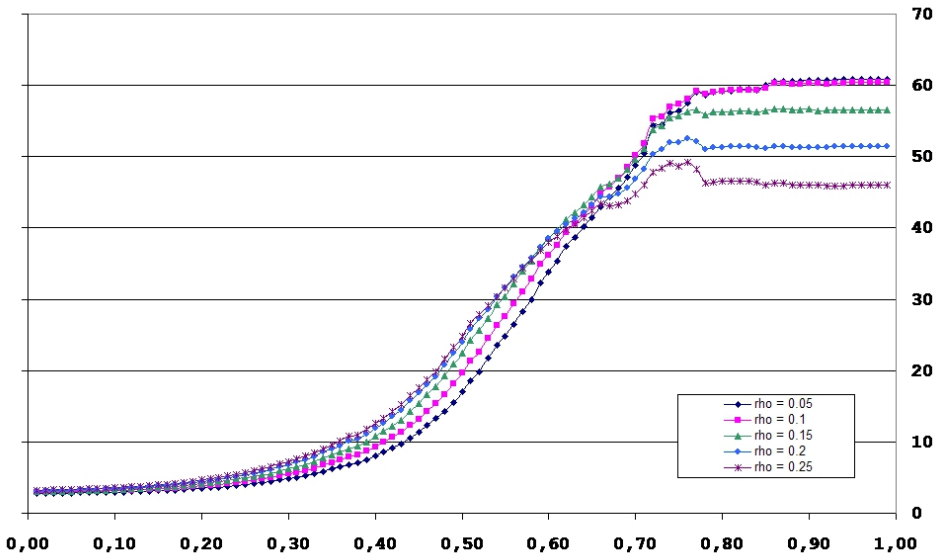

Figura 3.10: Grafico che mostra i valori di F1 in relazione alle soglie di rho (in leggenda) e relatedness (asse x).

Si osserva che ad alti valori di δ (ascisse) corrispondono alte performance in termini di F1. I risultati migliori sono quindi dati quando si restituiscono le news contenenti un concetto che abbia una relazione molto stretta con quello cercato. Si può comunque notare che le varie curve si assestano presentando solo minime variazioni percentuali per δ che supera un valore approssimabile allo 0.75. Si vede anche che valori di ρ bassi favoriscono il compito di ricerca: questo significa che è vantaggioso avere molte annotazioni (anche se alcune sono potenzialmente poco o affatto rilevanti) e quindi che le annotazioni più incerte in termini di correttezza (quelle con un rho basso) apportano informazioni utili alla ricerca.

Nel grafico in figura 3.11 si può notare che ρ continua ad avere una funzione di bilanciamento tra Precision e Recall: questo perché quanto più numerosi sono i concetti annotati in una news tanti più saranno quelli vicini a quello cercato (con una relatedness $> \delta$) e quindi tanto più facilmente la notizia verrà recuperata. Diminuire ρ significa identificare come rilevanti in un testo un numero maggiore di concetti: utilizzare però un valore troppo

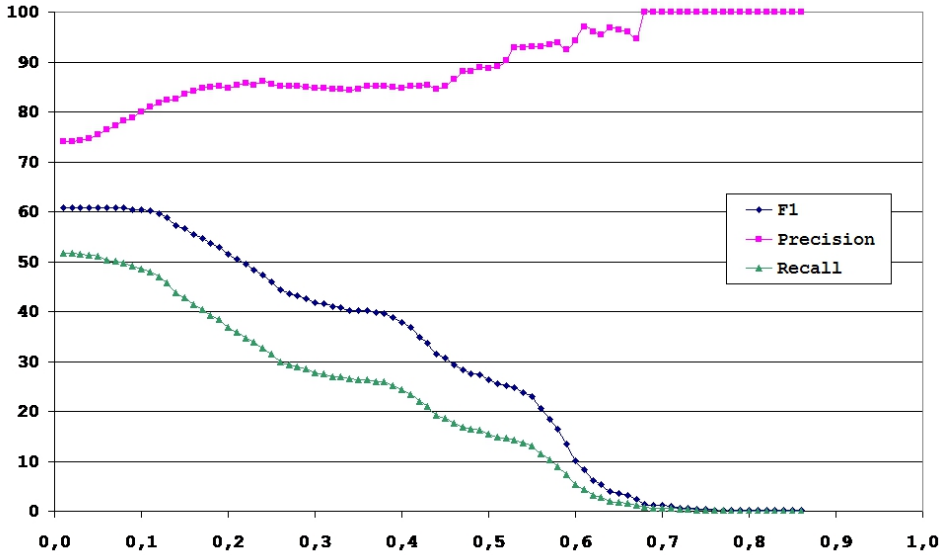

Figura 3.11: Variazioni della percentuale di F1, Precision e Recall (in ordinata) al crescere di ρ (in ascissa)

piccolo per questa soglia porta a includere tra le annotazioni anche concetti non interessanti inficiando quindi anche il task del recupero di news. Le notizie estratte per una query sarebbero molte, in buona parte tuttavia non inerenti.

Il bilanciamento tra Precision e Recall nella selezione delle notizie è condizionato anche dalla soglia di relatedness δ.
Nelle figure 3.12 e 3.13 sono riportate per 5 valori di rho le variazioni rispettivamente della Precision e della Recall al crescere della soglia di relatedness.

Come si vede a un aumento di δ corrisponde l'incremento dei livelli di Precision, dato dal fatto che si circoscrive in maniera sempre più dettagliata il campo di interesse. Diminuendo la soglia di relatedness infatti vengono recuperate informazioni su entità sempre meno vicine a quella cercata.

In maniera analoga ma contraria si assiste con l'aumentare di δ a una riduzione della Recall. Ciò indica che alcune notizie sono interessanti per una ricerca pur non avendo una relazione molto stretta con l'entità cercata.

In entrambi i gruppi di curve si assiste a un graduale assestarsi nella parte finale, con un cambiamento evidente della curva per valori di δ che si approssimano allo 0.75.

