Università degli Studi di Pisa
Facolta di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Magistrale in Informatica

Tesi di Laurea Magistrale

Analisi e miglioramento delle performance di un annotatore testuale

Candidato
Francesco Mele

Relatore
Prof. **Paolo Ferragina**
Dott. **Francesco Piccinno**

Controrelatore
Prof. **Francesco Romani**

Anno Accademico 2015/2016

Indice

Introduzione **3**

1 Annotatori Semantici **5**
1.1 Dal sistema classico all'annotazione testuale 5
1.2 Problematiche del paradigma classico 5
1.3 Terminologia 6
1.4 Processo di annotazione 6
1.5 Fasi critiche 7
1.6 Caratteristiche sfruttate da un annotatore 8
1.7 Applicazioni pratiche 8

2 Benchmarking **10**
2.1 Analisi delle prestazioni di un annotatore 10
2.1.1 Problemi di annotazione 10
2.1.2 Match 11
2.1.3 Misure di valutazione 12
2.2 Framework per il benchmarking di annotatori 13
2.2.1 Caratteristiche generali 13
2.2.2 BAT-Framework 14
2.2.3 GERBIL 15
2.2.3.1 Caratteristiche 15
2.2.3.2 Annotatori 15
2.2.3.3 Dataset 16
2.2.3.4 Problemi 17
2.2.3.5 Espandibilità 18
2.2.3.6 Esperimenti 18
2.2.3.7 Risultati 19

3 Il nuovo sistema di Error Reporting **21**
3.1 Terminologia 21
3.2 Caratteristiche principali 22
3.3 Metriche 23
3.4 Statistiche 23
3.4.1 Entity Type stats 23
3.4.2 NE stats 24

3.4.3 Statistiche congiunte Entity Type-NE . . . 25
3.5 Risultati dell'annotazione . . . 26
3.5.1 Visualizzazione globale . . . 26
3.5.2 Singolo documento . . . 26
3.5.3 Statistiche sul dataset . . . 29
3.5.4 Analisi FP/FN/Missing ed Excess . . . 31
3.5.5 Differenza tra due report . . . 33

4 TAGME **34**
4.1 Caratteristiche . . . 34
4.2 Fasi dell'annotazione . . . 34
4.2.1 Creazione degli indici . . . 35
4.2.2 Parsing . . . 35
4.2.3 Disambiguazione delle àncore . . . 35
4.2.4 Potatura delle àncore . . . 36
4.3 Prestazioni . . . 37

5 Da TAGME a WAT **39**
5.1 Fasi dell'annotazione . . . 39
5.1.1 Spotting . . . 39
5.1.2 Disambiguazione . . . 40
5.1.2.1 Algoritmi voting-based . . . 40
5.1.2.2 Algoritmi graph-based . . . 40
5.1.2.3 Ottimizzazione . . . 42
5.1.3 Misure di correlazione . . . 42
5.1.4 Pruning . . . 43
5.2 Prestazioni note di WAT . . . 43
5.3 Considerazioni sull'analisi delle prestazioni . . . 45

6 Il nuovo WAT **47**
6.1 Prestazioni degli annotatori . . . 47
6.1.1 Dataset . . . 47
6.1.2 Analisi delle prestazioni di WAT . . . 47
6.1.3 Valutazione iniziale . . . 48
6.2 La fase di spotting . . . 50
6.3 La fase di disambiguazione . . . 57
6.4 Modifiche a WAT . . . 59
6.4.1 Spotter . . . 59
6.4.2 Disambiguatore . . . 66
6.4.2.1 Disambiguazione con convergenza a step . . . 66
6.4.2.2 Filtro sulle votazioni . . . 68
6.5 Sulle prestazioni del nuovo WAT . . . 69

7 Conclusioni **75**

Bibliografia **78**

Introduzione

Nell'Information Retrieval classico, i documenti composti in linguaggio naturale vengono abitualmente trattati seguendo il paradigma bag-of-words (BOW). In questo tipo di approccio ogni documento è rappresentato come insieme delle parole in esso contenute, ignorando qualunque peculiarità strutturale e di significato insita in quelle parole e, quindi, nel documento stesso. Sebbene questo modo di operare sia sufficientemente efficace in gran parte dei casi, tanto da essere alla base della stragrande maggioranza dei sistemi di Information Retrieval (tra cui i motori di ricerca di prima generazione, quali Altavista), sono evidenti i limiti che lo contraddistinguono, soprattutto in relazione a tipologie di testi particolarmente diffuse su internet come ad esempio i tweet, e più in generale documenti di breve lunghezza e composti in un linguaggio non preciso. Oltre al problema della lunghezza, che mette in difficoltà i principi statistici alla base del paradigma BOW, esistono altri problemi non risolti. Ad esempio è impossibile per un sistema di Information Retrieval basato sul paradigma BOW sapere, in caso di polisemia, a quale dei possibili significati una determinata parola faccia riferimento. Lo stesso vale per i casi di sinonimia, nei quali un termine solitamente più diffuso è sostituito nel testo da un'espressione relativa allo stesso concetto ma la cui correlazione con il concetto stesso non è individuabile. Nell'approccio BOW si perde inoltre qualunque concetto di contesto, essendo le parole che compongono i documenti utilizzate singolarmente a prescindere da dove appaiono nel testo e ciascuna mappata a una dimensione indipendente dalle altre.

Nel corso degli ultimi anni, la comunità dell'Information Retrieval ha concentrato attenzioni sempre maggiori su una nuova categoria di software in grado di porre rimedio ai limiti del paradigma tradizionale che spesso ostacolano la corretta risoluzione di problemi diffusi nell'analisi di Big Data, quali clustering, classificazione, similarità, ecc. Tali software prendono il nome di Annotatori Testuali, e consentono, a partire da testi espressi in linguaggio naturale, di individuarne all'interno i concetti principali, collegandoli a entità che li descrivono (nella gran parte dei casi pagine di Wikipedia). Allo scopo di confrontare tra di loro le prestazioni dei vari annotatori sviluppati, sono sorti parallelamente strumenti di benchmarking atti a definire una comune terminologia e insieme di problemi che permettesse di mettere in relazione le performance di diversi sistemi. Questi strumenti di benchmarking mettono a disposizione un insieme di problemi e di dataset, per i quali sono conosciute le soluzioni corrette, e permettono di verificare il funzionamento di vari sistemi di annotazione su quei problemi e quei dataset confrontando le soluzioni (annotazioni) ottenute con la soluzione corretta in modo da derivare misure che permettano di esprimere numericamente le prestazioni dell'annotatore in oggetto.

Il lavoro che sarà presentato nei successivi capitoli ha avuto come obiettivo, partendo

dai due concetti principali appena esposti, quello di migliorare quanto più possibile le performance di un sistema di annotazione già esistente, noto con il nome di WAT [2], e nel contempo analizzare in profondità le sue componenti algoritmiche al fine di poter individuare eventuali inefficienze che ne pregiudicano il funzionamento.

Il sistema analizzato si pone già come stato dell'arte nel mondo dell'annotazione [4] e il lavoro svolto ha avuto l'obiettivo precipuo di consolidarne ulteriormente tale posizione. Per ottenere tale scopo si è proceduto in due fasi. Nella prima si è integrata nel framework di benchmarking di riferimento nella comunità degli annotatori semantici, GERBIL [4], una funzionalità di error reporting, che consentisse di effettuare un'analisi dettagliata dei risultati di un problema di annotazione, al fine di individuare punti di forza e debolezze dell'annotatore, correggendo queste ultime, ove possibile. Nella seconda fase si è proceduto, con il supporto dato dall'analisi dei report ottenuti tramite il framework di benchmarking, ad individuare possibili strategie per affrontare i problemi emersi nell'annotazione, concentrando separatamente l'attenzione sulle diverse fasi della pipeline di WAT (esaminate più avanti in dettaglio), e confrontando di volta in volta i risultati ottenuti non solo con le precedenti prestazioni ottenute dal medesimo sistema, ma anche con quelle ottenute dai sistemi concorrenti, in modo da stabilire quale fosse il sistema in grado di fornire le migliori prestazioni globali.

Nel capitolo 1 verranno espressi i concetti principali alla base del problema dell'annotazione testuale, indicando le problematiche che questa tecnica mira a risolvere e quali siano le sue applicazioni pratiche. Nel capitolo 2 si introdurrà il concetto di benchmarking di annotatori e si presenteranno i più diffusi framework per la valutazione delle loro prestazioni, e in particolare il più utilizzato, GERBIL, sistema al quale è stata aggiunta la funzionalità di error reporting esaminata nel dettaglio nel capitolo 3. Nei capitoli 4 e 5 verranno presentati due sistemi di annotazione: TAGME e il suo successore, WAT, oggetto del lavoro di tesi svolto, dei quali si presenteranno le fasi del funzionamento e le prestazioni. Nel capitolo 6 sarà effettuata una breve panoramica sulle performance ottenute dagli annotatori pubblicamente disponibili, proseguendo con la presentazione dell'analisi approfondita svolta sul sistema e delle modifiche apportate a WAT al fine di renderne le prestazioni superiori agli altri sistemi di annotazione. Nel capitolo 7 saranno tratte le conclusioni sul lavoro svolto illustrando possibili scenari per ricerche future.