Nei grafici su riportati si nota anche come ρ attutisca la crescita e la diminuzione rispettivamente di Precision e Recall all'aumentare di δ. Questo perché i concetti annotati nel testo con rho basso sono meno precisi di quelli

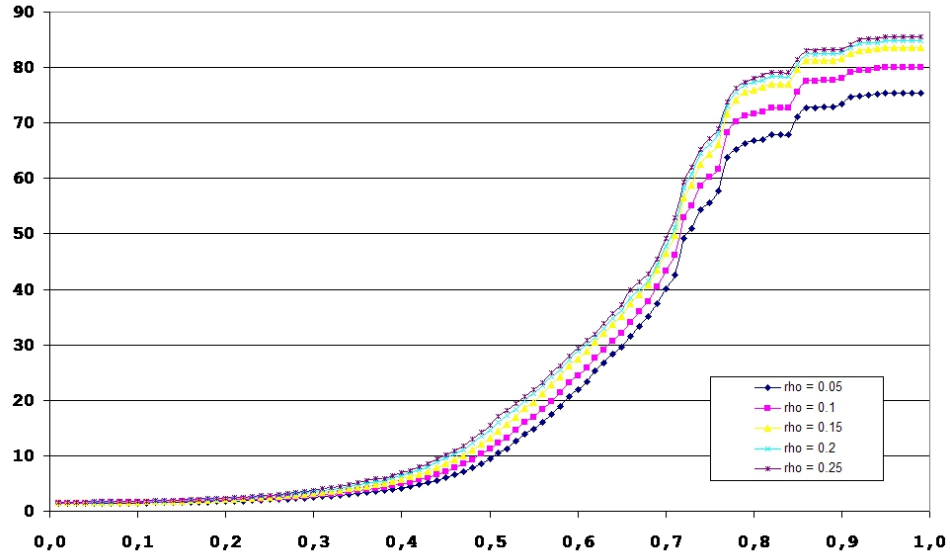

Figura 3.12: Variazioni della Precision al crescere del valore della soglia di relatedness per 5 valori di rho

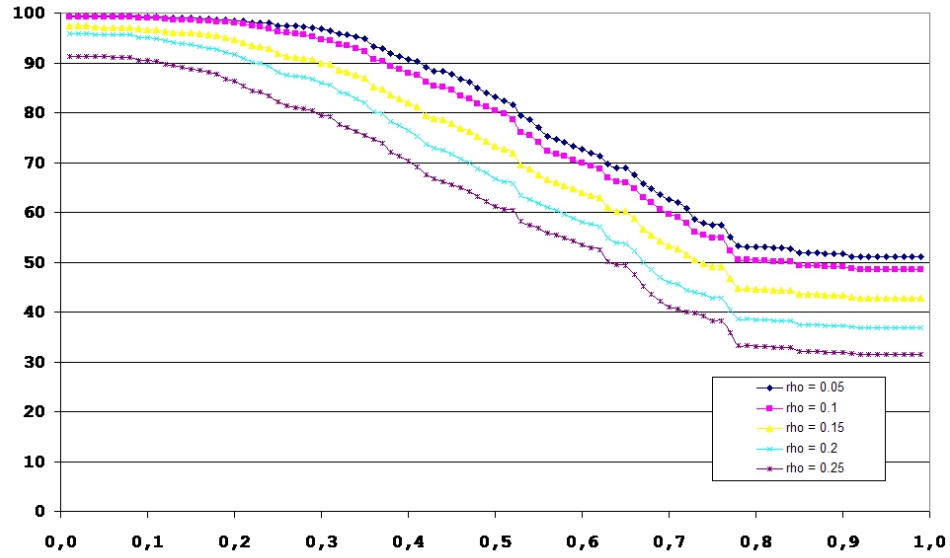

Figura 3.13: Variazioni della Recall al crescere del valore della soglia di relatedness per 5 valori di rho

annotati utilizzando una soglia più alta. Anche aumentando il grado di relatedness necessaria dunque le news estratte in caso di rho basso saranno meno precise di quelle estratte con valori di rho più alti. Il discorso opposto viene fatto invece per la Recall che di conseguenza aumenterà.

Nel grafico 3.14 si può vedere l'andamento del rapporto tra Precision (asse delle ascisse) e Recall (ordinate) variando la relatedness. Si osservi come aumentando il valore di rho la curva Precision/Recall si allontana sempre più dalla curva desiderabile.

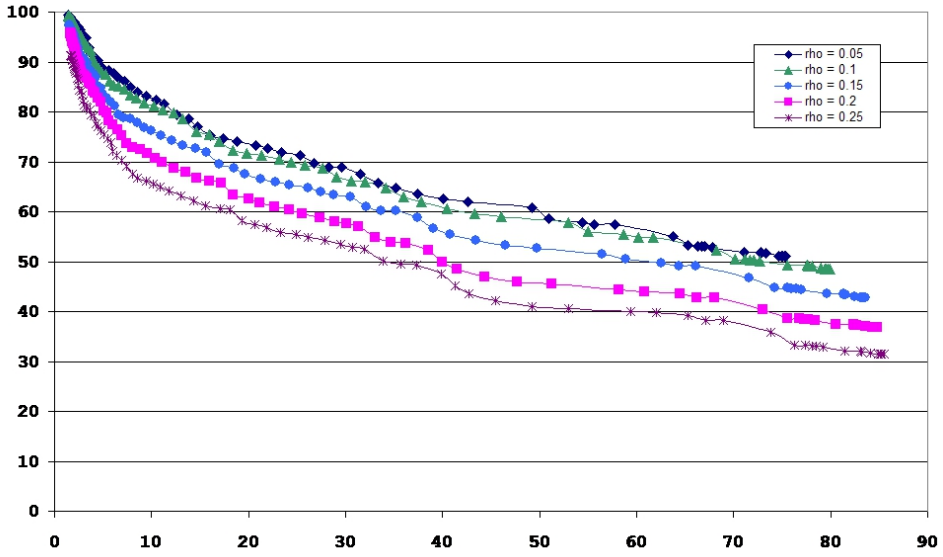

Figura 3.14: Rapporto tra Precision e Recall al variare della relatedness per 5 valori di rho

In base a quanto fino ad ora osservato i valori di ρ e δ possono essere utilizzati per dare un ordinamento alle news in fase di visualizzazione. Si vede infatti che per δ costante la precisione delle risposte è tanto maggiore quanto maggiore è il grado di sicurezza dell'annotazione per la quale è stata recuperata la news. Questo grado di sicurezza è rappresentato dal valore ρ che viene calcolato per le singole ancore nella fase di pruning. La visulizzazione dei risultati potrebbe essere gestita come segue:

- si ordinino per δ le notizie;
- in base alla loro distribuzione si renda discreto questo valore;
- per ogni soglia si segua un ordinamento basato sul ρ dell'annotazione disambiguata con un concetto vicino a quello cercato.

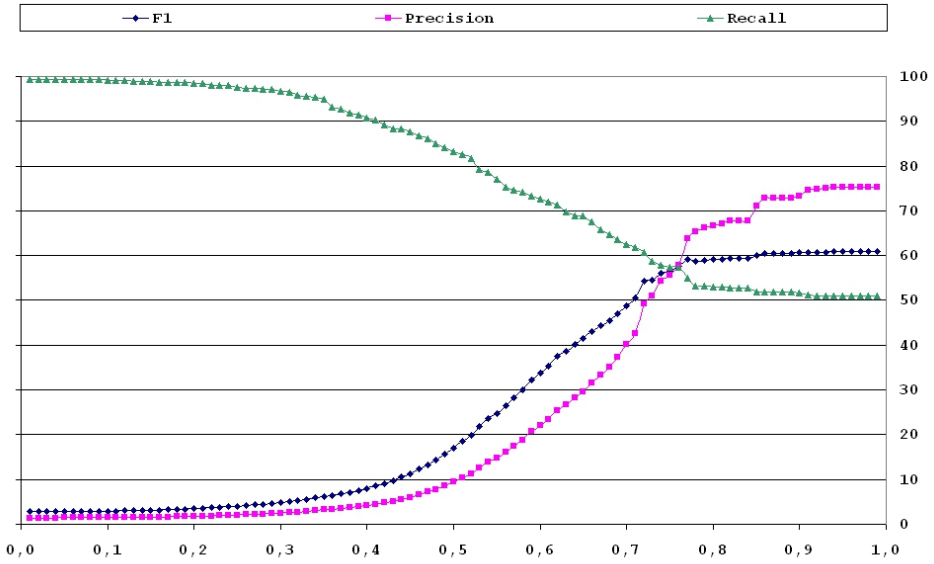

Figura 3.15: Precision, Recall e F1 per l'intero corpus al variare della soglia di relatedness. Leggenda valida per tutta la sequenza di immagini.

Valori per le singole sezioni

Nel presentare i risultati ottenuti sulle singole sezioni si è notato che alcuni di questi erano molto lontani da quelli ottenuti sull'intero corpus. Per cercare di comprendere le ragioni di tale comportamento si inseriscono anche i grafici dell'andamento di Precision, Recall e F1 per ogni sezione al variare della soglia δ (figure 3.15 - 3.26) e le tabelle con i risultati ottenuti per i tre modelli di estrazione (tabelle 3.6 e 3.7).

Tra i grafici sono evidenti alcune macro-differenze. La prima riguarda la sezione *sport*. Qui a differenza di tutti gli altri grafici le curve di precision e recall non si intersecano: particolarmente evidente è la lenta crescita della precision al variare di δ, tanto che quando tale soglia raggiunge il massimo valore la precision non tocca il 50%. Ciò significa che ogni 5 news correttamente recuperate altrettante sono errate. A riportare nella media i valori della F1 c'è la recall che decresce con ugual lentezza. Si noti a questo proposito la bassissima recall (6,2%) che questa sezione ha nella ricerca testuale. Una spiegazione di questo valore sta nel fatto che tra le query con le quali si è annotato il corpus quella più inerente il tema sportivo è "Nazionale di calcio". Una bassa recall indica quindi che negli articoli sportivi questa squadra è stata chiamata in modi diversi (non di rado facendo riferimento all'allenatore oltre che al colore della maglia).

Valori decisamente anomali sono quelli relativi alla sezione *24 ore*. Qui infatti precision e recall ottengono degli ottimi risultati (superiori entrambi