Capitolo 1

Annotatori Semantici

1.1 Dal sistema classico all'annotazione testuale

L'approccio classico dell'information retrieval ai problemi quali l'indicizzazione, il clustering, la classificazione, ecc. è quello basato sul paradigma bag of words, incentrato sull'utilizzo dei termini presenti all'interno di un documento e sulla loro frequenza. Nel corso degli ultimi anni si è verificata una tendenza allo sviluppo di soluzioni che consentissero di ottenere risultati soddisfacenti anche nel caso di documenti non strutturati o semi strutturati, non particolarmente adatti ad essere trattati con il paradigma classico. Tra i possibili metodi sfruttabili per il conseguimento di tale risultato, grande importanza riveste quello dell'annotazione testuale. Con annotazione testuale si intende quella procedura che, a partire da un testo, individua al suo interno sequenze di termini che vengono annotate collegandole ad entità univoche prelevate da un catalogo. Grande importanza in tale tipo di sistema riveste ovviamente la scelta del catalogo dal quale estrarre le entità con cui annotare i testi, tale scelta nella letteratura ricade in gran parte dei casi su Wikipedia. Tra le motivazioni del successo di Wikipedia come base di conoscenza d'elezione per la soluzione del problema dell'annotazione è sicuramente da annoverare la costante espansione del suo già corposo catalogo di pagine, nonché il suo offrire il miglior rapporto quantità/qualità, garantito dalla struttura rigorosa e dalla grande quantità di informazioni, che la pone al di sopra di altre possibilità caratterizzate dal privilegiare un criterio rispetto all'altro.

1.2 Problematiche del paradigma classico

Oltre alla mancanza di struttura precisa di alcune categorie di documenti, esistono altre tipologie di inconvenienti che impediscono di ottenere risultati di livello superiore facendo affidamento sul solo paradigma bag-of-words. Tra questi particolare rilevanza è posta sui problemi della polisemia e dei sinonimi. Il primo dei due si verifica quando una parola contenuta in un documento può avere più di un significato, sarebbe quindi utile individuare tra i possibili significati quello più coerente col contesto in cui il termine è posizionato. Analogo è il problema dei sinonimi, che vede al contrario la presenza di più termini possibili per un medesimo significato. Entrambe le problematiche evidenziano in modo particolare

le limitazioni di un modello basato esclusivamente sulle parole contenute in un documento e non sulle possibili strutture implicite in esso celate. L'utilizzo dell'annotazione consente non solo di arricchire il testo originale con una lista di entità ad esso pertinenti ma anche di evidenziare, rendendole parte attiva del processo informativo, strutture implicite e relazioni tra parti del testo che rimarrebbero altrimenti inutilizzate.

1.3 Terminologia

L'applicazione di un processo di annotazione semantica ad un documento dà origine alla produzione di una lista di **annotazioni**. Un'annotazione è una coppia formata dalla cosiddetta **menzione** (*mention*), una sequenza di termini appartenenti al documento, rappresentata da una coppia di interi ad indicarne la posizione nel testo, e da un'**entità**, ossia un elemento univoco appartenente al catalogo scelto per l'annotazione, in gran parte dei casi Wikipedia. Prima di associare alle menzioni individuate le opportune entità, un sistema di annotazione deve provvedere ad una fase preliminare di analisi, durante la quale devono essere selezionati, all'interno del testo, i cosiddetti **spot**, porzioni di testo candidate ad essere annotate col senso ritenuto più pertinente. Nell'ambito di un catalogo caratterizzato da una struttura a grafo realizzata per mezzo di collegamenti ipertestuali quale è Wikipedia, prende il nome di **àncora** (*anchor*), la porzione di testo utilizzata per descrivere un collegamento ad una determinata pagina. In sostanza è possibile individuare come sinonimi di una pagina i termini utilizzati per riferirsi ad essa dall'interno di un'altra entità.

1.4 Processo di annotazione

Il procedimento che porta a produrre una lista di annotazioni a partire da un testo ricevuto in input si compone di diverse fasi interconnesse tra loro, tutte concorrenti per determinati aspetti al conseguimento di un risultato soddisfacente.

La prima azione che un generico annotatore deve svolgere una volta ricevuto un documento da annotare è quella di effettuarne la *tokenizzazione*. In questa fase il testo viene suddiviso in porzioni (generalmente parole) costituenti l'unità base dell'analisi successiva. Durante la fase di tokenizzazione è pratica comune quella di avvalersi di strumenti in grado di arricchire i *token* con informazioni utili alla fase successiva. Tra queste informazioni particolare importanza rivestono le etichette caratterizzanti la tipologia di testo contenuto nel token, esse possono essere ad esempio di tipo *part-of-speech*, rappresentando quindi quale parte della frase il token rappresenti (nome proprio, aggettivo, ecc.), sia di tipo più generale, in cui ad un token viene associata una categoria appartenente ad un insieme predefinito, utilizzato per indicare la tipologia di entità identificata analizzando il token (luogo, persona, ecc.).

Effettuata l'operazione preliminare di tokenizzazione il testo deve essere analizzato al fine di individuare gli spot, ossia le porzioni in esso contenute che potrebbero diventare menzioni da utilizzare ai fini del calcolo del risultato finale. In questa fase il testo viene scorso, sfruttando qualora presenti, le informazioni inserite dal tokenizer, alla ricerca di token, o gruppi di token, che facciano riferimento ad un concetto la cui annotazione

potrebbe risultare pertinente. Il prodotto finale di questa fase consiste in una lista di menzioni, ritenute annotabili dallo spotter, ognuna di esse associata ad una lista di entità che potrebbero rappresentarne il significato.

Il passo successivo consiste nell'esaminare, per ogni menzione prodotta dalla fase di spotting, la lista delle possibili entità associabili, in modo da individuare al suo interno l'entità ritenuta più pertinente per la menzione in oggetto. Esistono diverse tecniche per portare avanti quest'analisi, alcune delle quali verranno analizzate nei prossimi capitoli quando si esamineranno nel dettaglio due annotatori semantici in particolare, TAGME e WAT. La fase di disambiguazione termina con una lista di menzioni alle quali è stata associata un'entità ritenuta opportuna, dando così origine alle annotazioni rilevate nel testo.

Tuttavia sarebbe rischioso e controproducente fornire come risultato tale lista senza effettuare un'ultima operazione di ripulitura del risultato. Tale operazione prende il nome di potatura (*pruning*) e vede il sistema impegnato nel rianalizzare i risultati prodotti dagli stage precedenti nel tentativo di identificare annotazioni che sono state prodotte perché il corrispondente spot era stato identificato nei momenti iniziali dell'annotazione, ma che risultano poco pertinenti o di scarsa rilevanza nel contesto globale del documento.

Terminata questa fase l'annotazione può dirsi conclusa e le annotazioni mantenute dal pruner in quanto ritenute pertinenti ne costituiscono il risultato.

1.5 Fasi critiche

Sebbene all'apparenza possa sembrare che sia sufficiente essere in possesso di un efficace algoritmo di disambiguazione per ottenere risultati ottimali, è bene invece sottolineare l'importanza delle restanti fasi e individuare i punti critici di ognuna di esse. È innegabile l'importanza di utilizzare un disambiguatore che sia in grado, sfruttando le informazioni contenute nei token e il contesto generale del testo, di individuare ogni volta l'entità più pertinente nel caso specifico, agendo con robustezza anche in caso di rumore nei dati di origine. Tuttavia non si può assolutamente sottovalutare il ruolo nel complesso degli altri stage.

Per comodità si analizzeranno congiuntamente le fasi di tokenizzazione e quella immediatamente successiva, indicandole nel loro insieme come fase di spotting. Questa fase preliminare risulta nella letteratura particolarmente importante, sebbene spesso tralasciata, in quanto è durante questo stage che vengono gettate le basi del percorso di annotazione, e un output quanto più possibile vicino a quello ottimale è fondamentale per la buona riuscita del processo. Una fase di spotting troppo lasca porterebbe infatti alla presenza, nella lista di quelle da disambiguare, di menzioni di scarsa rilevanza all'interno del documento, che, una volta annotate, in caso di una potatura finale non sufficiente, porterebbero le performance complessive dell'annotatore a peccare di precisione, essendo comprese nel risultato informazioni non influenti. Al caso opposto uno spotting troppo restrittivo rischierebbe di tralasciare concetti importanti che verrebbero quindi persi, portando il risultato complessivo ad essere probabilmente molto preciso ma di scarsa rilevanza in quanto mancante di elementi fondamentali.

Di primaria importanza come è facile intuire, è l'interazione tra le fasi, e in particolare l'equilibrio da garantire tra la fase di spotting e quella di potatura. È infatti chiaro come, nel caso si decida di gestire in maniera lasca la fase di spotting, sia assolutamente necessario un certosino lavoro di potatura che garantisca buone performance in termini di pertinenza del risultato. Nei casi limite tuttavia questo richiede un algoritmo di potatura particolarmente robusto, perché sia in grado di individuare tutti gli elementi spuri prodotti dallo spotter. Al contrario, in caso di spotting più restrittivo, è sufficiente un algoritmo di pruning più semplice, in quanto la supposizione iniziale è che le annotazioni prodotte contengano pochi, se non nessuno, valori che devono essere rimossi.

Come si vedrà nei capitoli successivi è fondamentale, ai fini della realizzazione di un annotatore che sia in grado di competere ad armi pari con i migliori concorrenti sulla piazza, trovare un'armonia fra le tre fasi fondamentali, cercando di sopperire con i pregi di un determinato stage ai difetti degli altri.

1.6 Caratteristiche sfruttate da un annotatore

La scelta di sviluppare una classe di software come quella degli annotatori, arricchendo i testi di informazioni estratte da una base di conoscenza, consente di sfruttare, per raggiungere lo scopo finale, alcune particolari caratteristiche del catalogo scelto, nella gran parte dei casi Wikipedia.

Nella fattispecie, gran parte dei vantaggi di questo approccio sono prodotti dalla struttura a grafo di Wikipedia, che consente, se opportunamente esaminata, di individuare particolari relazioni tra le entità non necessariamente espresse in maniera esplicita.

La più basilare delle informazioni estratte che viene utilizzata nell'annotazione ha a che fare con le cosiddette àncore, porzioni di testo utilizzate all'interno di una pagina per fare riferimento ad un'altra, che vengono quindi assunte come possibili descrizioni dell'entità stessa e usate come base di partenza per la ricerca nel testo di menzioni da annotare. Per quanto riguarda le informazioni ricavabili dalle àncore testuali, sono preminenti i concetti di *commonness* di una determinata pagina in relazione ad un'àncora e di *link probability* di un'àncora.

La commonness di una pagina p rispetto ad un'àncora a è espressa come il rapporto tra il numero di volte in cui a è utilizzata per linkare p e il numero di volte che a è usata come àncore nelle varie pagine di Wikipedia. La link probability di un'àncora a è invece il rapporto tra la frequenza di a come àncora e la frequenza di a nel testo.

In ultimo di frequente i sistemi di annotazione si avvalgono, per effettuare analisi più efficienti, del concetto di *contesto* intorno ad una determinata menzione, nel quale il testo circostante una menzione viene utilizzato per contribuire all'individuazione della più pertinente entità con la quale annotarla.

1.7 Applicazioni pratiche

Vale la pena di effettuare un rapido excursus sulle possibili applicazioni di una tecnologia in rapida ascesa come quella degli annotatori tesuali. Un primo campo di utilizzo è quello della categorizzazione di testi, che consiste nell'esaminare documenti formulati in

linguaggio naturale etichettandoli con una o più categorie tematiche, appartenenti ad un insieme predefinito di possibilità, che ne rappresentino il significato. Le soluzioni tipiche per questo genere di task sono studiate per sfruttare le peculiarità di testi lunghi (e quindi ricchi di termini), e risultano pertanto poco efficaci in caso di testi corti e di qualità non verificabile, punto di forza invece degli strumenti di annotazione.

Analogo è anche l'utilizzo di annotatori testuali allo scopo di effettuare una classificazione per argomento di una lista di notizie. Tale applicazione è resa possibile dagli annotatori effettuando un processo in due fasi: una di training, dove una porzione di dati viene annotata e per ogni classe desiderata viene creato un sottoinsieme di argomenti ad essa correlato. In fase di classificazione vera e propria sarà sufficiente annotare il testo desiderato e calcolare la correlazione tra le annotazioni rilevate e gli elementi appartenenti alle liste di argomenti assegnate ad ogni classe, individuando così, in base alla correlazione tra il testo e gli argomenti, la classe giusta nella quale inserire la notizia.

Altra famiglia di applicazioni presente in letteratura è la profilazione di utenti di social network in base all'annotazione dei testi pubblicati, i cui concetti annotati costituiscono un'immagine definita dell'utente e dei suoi interessi, che permette ad esempio di predire, dato un tweet, quale utente scelto tra una lista sarà teoricamente "più interessato" ai contenuti pubblicati; o di individuare un insieme di notizie che risultino interessanti per un utente precedentemente profilato.

Non vanno infine trascurate due applicazioni strettamente legate all'Information Retrieval classico applicato ai motori di ricerca e al recupero di informazioni pertinenti alla richiesta effettuata da un utente. Il primo dei due ha a che fare con l'arricchimento di una query effettuata su un motore di ricerca con concetti strettamente correlati ma non esplicitamente espressi tra le sue keyword e quindi non individuabili dal paradigma bag-of-words. È il caso di una query che fa uso di un'espressione comune per indicare un'entità ben precisa ma che non condividendone i termini non verrebbe collegata alla query. Si prenda ad esempio la query *"Migliorare le prestazioni del browser di Microsoft"*, che, grazie all'annotazione permetterebbe di arrivare ai risultati aventi a che fare con Internet Explorer senza che questo sia mai esplicitamente nominato nella richiesta effettuata dall'utente. Correlato a quello appena descritto, troviamo un secondo campo di applicazione: l'anticipazione del desiderio di informazione di un utente. Questo problema consiste nell'individuare quali possano essere gli argomenti interessanti per l'utente in seguito alla visita della pagina corrente. Una soluzione a questo problema fornita dagli annotatori consiste nell'estrarre mediante annotazione le entità della pagina che l'utente sta visitando e costruire in base ad esse una lista di query correlate che possano essere proposte all'utente per venire incontro alle sue necessità riuscendo talvolta a prevederle.

Capitolo 2

Benchmarking

Il recente sviluppo di numerosi annotatori ha reso più stringente la necessità di una categoria di sistemi che offrissero una comparabilità tra i risultati di tali strumenti, resa più difficile da raggiungere proprio dal numero di differenti sistemi presenti e dalla loro eterogeneità. Il problema del benchmarking è da vedersi come totalmente slegato dal problema dell'annotazione in sé, ed è esclusivamente finalizzato a fornire ai ricercatori e agli sviluppatori software uno strumento che permetta loro di effettuare in maniera comoda ed esaustiva test approfonditi sui sistemi disponibili senza dover entrare nel merito del loro calcolo.

2.1 Analisi delle prestazioni di un annotatore

Per effettuare un'efficiente analisi comparativa delle prestazioni di vari annotatori, è necessario definire un insieme comune di problemi da risolvere e di misure di valutazione dei risultati che permettano quindi un raffronto dettagliato di sistemi eterogenei per terminologia e funzionamento. Per eseguire una corretta valutazione dei risultati, tali esperimenti vengono eseguiti su insiemi di dati, corredati di golden standard, il quale viene raffrontato con l'output prodotto dall'annotatore, calcolando le relative metriche.

2.1.1 Problemi di annotazione

Nella letteratura, seppur con sfumature leggermente differenti, si è giunti alla definizione di un insieme preciso di problemi di annotazione e di metriche, utilizzato per la valutazione di sistemi di questo genere.

- **Disambiguate to Wikipedia (D2W)**: ricevuto in input un testo ed un set di mention, l'annotatore assegna ad ogni mention ricevuta la sua corrispondente entità (eventualmente null);
- **Annotate to Wikipedia (A2W)**: dato un testo, il compito dell'annotatore è di identificare le mention rilevanti ed assegnare loro un'entità pertinente;

- **Scored-annotate to Wikipedia (Sa2W)**: questo problema è strutturato in maniera identica al precedente, con la differenza che ad ogni annotazione è assegnato un punteggio che indica la probabilità che l'annotazione sia corretta;
- **Concepts to Wikipedia (C2W)**: a partire da un testo ricevuto in input, l'annotatore calcola un insieme di tag, ossia le entità rilevanti in esso menzionate;
- **Scored-concepts to Wikipedia (Sc2W)**: questo genere di esperimento viene svolto con gli stessi criteri del precedente, ma ad ogni tag viene assegnato un punteggio che indica la probabilità di correttezza della corrispondente annotazione;
- **Ranked-concepts to Wikipedia (Rc2W)**: l'input iniziale di questo problema è un testo, il compito dell'annotatore è di identificare le entità al suo interno menzionate e ordinarle in termini della loro pertinenza con gli argomenti trattati nel testo stesso.

2.1.2 Match

Una volta terminata la fase di annotazione, è necessario eseguire un raffronto dettagliato tra il risultato fornito dall'annotatore e il golden standard fornito dal dataset utilizzato. Si rende quindi necessario l'utilizzo di una misura che indichi se due annotazioni che si stanno confrontando siano o meno uguali. Tale criterio di uguaglianza prende generalmente il nome di match. Esso è calcolato in relazione a due annotazioni.

Match per il problema C2W Partendo dalla tipologia di problema è facile intuire quale possa essere un criterio di match tra due output di questo genere, è infatti sufficiente verificare l'uguaglianza delle entità linkate dalle annotazioni, si dice quindi che esiste uno **Strong entity match** tra due entità e_1 ed e_2 se e solo se $e_1 == e_2$;

Match per il problema D2W Considerato che l'output per questo problema è costituito da un insieme di annotazioni, il concetto di match deve necessariamente fare riferimento a coppie $\langle$mention, entity$\rangle$. Dovendo verificare il match $M(a_1, a_2)$, esso sarà vero se e solo se le mention e le entity delle due annotazioni sono uguali, e si potrà dire che le due annotazioni sono legate da uno **Strong entity match**;

Match per il problema A2W Anche in questo caso l'output dell'annotatore sarà un insieme di coppie $\langle$mention, entity$\rangle$, ma, a differenza del caso precedente, le mention dovranno essere calcolate e non ricevute in input, è quindi necessario estendere il concetto di match tenendo conto anche della validità delle menzioni. Vengono quindi utilizzati due tipi di match:

- **Strong annotation match**, il quale è verificato se e solo se le due mention sono perfettamente allineate (iniziano e finiscono nella medesima posizione) e le due entity sono uguali;
- **Weak annotation match**, introdotto per offrire un concetto di uguaglianza che assecondasse la natura fuzzy del problema, che si distingue dal precedente

per il fatto che è verificato se e solo se le due mention sono sovrapposte l'una all'altra e non più perfettamente allineate.