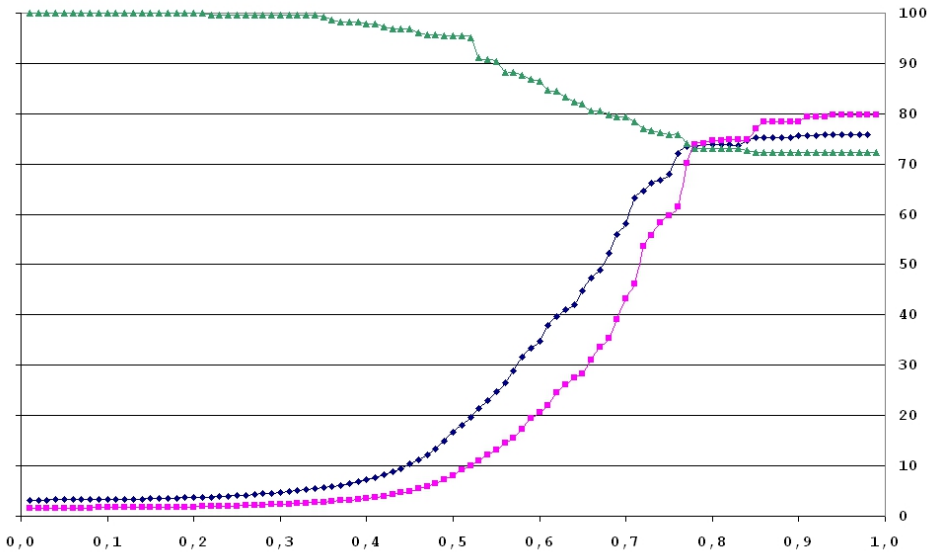

Figura 3.16: Precision, Recall e F1 al variare di δ nella sezione 24 ORE

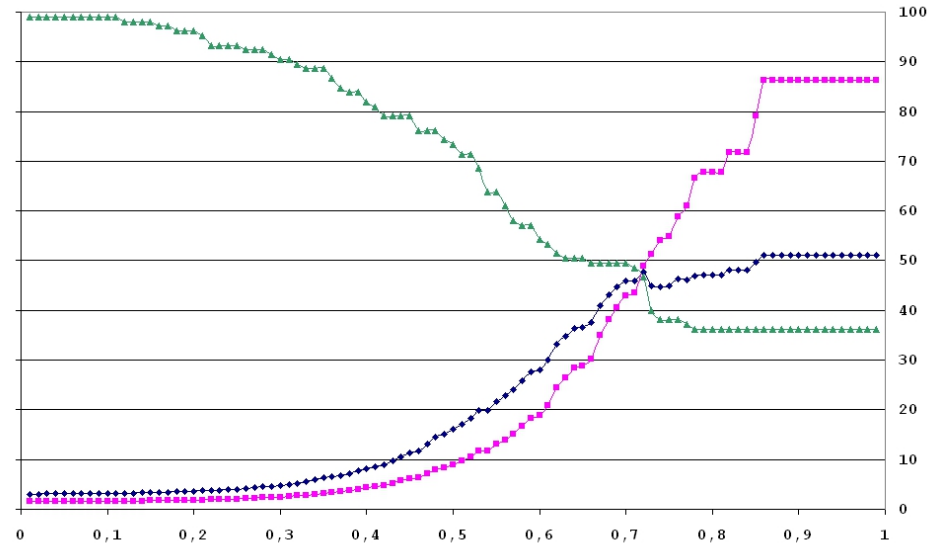

Figura 3.17: Precision, Recall e F1 al variare di δ nella sezione CRONACA

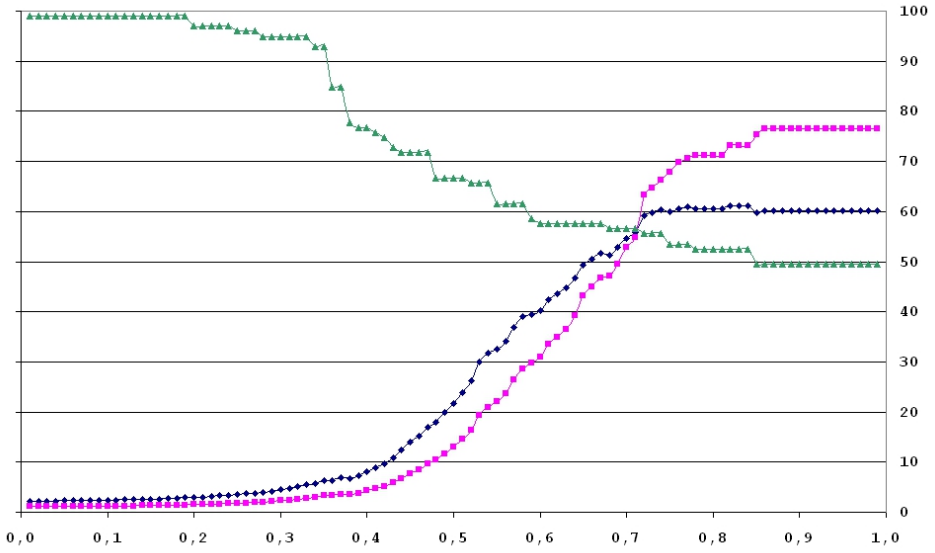

Figura 3.18: Precision, Recall e F1 al variare di δ nella sezione CULTURA

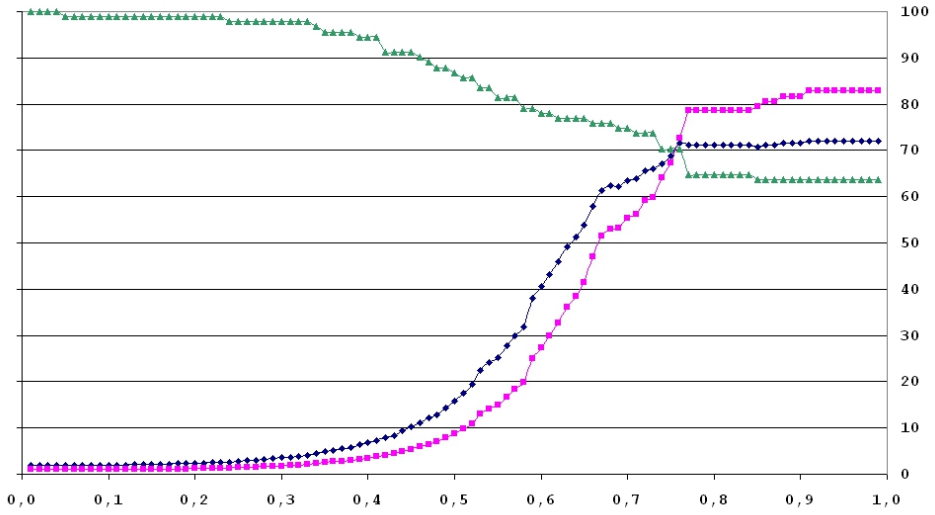

Figura 3.19: Precision, Recall e F1 al variare di δ nella sezione ECONOMIA

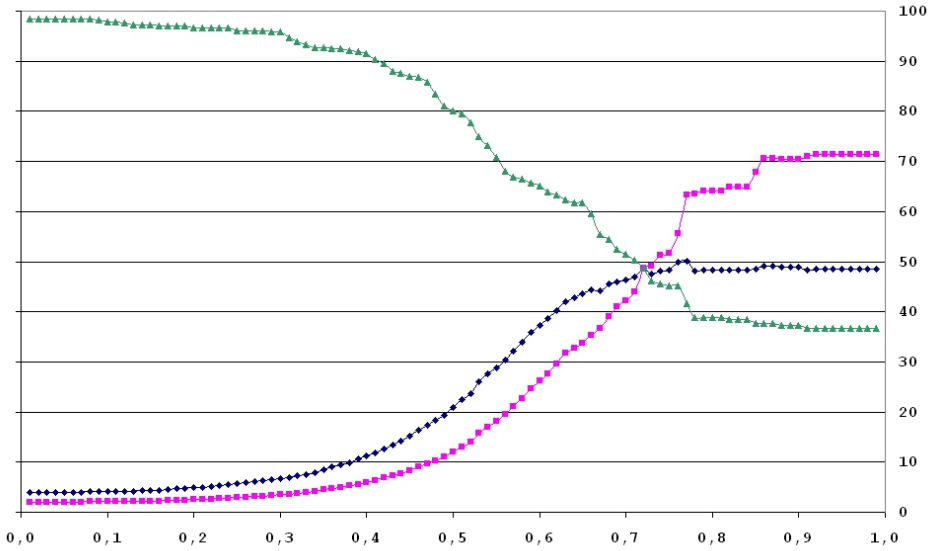

Figura 3.20: Precision, Recall e F1 al variare di δ nella sezione HOMPAGE

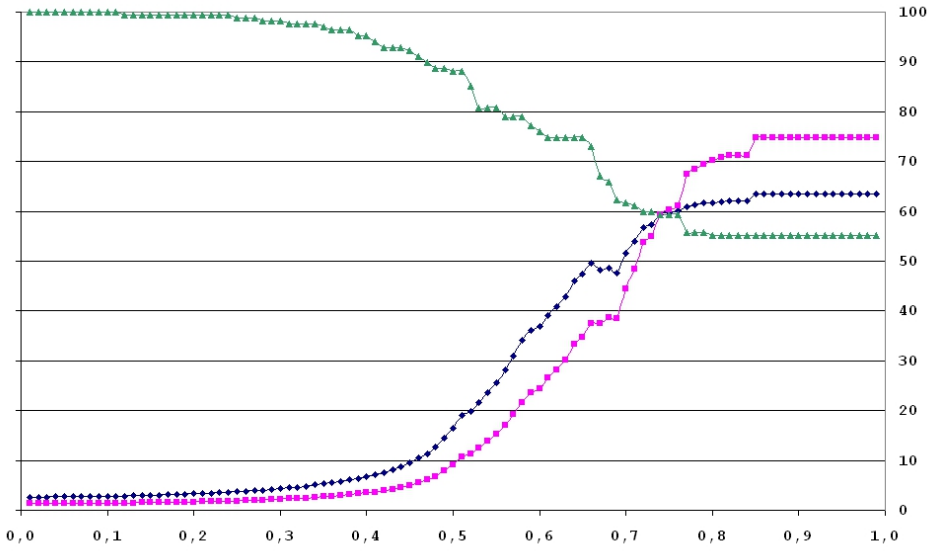

Figura 3.21: Precision, Recall e F1 al variare di δ nella sezione MONDO

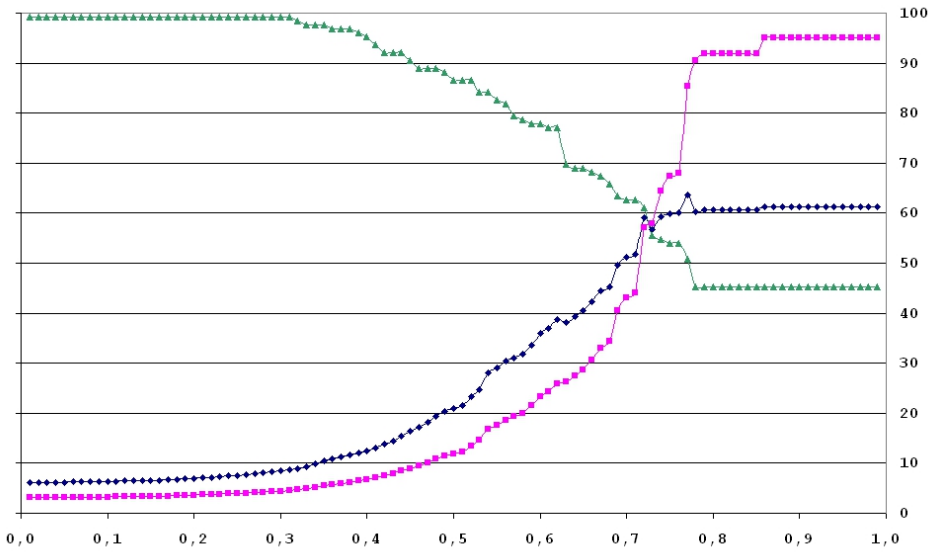

Figura 3.22: Precision, Recall e F1 al variare di δ nella sezione POLITICA

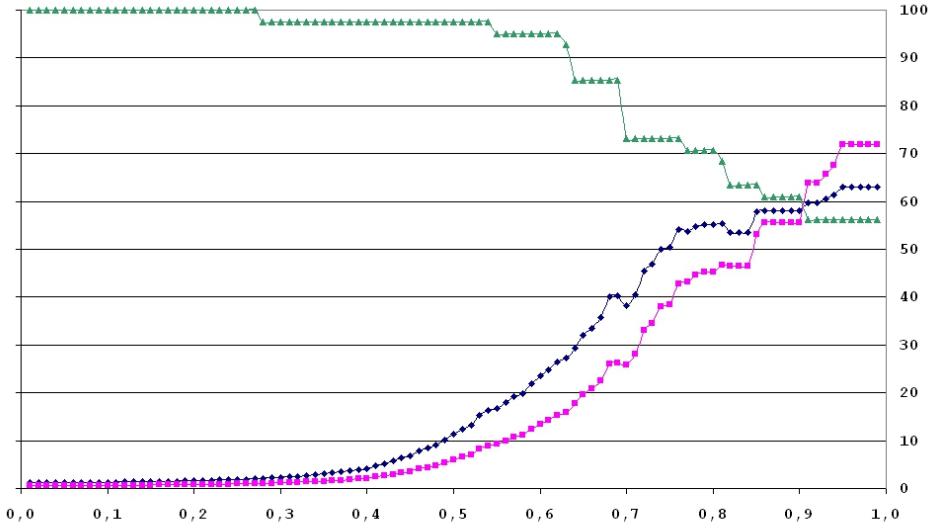

Figura 3.23: Precision, Recall e F1 al variare di δ nella sezione SCIENZE

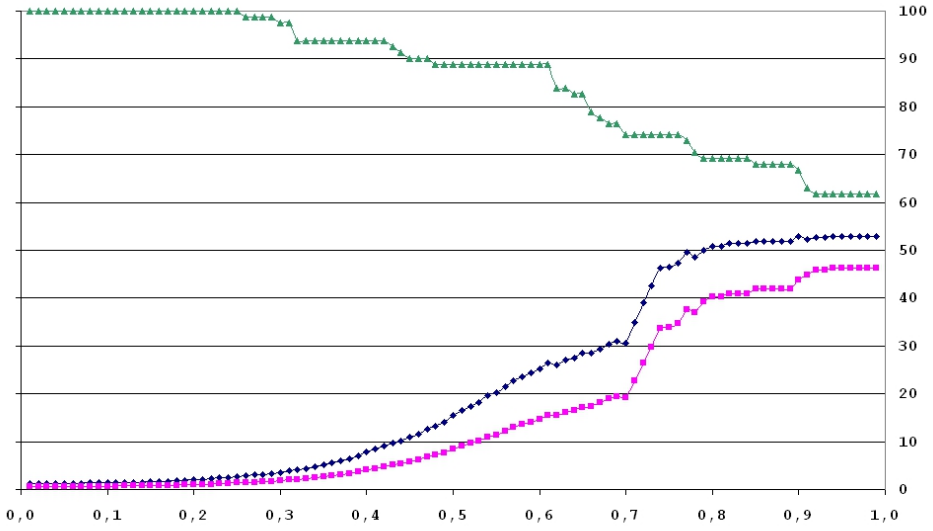

Figura 3.24: Precision, Recall e F1 al variare di δ nella sezione SPORT

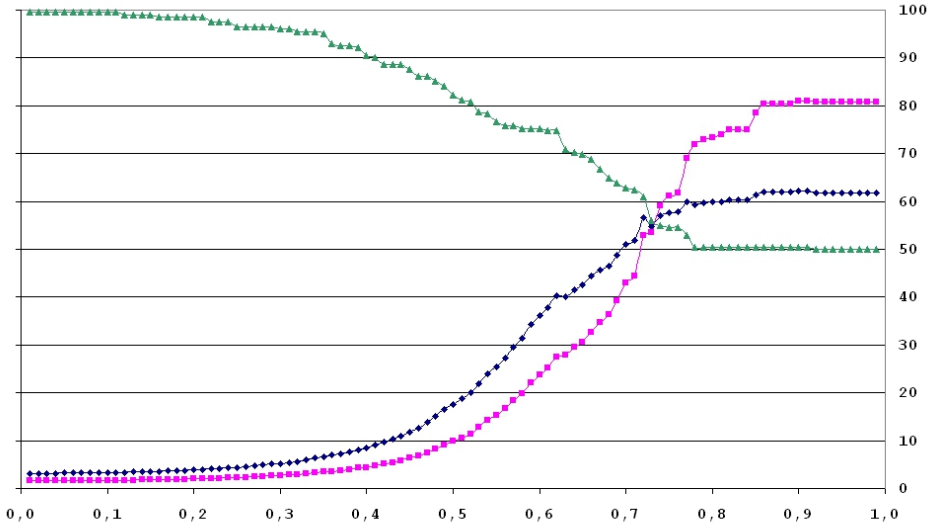

Figura 3.25: Precision, Recall e F1 al variare di δ nella sezione TOPNEWS

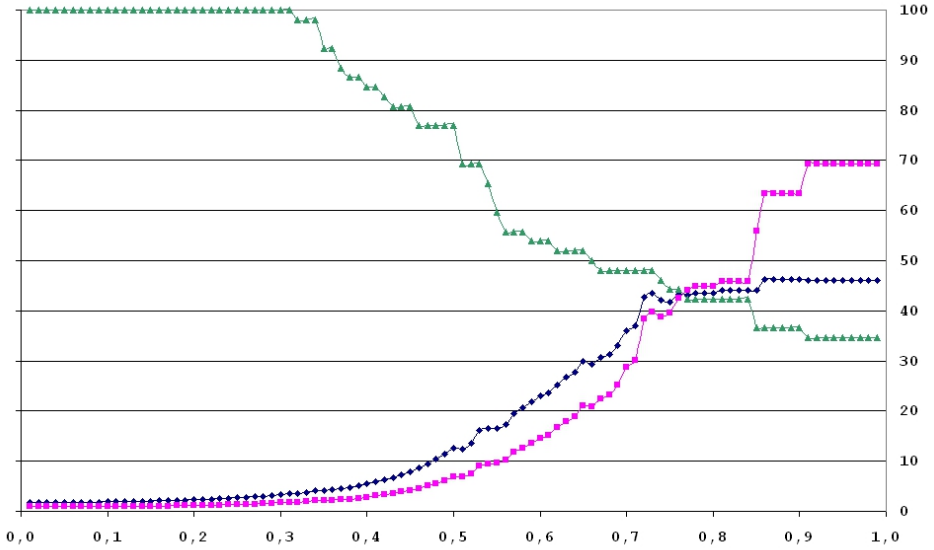

Figura 3.26: Precision, Recall e F1 al variare di δ nella sezione Altro

SEZIONI	**Precision**	**Recall**	**F1**
24 ore	79.7	72.3	75.8
Cronaca	86,4	36,2	51
Cultura e Spettacolo	76,5	49,5	60,1
Economia	82,8	63,7	72
Home	71,3	36,7	48,5
Mondo	74,8	55,1	63,4
Politica	95	45,2	61,3
Scienza e Tecnologia	71,8	56,1	63
Sport	46,3	61,7	52,9
Topnews	80.8	50.5	61.8
Altro	69.2	34.6	46.1
Tutte le sezioni	75.4	51	60.9

Tabella 3.6: Tabella riportante i valori di Precision, Recall e F1 ottenuti al variare della soglia δ sulle singole sezioni

	Anchor Parser			Testuale		
SEZIONI	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
24 ore	79.1	72.3	75.6	88.3	61.3	72.4
Cronaca	92.7	36.2	52.1	78.6	10.5	18.5
Cultura	62.9	25.7	36.4	80.0	10.5	18.6
Economia	85.3	63.7	73.0	90.3	30.8	45.9
Homepage	72.3	36.1	48.2	85.3	19.3	31.4