2.1.3 Misure di valutazione

Stabilito il criterio di uguaglianza tra due annotazioni, è possibile generalizzare le metriche classiche di valutazione utilizzate nell'informazion retrieval (*true/false positive, true/false negative, f1, precision e recall*) ridefinendole in funzione della relazione binaria di match appena definita. Dato in input un testo t, si consideri g come l'insieme di elementi corretti (siano essi mention, entità o annotazioni). Le misure base dalle quali vengono derivate le altre sono quelle di **true positive/negative** (tp/tn) e **false positive/negative** (fp/fn). Con true positive si intendono gli elementi corretti che il sistema è stato in grado di individuare come tali, mentre con true negative si indica l'analogo concetto riguardante però i valori che non fanno parte dell'insieme dei corretti. True positive e true negative sono pertanto misure che offrono una misura della quantità di scelte corrette effettuate dal sistema. Per quanto riguarda invece la misura degli errori commessi dall'annotatore, si fa uso dei concetti di false positive e false negative, dove il primo rappresenta i valori non appartenenti alla soluzione ideale ma inclusi nella soluzione dall'annotatore, mentre il secondo indica quegli elementi che avrebbero dovuto far parte del risultato finale ma sono sfuggiti all'analisi dell'annotatore. Abitualmente quando si utilizzano queste misure base, si fa riferimento al numero di elementi di ognuna delle quattro categorie presentate. Dato un testo t, sia indicato con g l'insieme degli elementi corretti (menzioni, entità o annotazioni a seconda del problema) presenti nel golden standard per il testo t, e sia s la soluzione calcolata dall'annotatore. È possibile calcolare le quattro misure secondo le seguenti definizioni:

$$
\begin{aligned}
tp(s, g, M) &= \{x \in s \mid \exists x' \in g : M(x', x)\} \\
fp(s, g, M) &= \{x \in s \mid \nexists x' \in g : M(x', x)\} \\
tn(s, g, M) &= \{x \notin s \mid \nexists x' \in g : M(x', x)\} \\
fn(s, g, M) &= \{x \in g \mid \nexists x' \in s : M(x', x)\}
\end{aligned}
$$

Introdotte queste unità base di valutazione, è possibile derivare nel metodo standard le metriche **precision, recall ed F1**. Come avviene tipicamente, con precision si indica la frazione di risultati pertinenti tra tutti i risultati ottenuti, mentre con recall si indica la frazione di risultati contenuti nel golden standard ottenuti dall'annotarore. Dal momento che tali misure applicate singolarmente rischiano di essere facilmente fuorvianti, sia sufficiente pensare che è sufficiente restituire un insieme vuoto per ottenere la massima precision possibile e restituire tutto ciò che il sistema ha trovato per massimizzare il recall, si fa generalmente utilizzo di una terza misura, chiamata F1. Essa rappresenta la media armonica tra le due misure, indicatore ottimale delle prestazioni generali di un sistema, poiché tiene conto sia della quantità di informazioni utili trovata che della qualità del risultato. È chiaro come casi anche meno estremi di quelli presentati possano influenzare il processo di valutazione. Date le definizioni di true positive/negative e false positive/negative, è possibile derivare le misure di precision, recall ed F1 nel modo

seguente:

$$P(s,g,M) = \frac{|tp(s,g,M)|}{|tp(s,g,M)| + |fp(s,g,M)|}$$
$$R(s,g,M) = \frac{|tp(s,g,M)|}{|tp(s,g,M)| + |fn(s,g,M)|}$$
$$F_1(s,g,M) = \frac{2 \cdot P(s,g,M) \cdot R(s,g,M)}{P(s,g,M) + R(s,g,M)}$$

Nella letteratura riguardante il benchmarking di annotatori tali metriche sono, quando utilizzate per valutare le prestazioni su un intero set di documenti, sdoppiate in **micro e macro misure**, le prime facenti riferimento a tutte le annotazioni calcolate in un determinato esperimento, le ultime costituite dalla media delle corrispondenti misure su tutti i documenti considerati. Indicata come s_t e g_t rispettivamente la soluzione trovata da un annotatore e quella riportata nel golden standard per un documento $t \in D$, le macro e micro misure sono ottenute come segue:

$$P_{mic}(S,G,M) = \frac{\Sigma_{t \in D}|tp(s_t,g_t,M)|}{\Sigma_{t \in D}(|tp(s_t,g_t,M)| + |fp(s_t,g_t,M)|)}$$
$$R_{mic}(S,G,M) = \frac{\Sigma_{t \in D}|tp(s_t,g_t,M)|}{\Sigma_{t \in D}(|tp(s_t,g_t,M)| + |fn(s_t,g_t,M)|)}$$
$$F1_{mic}(S,G,M) = \frac{2 \cdot P_{mic}(S,G,M) \cdot R_{mic}(S,G,M)}{P_{mic}(S,G,M) + R_{mic}(S,G,M)}$$
$$P_{mac}(S,G,M) = \frac{\Sigma_{t \in D} P(s_t,g_t,M)}{|D|}$$
$$R_{mac}(S,G,M) = \frac{\Sigma_{t \in D} R(s_t,g_t,M)}{|D|}$$
$$F1_{mac}(S,G,M) = \frac{2 \cdot P_{mac}(S,G,M) \cdot R_{mac}(S,G,M)}{P_{mac}(S,G,M) + R_{mac}(S,G,M)}$$

2.2 Framework per il benchmarking di annotatori

2.2.1 Caratteristiche generali

Al netto delle differenti scelte implementative attuabili nella realizzazione di un sistema di benchmarking di annotatori, esistono caratteristiche generali condivise da tale gruppo di sistemi. Criterio fondamentale è quello di trattare i sistemi di annotazione come scatole nere, prescindendo quindi da eventuali particolarità e riducendo a forma comune i risultati prodotti per un dato problema di annotazione, limitandosi pertanto alla mera interrogazione delle interfacce disponibili esclusivamente per scopi di valutazione delle prestazioni. Un sistema di benchmarking avrà quindi a disposizione un set di annotatori, eventualmente estendibile, un set di problemi e un gruppo di dataset adatti alla valutazione dei risultati di ogni dato problema. Il principio generale di funzionamento è quello di risolvere con i vari annotatori uno dei problemi precedentemente indicati utilizzando come input

i documenti presenti nei dataset, per poi valutare i risultati e calcolare le metriche che facciano da base al confronto tra gli annotatori. Lo scopo finale è quello di ottenere un sistema che riesca ad uniformare i criteri di valutazione delle prestazioni e offra una panoramica esaustiva delle potenzialità di un dato sistema, senza costringere l'utente finale a concentrarsi sulle particolarità, siano esse implementative o terminologiche, dei singoli sistemi.

2.2.2 BAT-Framework

Un primo approccio realizzativo nei confronti del problema del benchmarking prende il nome di BAT-Framework [3], realizzato nel dipartimento di Informatica dell'Università di Pisa e tuttora usato come nucleo centrale del più affermato sistema di raffronto tra annotatori, GERBIL [4]. Lo scopo del BAT-Framework è quello di fornire una API di facile utilizzo che permetta l'esecuzione di diversi tipi di test che consentano di misurare la capacità di un sistema di annotare le istanze di un dataset. Il framework consente di eseguire un confronto tra i più efficienti sistemi di annotazione pubblicati: AIDA, Illinois Wikifier, TAGME, Wikipedia-miner e DBpedia Spotlight. Per eseguire il raffronto, all'interno del framework è stata definita un'implementazione delle misure di valutazione precedentemente esposte e una gerarchia di problemi, costruita sulla base dei problemi classici di annotazione. I problemi introdotti, essendo strettamente correlati tra di loro, si prestano ad un'organizzazione gerarchica originata da una serie di riduzioni che permettono di ottenere il risultato di un problema a partire dal risultato di un altro. La gerarchia risultante è facilmente rappresentabile per mezzo di un directed acyclic graph (DAG), riportato in figura 2.1, la cui struttura permette di verificare con immediatezza le relazioni che intercorrono tra i vari problemi proposti.