Mondo	79.3	55.1	65.0	75.7	16.8	27.5
Politica	93.3	44.4	60.2	100.0	31.0	47.3
Scienza	67.6	56.1	61.3	76.9	24.4	37
Sport	44.3	48.1	46.2	33.3	6.2	10.4
Topnews	84.8	47.0	60.5	92.7	18.8	31.3
Altro	80.8	40.4	53.8	100.0	15.4	26.7

Tabella 3.7: Tabella riportante i valori di Precision, Recall e F1 ottenuti utilizzando il metodo basato sull'anchor parsing e quello testuale al variare della soglia δ sulle singole sezioni del corpus

al 72%). Abbiamo, come si vede nel grafico 3.16, una curva della recall che scende molto lentamente tanto che il minimo raggiunto intorno a $\delta = 0.85$ è pari a 72,3%. Anche per quanto riguarda la ricerca testuale si nota un'alta recall mostrando che questa sezione tende a far riferimento alle entità che compongono la query, chiamandole (6 volte su 10) nella forma indicata nel corpus.

Si osserva inoltre che i valori della precision si attestano nella maggior parte dei casi tra il 70 e l'80%. Nel caso della *politica* tuttavia questo raggiunge il 95%. Il variare di questo valore in funzione della soglia di relatedness mostra che al passaggio da $\delta = 0.7$ a $\delta = 0.8$ si ha un'impennata della curva della precision che passa dal 40 all'92%. Questo valore viene tuttavia controbilanciato dalla recall.

Aggiornamento di Wikipedia

Si è ritenuto opportuno introducendo i risultati ottenuti mettere subito in rilievo che viene utilizzata un versione di Wikipedia antecedete di alcuni mesi alla maggior parte delle news nel corpus.

Una delle problematiche maggiori legate all'estrazione di news è data proprio dal gap temporale e quindi dalla mancanza di informazioni adeguate da parte del sistema sulla realtà contemporanea. La capacità di trovare le ancore del testo, di disambiguarle e infine di eliminare quelle meno rilevanti dipende esclusivamente dai concetti e dalle relazioni inserite nella base di conoscenza cui TAGME fa riferimento. Quanto più datata questa è rispetto ai testi da analizzare tante più difficoltà il software troverà nel risolvere il task.