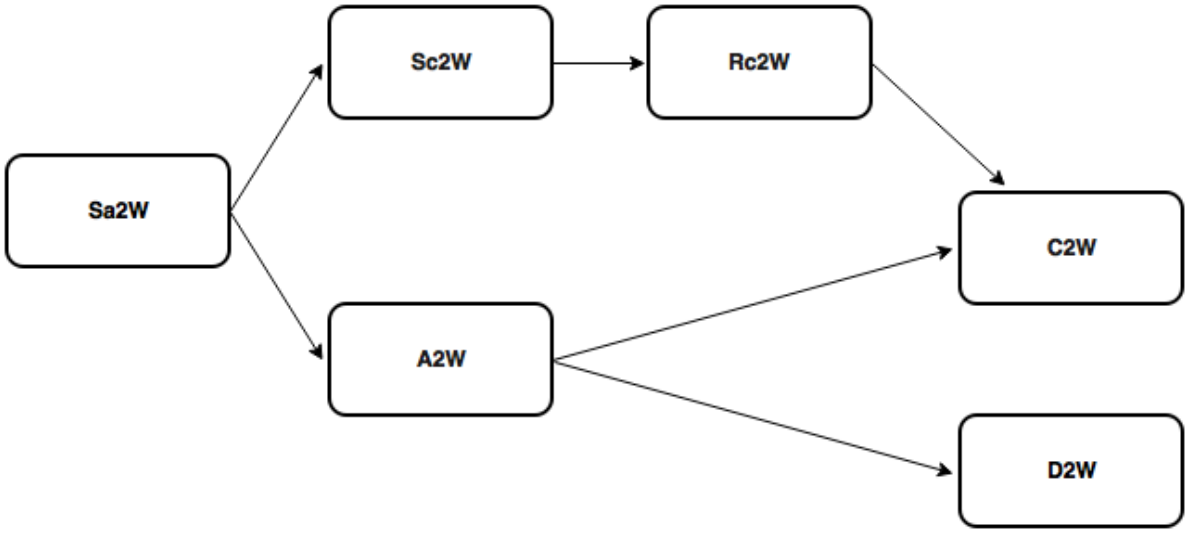


Figura 2.1: Dag dei problemi, in ordine dal più complesso al più semplice

L'utilizzo del DAG permette di intuire il modo in cui siano stati implementati i processi di risoluzione dei problemi anche dataset che non offrissero esplicitamente supporto

a determinati problemi. Infatti è possibile sfruttare il grafo per derivare dalla soluzione (conosciuta) di un problema più generale, la soluzione per problemi calcolabili come riduzione di tale problema. Lo schema di funzionamento è concettualmente semplice: l'insieme dei documenti di un dataset viene trasmesso in input ai sistemi di annotazione che partecipano all'esperimento, il loro output viene in seguito confrontato con il golden standard offerto dal dataset stesso, calcolando quindi e misure di valutazione che permetono di stabilire quando i risultati dell'annotatore si avvicinino alla corretta soluzione del problema.

2.2.3 GERBIL

Utilizzando come punto di partenza il BAT-Framework, è stato sviluppato all'università di Lipsia, con la collaborazione di diverse università europee, tra le quali quella di Pisa, GERBIL (General Entity Annotator Benchmarking Framework), con l'obiettivo di arricchire le funzionalità offerte dal framework di partenza, aggiungendo caratteristiche strutturali e di user-experience che lo hanno portato ad essere il punto di riferimento per il benchmarking dei sistemi di annotazione testuale.

2.2.3.1 Caratteristiche

Rispetto al punto di partenza, costituito dal BAT-Framework, GERBIL offre diverse funzionalità aggiuntive. In primo luogo il sistema è progettato perché l'interazione con l'utente finale avvenga tramite interfaccia Web. Non presente nel BAT-Framework originale, essa permette all'utente di creare gli esperimenti, raggiungibili anche in un secondo momento mediante URL persistenti, attraverso un'interfaccia grafica che offre la possibilità di scelta tra tutti i parametri con i quali un esperimento può essere eseguito. Proprio il già espresso concetto di URL persistente è stato reso possibile dall'introduzione del sistema di un archivio degli esperimenti, rappresentato da una base di dati che custodisce i resoconti degli esperimenti svolti, permettendone la consultazione anche in seguito al termine della procedura sperimentale.

Oltre alle sopraccitate innovazioni relative all'utente è importante notare come la progettazione di GERBIL sia stata svolta con in mente l'introduzione del cosidetto Knowledge-base agnostic benchmarking: la base di conoscenza dalla quale estrarre le entità non è necessariamente Wikipedia, come accade nel BAT-Framework, ma una qualunque base di conoscenza usata come catalogo.

2.2.3.2 Annotatori

In GERBIL sono presenti out of the box, 9 sistemi di annotazione, contro i 5 del BAT-Framework. Tali sistemi sono, nel dettaglio:

- **Wikipedia Miner**: un'implementazione dell'algoritmo di Wikification , uno dei primi approcci relativi alla risoluzione del problema dell'annotazione. Il sistema è basato su una componente di machine learning, che usa come training set link e contenuti prelevati da pagine Wikipedia. All'interno del sistema è presente un classificatore, che seleziona le annotazioni rilevanti e scarta quelle ritenute non pertinenti,

allenato sulla base di tre nozioni: (i) la probabilità che una menzione si riferisca ad una specifica entità, (ii) la correlazione tra il contesto originario dell'entità e (iii) la qualità del contesto, calcolata sulla base del numero di termini presenti, la misura della relazione tra essi e la loro frequenza in Wikipedia;

- **DBpedia Spotlight**: uno dei primi approcci semantici al problema, cerca all'interno del testo in input sottostringhe da confrontare con àncore, titoli e redirect presenti in Wikipedia. Associa in seguito un set di entità candidate ad ognuna delle menzioni individuate. Una volta costituito un insieme di coppie ⟨mention, set di entity⟩, per ognuna di essere converte entrambe le componenti in un vettore (usando il classico approccio bag of words) e sceglie come entità più pertinente quella che ha una cosine similarity maggiore con la mention;

- **TAGME**: sviluppato dall'Università di Pisa nel 2012, dopo aver costruito un insieme di coppie ⟨mention, set di entity⟩ come nell'annotatore precedente, disambigua le entità sfruttando la struttura del grafo di Wikipedia, facendo affidamento su una relazione binaria di relatedness tra pagine, e avvalendosi di un voting scheme, in cui tutte le possibili associazioni menzione-entità ricevono un punteggio e a loro volta esprimono un punteggio per le altre associazioni;

- **NERD-ML**: questo approccio fa affidamento su un classificatore che individua la tipologia di un'entità basandosi su un insieme di caratteristiche di tipo linguistico, e su procedure di NER extraction;

- **KEA NER/NED**: in questo sistema il processo di annotazione inizia con l'individuazione di gruppi di parole consecutive (n-grammi) e il conseguente lookup di tutte le potenziali entità DBpedia per ognuno degli n-grammi. Le entità vengono in seguito disambiguate mediante un sistema a punteggi;

- **WAT**: il successore di TAGME, che include una riprogettazione di tutte le componenti dell'annotatore, introducendo una nuova famiglia di disambiguazione basata sui grafi, oltre al voting scheme eredidato dal suo predecessore;

- **Babelfy**: Il punto centrale di questo sistema è l'uso di random walks e di un densest subgraph algorithm, utilizzati per risolvere il problema della disambiguazione del significato delle parole e dell'individuazione delle corrette entità corripondenti in un contesto poliglotta;

- **Dexter**: è un'implementazione open-source di un framework per la disambiguazione delle entità, realizzata allo scopo di semplificare l'implementazione di un approccio di individuazione delle entità permettendo di sostituire singole parti di tale processo.

2.2.3.3 Dataset

Oltre ai dataset originariamente utilizzabili nel BAT-Framework, GERBIL ne introduce complessivamente sei. I dataset originariamente presenti erano:

- **AIDA/CoNLL**: costituito da documenti prelevati dal Reuters Corpus V1. Al suo interno sono annotate gran parte delle menzioni relative a named entities, ma non i nomi comuni. Le entità sono annotate per ogni occorrenza della mention;
- **AQUAINT**: contiene un sottoinsieme del corpus AQUAINT originale, costituito da testi relativi a notizie di agenzia in inglese. A differenza del precedente dataset non tutte le occorrenze delle mentioni sono annotate: solo la prima menzione di ogni entità lo è e solo le più importanti entità sono considerate;
- **IITB**: contiene oltre 100 documenti prelevati da pagine Web riguardanti sport, intrattenimento, scienza, tecnologia e salute, annotati da esseri umani. In questo dataset pressoché tutte le mention, considerate quelle relative a concetti non troppo importanti, sono annotate;
- **Meij**: contiene tweet annotati con tutte le entità presenti. Trattandosi di tweet i documenti sono molto brevi e dalla dubbia struttura;
- **MSNB**: un dataset formato da notizie di agenzia del canale MSNBC, nel quale sono annotate solo le entità importanti e le relative mention.

Oltre a quelli appena presentati, GERBIL introuce la possibilità di utilizzare il dataset ACE2004, implementato dagli stessi autori. Il sistema offre inoltre la possibilità di sfruttare dataset per i quali sono stati sviluppati appositi wrapper, tra i quali:

- Microposts2014, utilizzato per la valutazione del sistema NERD-ML
- N3-RSS-500
- N3-Reuters-128
- DBpediaSpotlight
- KORE 50

2.2.3.4 Problemi

Seppur utilizzando una nomenclatura leggermente differente, in virtù del già citato Knowledge-base agnostic benchmarking, i problemi che GERBIL permette di risolvere sono i medesimi esposti precedentemente, che assumono all'interno del framework il nome di:

- D2KB (Disambiguate to Knowledge Base)
- A2KB (Annotate to Knowledge Base)
- Sa2KB (Scored-annotate to Knowledge Base)
- C2KB (Concepts to Knowledge Base)
- Sc2KB (Scored-concepts to Knowledge Base)
- Rc2KB (Ranked-concepts to Knowledge Base)

Lo scopo di tale modifica rispetto alla struttura (e al nome) originario dei problemi, è quello di consentire al framework di lavorare correttamente con dataset e annotatori facenti riferimento ad ogni Knowledge Base (e.g. DBpedia, BabelNet, ecc.), a patto che gli identificatori delle entità siano URI.

Corpus	Topic	# Documenti	# Annotazioni	Problema
ACE2004	news	57	250	Sa2KB
AIDA/CoNLL	news	1393	27815	Sa2KB
Aquaint	news	50	727	Sa2KB
IITB	misto	103	11242	Sa2KB
KORE 50	misto	50	143	Sa2KB
Meij	tweet	502	812	Rc2KB
Microposts2014	tweet	3505	2278	Sa2KB
MSNBC	news	20	650	Sa2KB
N3 Reuters-128	news	128	621	Sa2KB
N3 RSS-500	feed RSS	500	495	Sa2KB
Spotlight Corpus	news	58	330	Sa2KB

Tabella 2.1: Caratteristiche dei dataset ed esperimenti supportati

2.2.3.5 Espandibilità

GERBIL è distribuito come progetto open-source per permettere a chiunque ne abbia la necessità di estenderlo con nuove tipologie di esperimenti o di renderlo adatto ai propri desideri di utilizzo. Allo scopo di garantire la facile espandibilità del sistema, l'architettura di GERBIL è stata realizzata in modo che sia facile integrare le funzionalità offerte con l'aggiunta al framework di nuovi dataset, annotatori e misure di valutazione dell'annotazione. In particolar modo l'estensibilità dei dataset in GERBIL è agevolata dal fatto che all'utente finale è concesso di effettuare l'upload di propri dataset che rispettino il protocollo NIF (caratteristica sfruttata da alcuni dei dataset già presenti all'interno del framework), tale possibilità è valida anche per gli annotatori, la cui lista può così essere estesa senza dover necessariamente apportare modifiche al codice sorgente.

2.2.3.6 Esperimenti

Come già anticipato, l'interazione dell'utente finale con GERBIL avviene per mezzo di una web application, che permette di superare le limitazioni tecniche presenti nel BAT-Framework, che richiedeva per essere sfruttato appieno, una conoscenza, seppur non necessariamente avanzata, della programmazione in Java. La terminologia insita in GERBIL prevede una divisione tra il concetto di esperimento e il concetto di task, con quest'ultimo sottoinsieme del primo. Con task è infatti indicata una tupla ⟨problema, match, annotatore, dataset⟩, mentre un esperimento è costituito da un'aggregazione di task appartenenti alla stessa configurazione fornita dall'utente. La fase principale della realizzazione di un

esperimento è infatti proprio la sua configurazione, che GERBIL permette di esprimere avvalendosi di semplice interfaccia che consente l'immissione dei parametri necessari all'esecuzione. I primi due parametri, condivisi tra tutti i task creati a partire dall'esperimento, sono il tipo di problema da risolvere e il tipo di matching da utilizzare per la valutazione dei risultati (qualora il problema selezionato sia valutabile con più di un tipo di matching). Vengono in seguito richieste una lista di uno o più annotatori, a patto che siano compatibili con il tipo di problema desiderato e una lista di uno o più dataset. Una volta configurato l'esperimento è il sistema ad occuparsi di tutto: vengono create le tuple relative ai task, mediante l'espansione di ogni possibile combinazione ⟨problema, match, annotatore, dataset⟩ realizzabile a partire dai dati dell'esperimento. Inizia poi la fase di annotazione vera e propria: i documenti dei dataset selezionati vengono trasmessi alle interfacce degli annotatori coinvolti, i risultati vengono raccolti da GERBIL che li valuta confrontandoli con il golden standard e calcola di conseguenza le misure di valutazione, sfruttando le definizioni precedentemente indicate.

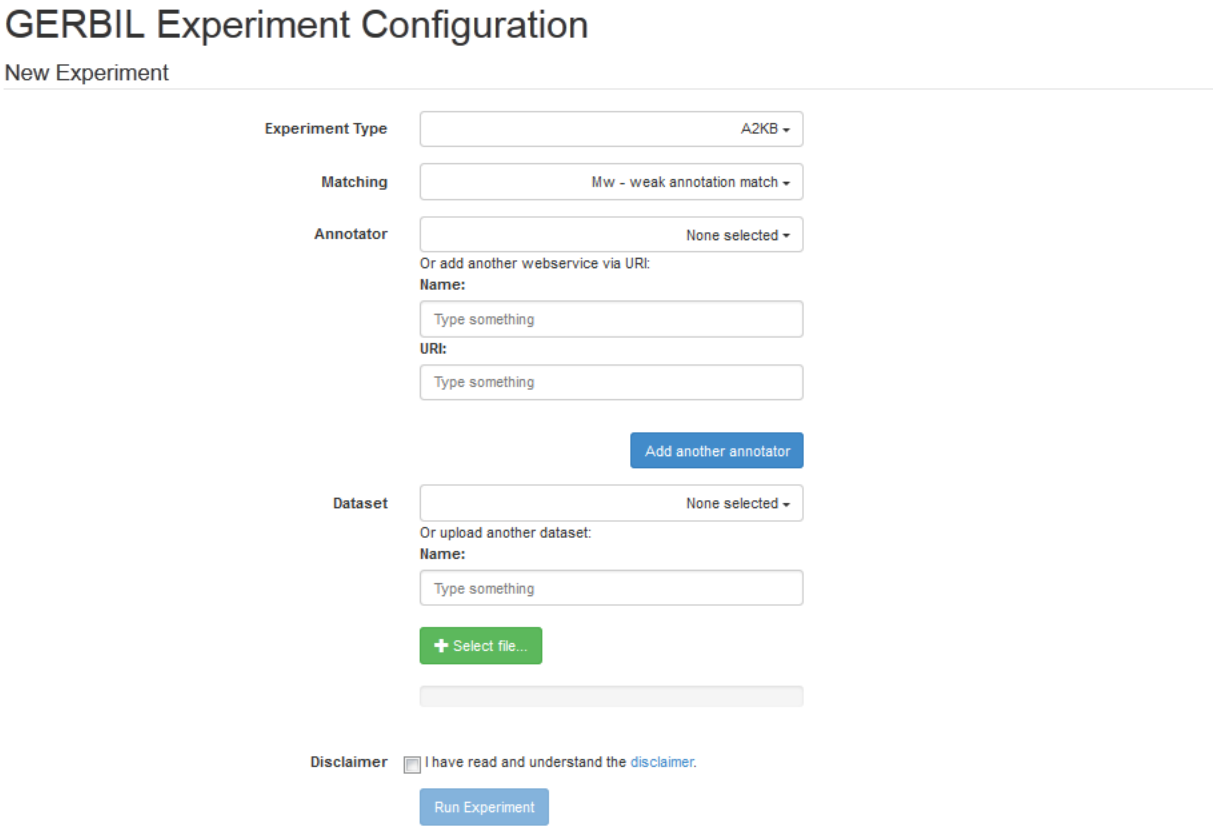


Figura 2.2: Configurazione di un esperimento in GERBIL

2.2.3.7 Risultati

Tra i punti di forza del framework si trova senza dubbio la caratteristica di persistenza degli esperimenti, che permette di ritrovare, mediante URL, i risultati di un esperimento

GERBIL Experiment

Type: A2KB
Matching: Mw - weak annotation match

Annotator	Dataset	Micro F1	Micro Precision	Micro Recall	Macro F1	Macro Precision	Macro Recall	Error Count	Timestamp	GERBIL version
NERD-ML	ACE2004	0,0822	0,044	0,6166	0,088	0,0531	0,7657	0	2015-05-08 18:13:01	1.1.3
TagMe 2	ACE2004	0,0505	0,0261	0,8142	0,0614	0,0338	0,8833	0	2015-05-08 18:07:49	1.1.3
WAT	ACE2004	0,1742	0,0989	0,7323	0,1695	0,1161	0,8161	0	2015-05-08 18:09:12	1.1.3

Figura 2.3: Risultati di un esperimento in GERBIL

precedentemente svolto, nonché di accelerare i tempi di esecuzione, sfruttando, se l'annotatore è configurato per farlo, eventuali task già presenti nella base di dati, in modo di evitare la ripetizione di procedure di annotazione già svolte. L'output di GERBIL è rappresentabile in due modi: una tabella contenente i risultati oppure un resoconto in formato JSON-LD RDF. La più immediata delle opzioni è la prima, integrata nell'interfaccia web, che offre, per ogni task presente nell'esperimento, una rapida panoramica delle misure di valutazione corredata di annotatore, dataset, match utilizzato e timestamp dell'esecuzione.

Un'ulteriore funzionalità offerta da GERBIL è quella che permette il raffronto simultaneo di tutti gli annotatori per i quali sono presenti task nella base di dati. Per ogni tipo di problema (e ogni match compatibile) recentemente risolto, è infatti possibile visualizzare uno spider chart che mette in evidenza le differenti prestazioni di ogni annotatore sui vari dataset presenti, nonché una tabella riepilogativa delle metriche ottenute su ogni dataset dagli annotatori. È inoltre possibile visualizzare uno spider chart e una tabella che mostrano le correlazioni di pearson tra gli annotatori e le caratteristiche dei dataset (e.g. numero di entità per documento, lunghezza media documenti, numero di entità per una data classe, ecc.).