Bisogna inoltre tener presente che Wikipedia, pur continuando ad aggiornarsi molto rapidamente, conserva una caratteristica tipica delle enciclopedie, ossia quella di conservare una conoscenza in buona parte storica: questo perché i nuovi dati vengono il più delle volte aggiunti a quelli vecchi che quindi, come giusto che sia, resistono nel tempo implicando dunque la presenza di una percentuale alta di relazioni non più attuali tra concetti.

Troviamo un esempio nei cambiamenti politici degli ultimi mesi, in particolare nel passaggio dal governo Berlusconi al governo tecnico. Quanto accaduto ha gradualmente modificato le relazioni di Mario Monti (persona e quindi concetto in Wikipedia) spostandole da quelle tipiche di un economista, accademico e commissario europeo per 4 anni, per allargarle a quelle che un presidente del consiglio e ministro dell'economia oggi può avere.

I cambiamenti nelle relazioni sono dovuti agli aggiornamenti inseriti dagli utenti dell'enciclopedia e quindi richiedono un certo tempo durante il quale avranno sempre notevole risalto le relazioni relative alla precedente situazione in questo caso politica. Il valore della relatedness tra il concetto "Mario Monti" e "Presidenza del Consiglio dei Ministri della Repubblica italiana" quindi assume valori differenti nel tempo andando perciò a modificare i risultati restituiti dal software.

Interessante sarebbe l'analisi delle variazioni della relatedness nel tempo e quindi osservare il funzionamento del modello di estrazione nelle news.

Capitolo 4

CONCLUSIONI

Durante questa tesi si è provveduto a:

- Creare un modello per l'estrazione di news contenute in feed RSS sfruttando l'annotazione fornita da TAGME;
- Creare un corpus di news per la valutazione del modello;
- Annotare manualmente il corpus;
- Creare dei modelli di ricerca alternativi per il confronto dei risultati;
- Implementare il modello di ricerca in modo da permettere l'estrazione dei feed attualmente presenti on-line.