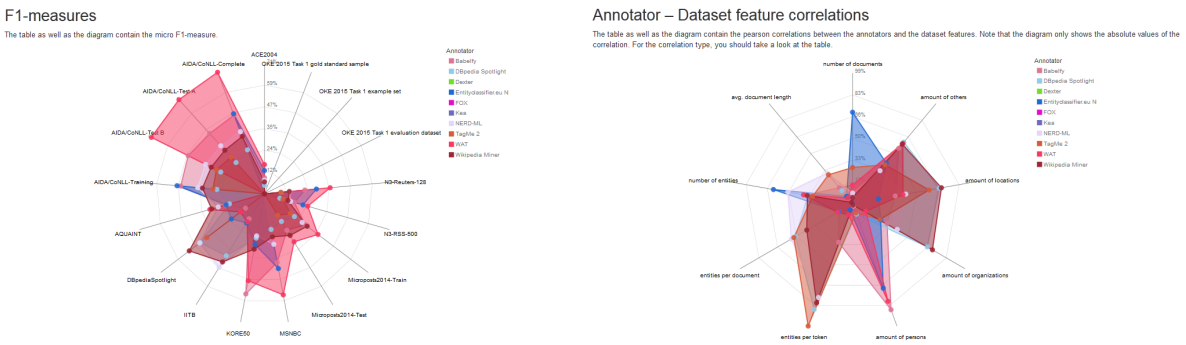


Figura 2.4: Spider chart per F1 e dataset features

Capitolo 3

Il nuovo sistema di Error Reporting

Pur essendo GERBIL lo strumento d'elezione per la procedura di valutazione delle prestazioni di un annotatore testuale, le sue funzionalità non sono pienamente sufficienti qualora si voglia sfruttare il framework per un'operazione lievemente diversa: il tuning di un annotatore allo scopo di migliorarne la performance. La struttura di GERBIL infatti non offre alcun mezzo per verificare in dettaglio gli errori commessi e che hanno dato origini alle metriche calcolate: gli unici dati relativi ad un task che vengono memorizzati nella base di dati sono quelli base che delineano le prestazioni degli annotatori ma appiattendole in sei misure di valutazione (macro-/micro- F1, precision e recall), e che non consentono di approfondire le molteplici sfaccettature di una procedura che si vuole migliorare. Essendo l'obiettivo di lavoro primario di questo progetto di tesi quello di migliorare le prestazioni di un annotatore in particolare (i.e. WAT), analizzandone nel dettaglio le performance, si è deciso, come fase preliminare, di integrare in GERBIL (e nel BAT-Framework) una funzionalità di error reporting che consentisse di avere una rapida visione d'insieme tutti i dati relativi ad un determinato task. Tale funzionalità è stata implementata soltanto per esperimenti di tipo Sa2KB e A2KB, in quanto su di essi si è concentrata l'attenzione nella messa a punto di WAT.

3.1 Terminologia

Per introdurre la questione dell'analisi dettagliata di un task di annotazione è necessario introdurre alcuni concetti che non sono stati trattati nella sezione sul benchmarking in quanto non utilizzati esplicitamente prima d'ora.

In primo luogo è bene ricordare che nella terminologia di GERBIL, con task si indica una tupla ⟨problema, match, annotatore, dataset⟩, che rappresenta un determinato *problema* di annotazione con una specifica valutazione di *match* per le metriche, risolto da un singolo *annotatore* e su un singolo *dataset* (e.g ⟨A2KB, Strong Annotation Match, WAT, KORE50⟩). Un insieme di task eseguiti insieme da GERBIL prende invece il nome di esperimento.

È bene precisare che, sebbene molti dei concetti espressi d'ora in avanti siano compatibili sia con il concetto di weak match che con quello di strong match, negli esperimenti presentati si farà riferimento solo a quest'ultimo e, salvo casi che verranno esplicitamente specificati, nel parlare di test svolti con match si intenderà strong match.

Dato l'insieme delle annotazioni calcolato dall'annotatore (talvolta indicato anche come **actual**) e quelle presenti nel golden standard, esse possono essere distinte in 4 tipi fondamentali:

1. **Correct**: un'annotazione (prodotta dall'annotatore) è considerata corretta se esiste un match tra essa ed una delle annotazioni presenti nel gold standard;

2. **Error**: un'annotazione (prodotta dall'annotatore) è considerata sbagliata se la sua menzione è perfettamente allineata a quella di un'annotazione del gold standard, ma l'entità individuata è differente;

3. **Missing**: un'annotazione (presente nel golden standard) è considerata missing se tra le annotazioni prodotte dall'annotatore non ne esiste una che abbia con essa una corrispondenza come correct o error;

4. **Excess**: un'annotazione (prodotta dall'annotatore) è considerata excess se tra le annotazioni del golden standard non ne esiste una che abbia con essa una corrispondenza come correct o error.

Oltre ai concetti appena definiti di correct, error, missing ed excess, sono particolarmente importanti ai fini di un'analisi dell'errore approfondita, i concetti di False Positive e False Negative, applicati al problema dell'annotazione testuale. Con **False Positive** si intende una porzione di testo che non sarebbe dovuta essere annotata e che invece lo è stata. Al contrario con **False Negative** si intende un'annotazione che il sistema avrebbe dovuto trovare perché presente nel golden standard ma che non è stata individuata. È utile sottolineare la differenza, apparentemente inesistente, tra Missing e False Negative e tra Excess e False Positive. Va infatti notato come missing ed excess siano sottoinsiemi delle rispettive due classi indicate, le quali contengono anche gli error, dal momento che ogni annotazione sbagliata produce sia un falso positivo che un falso negativo.

Allo scopo di verificare alcune proprietà particolari del risultato, in alcuni punti del report viene fatto uso delle cosidette **named entity (NE)**, porzioni di testo per le quali è assegnabile, tra un insieme predefinito, una particolare categoria che le identifichi. L'individuazione di tale entità viene effettuata sfruttando le funzionalità del Named Entity Recognizer sviluppato dal Natural Language Processing Group dell'Università di Stanford.

3.2 Caratteristiche principali

Essendo la web application il mezzo principale di interazione con GERBIL, si è deciso di procedere con la realizzazione di una pagina web il cui contenuto potesse essere calcolato al termine di ogni task e memorizzato nel database insieme agli altri dati relativi all'annotazione già calcolati dal sistema, in modo tale da integrare la persistenza degli esperimenti garantendo la stessa caratteristica anche per i report. Il processo di creazione dei report è quindi il seguente: al termine di ogni task il sistema calcola tutti i dati di cui il resoconto finale avrà necessità ed essi vengono serializzati in formato JSON e memorizzati nella base dati persistente di GERBIL. All'atto della consultazione dei risultati di un esperimento, all'utente è data la possibilità, tramite collegamento ipertestuale associato ad ogni task

dell'esperimento, di visitare una pagina web contenente tutti i dati contenuti nel JSON memorizzato subito dopo il termine dell'esecuzione.

Il report è strutturato su tre macro-tipologie di contenuto: esiste una sezione dedicata a metriche, statistiche e analisi degli errori, tutte e tre declinate sia a livello globale che, con le dovute modifiche, a livello di singolo documento; oltre a questa prima sezione sono disponibili un set di statistiche relative al dataset, calcolate quindi a prescindere dall'annotazione, e infine una funzionalità di analisi dettagliata di falsi positivi, falsi negativi, missing ed excess.

GERBIL Experiment

Type: Sa2KB
Matching: STRONG_ANNOTATION_MATCH

Annotator	Dataset	Micro F1	Micro Precision	Micro Recall	Macro F1	Macro Precision	Macro Recall	Error Count	Timestamp	GERBIL version	Task Report	Compare
WAT	MSNBC	0,6897	0,7416	0,6446	0,6962	0,7613	0,6537	0	2015-11-07 16:04:46	1.1.4		
WAT	N3-Reuters-128	0,5146	0,5347	0,496	0,5121	0,6583	0,5749	0	2015-11-07 16:04:46	1.1.4		

Figura 3.1: Risultato di un esperimento con visualizzazione report e comparazione

3.3 Metriche

Le metriche di valutazione contenute nell'error report sono quelle che, esplicitamente o implicitamente, sono utilizzate di default all'interno di GERBIL e del BAT-Framework. Esse sono presenti sia a livello globale, in relazione quindi all'intero task di annotazione e all'intero dataset, che a livello di singolo documento. Sono presenti le tre metriche F1, precision e recall, definite a livello globale nelle versioni macro e micro e a livello di documento in versione classica; il conteggio delle annotazioni golden e actual e delle named entity; il numero di true positive, false positive e false negative.

3.4 Statistiche

Sono state definite, sia a livello di documento singolo che a livello di dataset (come agglomerazione delle singole), una serie di statistiche divisibili in tre categorie: **Entity type stats**, definite sulla base del concetto di classe di un'entità sfruttato in DBpedia, **NE stats**, definite sulla base dell'etichettatura delle named entity effettuato dallo Stanford Recognizer, e, infine, le **statistiche congiunte Entity type-NE**, che sono calcolate in funzione delle due tipologie precedenti.