La prima evidenza ottenuta grazie a questo lavoro è l'efficacia dell'utilizzo dell'annotazione fornita da TAGME nel processo di recupero di news rilevanti per un argomento. Il modello proposto infatti ottiene per la F1 un risultato migliore di 2 punti percentuali rispetto ai suoi competitor che lavorano sul livello testuale.

Un vantaggio evidente si ottiene sulle query ambigue: in questi casi infatti il semplice match restituisce le news in cui la parola compare indipendentemente dal senso che ne riveste.
Malgrado la brevità del testo a disposizione TAGME è capace di disambiguare le ancore rilevate in modo da recuperare solo le news per le quali è stato annotato un concetto la cui similarità con quello cercato fosse tale da richiederlo.

Inoltre il tipo di ricerca proposto restituisce i risultati associandoli a due valori (la sicurezza dell'annotazione e la similarità che questa ha con la query) che possono essere utilizzati come indici per l'ordinamento delle news nell'output.

Scenari applicativi Vista la grande quantità di informazioni a disposizione uno degli scenari applicativi per i quali risulterebbe molto utile l'esplorazione semantica di feed RSS e la selezione di news interessanti per un concetto cercato è dato dai motori di ricerca e da filtri. Esistono infatti software in grado di selezionare (o eliminare) news che rispondono a determinate caratteristiche. Tuttavia questi sono fortemente limitati dalla loro incapacità di superare la forma rappresentata dalla componente testuale.

Analogamente questo modello potrebbe essere utilizzato come base per un sistema di alerting in cui non si selezionano più parole chiave ma concetti e una distanza che definisce l'intorno semantico sul quale essere informati.

Lo studio fin qui proposto è sfociato nel prototipo di applicazione che permette di selezionare in base al contenuto le notizie inserite in feed indicati dall'utente (vedi cap. 5).

A quanto implementato potrebbero essere apportati cambiamenti tesi a migliorare l'usabilità e la flessibilità della struttura già creata. In particolare sarebbe interessante:

- fornire una veste grafica sostituendo l'interazione da prompt dei comandi con un'interfaccia web;
- dare all'utente la possibilità di scegliere il livello di relatedness che i concetti nelle news recuperate devono avere con quelli cercati;
- visualizzare i risultati, associando al file XML un XSL (Extensible Stylesheet Language) che permetta anche di rispettare l'ordinamento suggerito dai valori di rho e relatedness;
- aggiungere una sezione per la memorizzazione dei feed annotati in modo da limitare quanto più possibile il tempo di risposta e le risorse utilizzate. Il numero di news da processare infatti è notevole e sebbene TAGME riesca a compiere velocemente il task è preferibile utilizzare dello spazio in memoria.

Limite della rete di Wikipedia: proposta di un grafo temporale. Purtroppo l'utilizzo di TAGME su testi estratti da news ha un inconveniente dato dalla necessità di aggiornare frequentemente il grafo scaricato da Wikipedia. Rispetto alle altre tipologie di testo infatti le news presentano una peculiarità che è data principalmente dal loro utilizzo: quello di informare sui nuovi avvenimenti. Accade dunque che sia proprio la stampa a rendere di pubblico dominio alcune entità (ad esempio persone) e ancora più frequentemente a creare nuovi legami. Per chiarire ciò che si intende basta osservare il caso del naufragio del 13 gennaio della Costa Concordia, episodio per il quale due giorni dopo è stata creata una pagina apposita che arricchisce la rete non solo di un nuovo concetto ma anche con nuovi collegamenti.

Solo attraverso l'aggiornamento del grafo dunque il software può arrivare a conoscere questi dati.

Si è inoltre osservato come le informazioni contenute in Wikipedia siano (come ci si aspetta da una enciclopedia) in buona parte storiche poichè tipicamente i nuovi dati si aggiungono a quelli vecchi. Questo aspetto tende a rendere sbilanciata la rete dando poco risalto all'attualità. Sarebbe interessante osservare la distribuzione nel tempo di link nuovi alle pagine, per valutare l'importanza nella disambiguazione di questi dati in una visione temporale della rete. Un esempio può renderne chiara la motivazione: si immagini di dover calcolare la relatedness tra il concetto "Costa crociere" e "Giglio". Fino al 13 gennaio di quest'anno tale valore era basso: la compagnia di navigazione infatti avrà avuto un valore di similarità alto con altre compagnie di navi da crociera o con centri benessere; analogamente l'isola del Giglio sarà stata nominata speso insieme alle altre isole italiane, alla regione Toscana, a altri posti turistici e naturalistici. L'incidente della "Concordia" ha portato in una notte ad avere un numero molto elevato di news in cui i due concetti si trovano affiancati. Contemporaneamente saranno aumentate le pagine di Wikipedia contenenti link sia alla "Costa Corciere" che all'"Isola del Giglio". Se tuttavia questi concetti avevano un numero già elevato di in-link (soprattutto se molti tra questi sono condivisi con pochi altri concetti) i nuovi collegamenti non saranno sufficienti a far aumentare la relatedness a livello tale da essere rispondente alla relatà giornalistica. Approfondire il tema studiando l'andamento dei nuovi link nel tempo (quindi in relazione agli avvenimenti) potrebbe portare ad attribuire un peso maggiore agli ultimi collegamenti creati in modo da attutire la rilevanza che la storia ha nella conoscenza enciclopedica.

Capitolo 5

HOW TO

Per poter fare uso del programma che implementa l'estrazione di news da feed basandosi sull'annotazione semantica è necessario disporre dell'accesso al computer Brie del Dipartimento di Informatica.
Attualmente per lanciare il programma bisogna:

- Aprire la shell dei comandi;
- Posizionarsi nella cartella /l/disc3/home/ilaria/tagme/codice/tms
- Scrivere `ant run -Dclass = interfaccia.RssFilter`

Seguono le fasi della ricerca:

PASSO 1. Cambiamento del percorso del file di output
Digitando `n` si lascia che il file di output venga inserito nella cartella inserita di default (/l/disc3/home/ilaria/news/rssDir/result/);
Digitando `s` viene chiesto di inserire il percorso desiderato.

PASSO 2. Scelta dei feed in cui cercare
Viene chiesto di inserire i link ai feed separandoli da uno spazio (ad es. `http://feeds.ilsole24ore.com/c/32276/f/566660/index.rss` `http://xml.corriereobjects.it/rss/homepage.xml` `http://www.ansa.it/web/ansait_web_rss_homepage.xml`) o di utilizzare (digitando `1`) quelli di default che sono:

- Il corriere - homepage (http://xml.corriereobjects.it/rss/homepage.xml);
- Il sole 24 ore - homepage (http://feeds.ilsole24ore.com/c/32276/f/566660/index.rss);
- La Repubblica - homepage (http://rss.feedsportal.com/c/32275/f/438637/index.rss);

- Il Messaggero - homepage
  (http://www.ilmessaggero.it/rss/home.xml)
- Ansa - homepage
  (http://www.ansa.it/web/ansait_web_rss_homepage.xml);
- La Stampa - homepage
  (http://www.lastampa.it/redazione/rss_home.xml);
- Adnkronos - prima pagina
  (http://rss.feedsportal.com/c/32375/f/448341/index.rss);
- TgCom24 - homepage
  (http://www.tgcom24.mediaset.it/rss/homepage.xml);
- Il Quotidiano
  (http://quotidianohome.feedsportal.com/c/33327/f/565662/index.rss);
- Il fatto quotidiano
  (http://www.ilfattoquotidiano.it/feed/);
- L'Unità - homepage
  (http://www.unita.it/cmlink/feed-homepage-1.244567);

PASSO 3. Inserimento una query
Viene chiesto di inserire una query sotto forma di parole chiave (ad es. `Parlamento UE`

PASSO 4. Selezione il tipo di ricerca
Si chiede di scegliere tra:

- ricerca tesutale (digitare `1`)
- ricerca semantica (digitare `2`)

Digitando `1` viene generato il file XML (così come descritto in 3.1.3) dal nome *textual_<n>.xml* (dove n rappresenta il numero delle query che sono state fatte nella sezione). Nel file sono riportate tutte le news che contengono almeno una delle parole della query non necessariamente adiacenti o nell'ordine di inserimento.

Digitando `2` l'interazione continua come segue:

PASSO 5. Scelta dei concetti da cercare
Vengono mostrati titolo e descrizione delle pagine di Wikipedia cui la query può fare riferimento. Se possibile vengono proposti i concetti selezionati da TAGME altrimenti ci si trova al PASSO 6. Si chiede quindi di scegliere se:

- cercare tutti i concetti mostrati (digitare `1`)
- selezionare alcuni tra i concetti mostrati (digitare `2`)
- mostrare altri concetti (digitare `3`)

Digitando `1` si passa al PASSO 7

Digitando `2` viene chiesto di inserire i numeri associati ai concetti che interessanto separandoli da spazio, quindi si passa al PASSO 7;

Digitando `3` l'interazione continua come segue:

PASSO 6. Scelta dei concetti da cercare in una rosa più ampia

Vengono mostrati titolo e descrizione delle pagine di Wikipedia cui la query può fare riferimento ordinandoli per la probabilità di essere stati scritti utilizzando quelle parole (ordinandoli quindi per la loro commonness con l'ancora). Si chiede quindi di inserire i numeri associati ai concetti che interessanto separandoli da spazio.

PASSO 7. Conferma dei concetti selezionati

Vengono mostrati i titoli relativi ai concetti selezionati e viene chiesto di confermare (scrivendo s) o non confermare (scrivendo n) la nuova query composta da concetti.

Digitando `n` si torna al PASSO 5;

Digitando `s` si prosegue come segue:

PASSO 8. Selezione del tipo di ricerca per concetto

Viene chiesto di scegliere se:

- cercare tutte le possibili forme attraverso il quale è possibile esprimere i concetti cercati (digitare `1`)
- cercare le news utilizzando l'annotazione fornita da TAGME (digitare `2`)

Digitando `1` viene generato il file *extendedTextual_<n>.xml* che riporta tutte le news che contengono un'espressione (ancora) che ha tra i suoi possibili sensi uno di quelli cercati.

Digitando `2` viene generato il file *tagme_<n>.xml* che riporta i risultati dell'estrazione del modello presentato in 3.

Entrambi questi file sono strutturati come descritto in 3.1.3 e per entrambi n rappresenta un numero identificativo per la query nella sessione.

PASSO 9. Selezione del passo successivo

Se per la query inserita non sono stati compiuti tutti i tipi ricerca verrà chiesto se:

- cercare la query già inserita con un'altro tipo di ricerca (digitare `1`)
- fare una nuova query (digitare `2`)
- uscire (digitare `0`)

Digitando `1` viene chiesto il tipo di ricerca desiderato tra quelli rimanenti
Digitando `2` si ritorna al PASSO 3.
Digitando `0` termina l'applicazione.

Bibliografia

[1] Mihalcea R., Csomai A., *Wikify! Linking Documents to Encyclopedic Knowledge*, Proc. ACMCIKM, 233-242, 2007.

[2] Milne D., Witten I. H., *An effective, low-cost measure of semantic relatedness obtained from Wikipedia links.* In Proc. ACM CIKM, 233-242, 2007.

[3] Milne, D., Witten, I.H. *Learning to link with Wikipedia.* In Proceeding of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, October 26-20, 2008(pp. 509-518).

[4] Cilibrasi R., Vitanyi P., *The Google Similarity Distance*, IEEE Transactions on Knowledge and Data Engineering, v.19 n.3, p.370-383, March 2007.

[5] Kulkarni S., Singh A., Ramakrishanan G., Chakrabarti S., *Collective annotation of Wikipedia entities in web text.* In Proc ACM KDD, 457-466, 2009.

[6] Ferragina P., Scaiella U., TAGME*: On-the-fly annotation of short text fragments (by Wikipedia entities).* In Proc. ACM CIKM 2010 (poster).

[7] Ferragina P., Scaiella U., *Fast and accurate annotation of short texts with Wikipedia pages*, IEEE Software Special issue on "Algorithms and Today's Practitioner".

[8] Ferragina P., Gulli A.,*A personalized search engine based on web-snippet hierarchical clustering.* In Proc.WWW, 801-810 2005

[9] Dulli S., Polpettini P., Trotta M. (a cura di), *Text Mining: teoria e applicazioni*, Franco Angeli, Milano, 2004.

[10] Manning C.D., Schülze H., *Collocation* in *Foundations of Statistical Natural Language Processing*, The MIT press, Cambridge, Massachusetts, 1999.

[11] Pierazzo E., *La codifica dei testi. Un'introduzione*, Carrocci editore, Roma, 2005.

[12] W3C, *World Wide Web Consortium*: http://www.w3c.org (visitato il 15 maggio 2006)

[13] http://http://tagme.di.unipi.it/

[14] http://www.xul.fr/en-xml-rss.html

[15] RSS Advisory Board: http://www.rssboard.org/

[16] Web Resource: http://web.resource.org/rss/1.0/spec

[17] Traduzione Italiana Specifiche: http://www.specifiche.it/rss/2.0/

[18] RSS World: http://www.rss-world.info/

[19] Dawn Foster, SXSW Hacking RSS: Filtering & Processing Obscene Amounts of Information http://www.slideshare.net/geekygirldawn/sxsw-hacking-rss-filtering-processing-obscene-amounts-of-information

[20] http://mashable.com/2008/10/30/slow-feed-movement-rss/

[21] http://www.readwriteweb.com/archives/6_ways_to_filter_your_rss_feeds.php

[22] http://pipes.yahoo.com/pipes/

[23] http://feedrinse.com/