3.4.1 Entity Type stats

Con Entity Type si intende la classe assegnata da DBpedia ad una specifica pagina Wikipedia (e quindi a una specifica entità), le statistiche sono state realizzate prendendo in

considerazione soltanto sette classi dell'ontologia di DBpedia: Character, Event, Location, Organization, Person, Product, Thing. Le Entity Type stats (Figura 3.2) associano

Entity Type Performance stats	CHARACTER	EVENT	LOCATION	ORGANIZATION	PERSON	PRODUCT	THING
Entity Type Correct	0	1	105	50	163	4	47
Entity Type Errors: Actual	0	2	14	6	7	0	13
Entity Type Errors: Golden	0	0	14	12	13	0	3
Entity Type Missing	0	0	48	50	43	3	94
Entity Type Excess	0	0	32	19	24	0	17

Figura 3.2: Entity Type stats

ad ogni Entity Type il numero di annotazioni, relative ad una determinata statistica, che appartengono a tale classe. Esistono al momento cinque statistiche appartenenti alla tipologia Entity type. La statistica che conteggia per ogni classe il numero di annotazioni correct ad essa associate prende il nome di **Entity Type Correct**. L'analoga statistica basata sul numero di annotazioni missing è invece chiamata **Entity Type Missing**, mentre quella che fa riferimento alle annotazioni in surplus (excess) è chiamata **Entity Type Excess**. Discorso a parte meritano le due tipologie di statistica implementate per effettuare l'analisi delle annotazioni classificate come error, esse sono infatti, a differenza delle altre tre, due statistiche separate ma strettamente collegate tra loro, dette **Entity Type Error Actual** e **Entity Type Error Golden**: nella prima delle due, ad ogni annotazione sbagliata viene incrementato il contatore della classe che l'entità dell'annotazione errata ha nell'insieme delle annotazioni prodotte dall'annotatore, nella seconda tipologia invece è il contatore della classe alla quale l'entità appartiene nel golden standard ad essere incrementato, per meglio comprendere queste ultime due statistiche si prenda ad esempio la Figura 3.2. In relazione alla statistica **Entity Type Error Golden** è possibile notare come, tra le annotazioni errate, 12 fossero collegate nel golden standard ad entità di classe ORGANIZATION. Per quanto riguarda invece la statistica **Entity Type Error Actual**, la tabella mostra che tra tutte le annotazioni sbagliate due siano state collegate ad entità aventi classe EVENT. In questo modo l'utente può valutare quanti e quali siano state le entità che l'annotatore ha sbagliato sul golden standard, e quali e quanti siano gli errori fatti in funzione del loro tipo.

3.4.2 NE stats

Questo genere di statistiche, un esempio delle quali è riportato in Figura 3.3, sfrutta il concetto di named entity precedentemente illustrato per individuare le corrispondenze tra l'annotazione e le entità presenti nel documento originario. Per calcolare tali statistiche il testo di ogni documento viene analizzato con il Named Entity Recognizer di Stanford, considerato lo stato dell'arte in questo campo, allo scopo di trovare le named entity presenti nel documento e assegnare loro una label, appartenente al seguente insieme: Person, Organization, Location, Misc. Una volta individuate le NE, si cercano nel golden standard le annotazioni la cui mention si sovrappone perfettamente ad una NE, e solo su

queste si calcolano le statistiche di seguito definite. Nella prima di esse, chiamata **NE Correct**, per ogni annotazione classificata come correct, la cui corrispondente annotazione golden è anche una NE, si incrementa il contatore corrispondente alla label della NE. Analoghe sono le statistiche **NE Missing**, che effettua un incremento del contatore di una determinata NE Label ogni qualvolta si incorra in un'annotazione individuata come missing la quale sia anche una named entity, e **NE Error**, nella quale per ogni annotazione errata la cui corrispondente annotazione golden, avente quindi stessa mention ma diversa entità, sia anche una named entity, viene incrementato il conteggio relativo alla sua label.

NER Performance stats	LOCATION	MISC	ORGANIZATION	PERSON
NE Correct	108	17	89	142
NE Missing	40	9	38	38
NE Error	17	3	6	13

Figura 3.3: NE stats

3.4.3 Statistiche congiunte Entity Type-NE

Le statistiche congiunte (Figura 3.4), come il nome suggerisce, associano le caratteristiche delle due tipologie precedenti e vengono sfruttate esclusivamente per analizzare gli errori. A differenza delle altre statistiche, rappresentabili per mezzo di una semplice tabella a singola entrata, i dati di queste statistiche sono rappresentati a coppie ⟨Entity Type, NE Label⟩. Come per la corrispondente Entity Type stat, viene sfruttato il dualismo actual/golden e vengono create due statistiche, la prima, che prende il nome di **Entity Type-NE Errors Actual** prevede che per ogni annotazione golden che ha una corrispondente annotazione errata assegnata dall'annotatore testato e che è riconosciuta come NE, si verifichi la classe DBpedia assegnata nell'actual all'annotazione ritrovata e si incrementi il valore del contatore relativo alla coppia ⟨Entity Type, NE Label⟩, che rappresenta, per ogni classe DBpedia, il numero di annotazioni sbagliate appartenenti a tale categoria che corrispondono ad una named entity con una certa Label; la controparte di questa statistica, indicata nel report sotto il nome di **Entity Type-NE Errors Golden** è calcolata in maniera congiunta con la precedente, con l'unica differenza che ad essere incrementata è la coppia formata dalla NE Label con la classe DBpedia che l'annotazione errata ha nel golden standard, e non nell'output dell'annotatore. Si prendano ad esempio i dati riportati in Figura 3.4, è possibile notare come tra le annotazioni sbagliate, tre siano state collegate a concetti di classe ORGANIZATION sebbene la loro mention corrispondesse ad una named entity di tipo LOCATION (statistica **Entity Type-NE Errors Actual**). La statistica **Entity Type-NE Errors Golden** offre invece una misura della corrispondenza tra le annotazioni golden e la classe assegnata alla named entity ad essa corrispondente, è visibile in figura come ad esempio tra le annotazioni sbagliate, 6 avessero nel golden standard un collegamento con un'entità Wikipedia di tipo ORGANIZATION, sebbene la corrispondente named entity le classificasse come LOCATION.

Entity Type-NE Errors Actual	LOCATION	MISC	ORGANIZATION	PERSON
CHARACTER	0	0	0	0
EVENT	2	0	0	0
LOCATION	6	1	0	6
ORGANIZATION	3	0	3	0
PERSON	0	0	0	7
PRODUCT	0	0	0	0
THING	6	2	3	0

Entity Type-NE Errors Golden	LOCATION	MISC	ORGANIZATION	PERSON
CHARACTER	0	0	0	0
EVENT	0	0	0	0
LOCATION	11	0	1	0
ORGANIZATION	6	1	4	0
PERSON	0	0	0	13
PRODUCT	0	0	0	0
THING	0	2	1	0

Figura 3.4: Statistiche congiunte Entity Type-NE

3.5 Risultati dell'annotazione

Come già anticipato, il resoconto sull'annotazione consente due tipologie di visualizzazione: quella d'insieme, con metriche e statistiche calcolate sull'intero dataset e quella dettagliata, relativa ai singoli documenti.

3.5.1 Visualizzazione globale

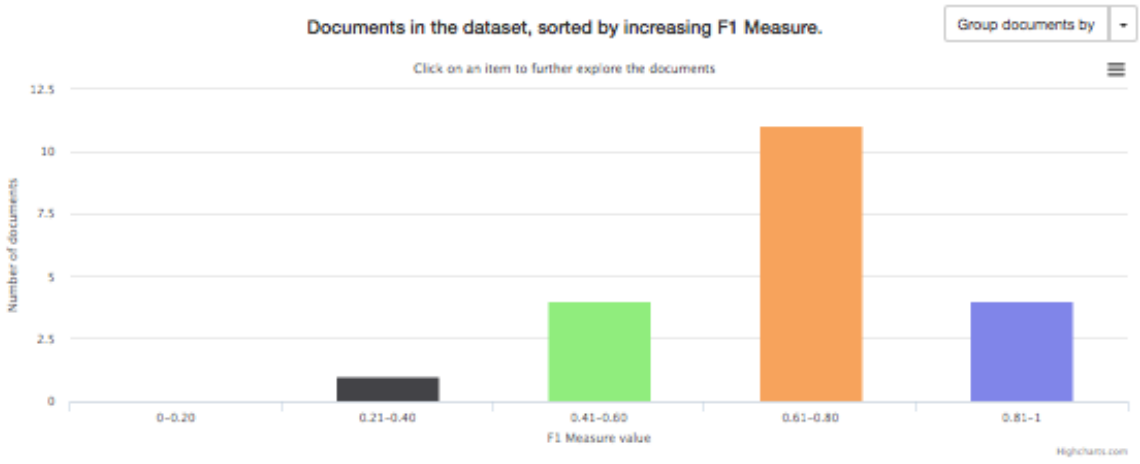
La prima schermata di dati analizzabili è quella relativa alla visione d'insieme sull'esperimento: sono quindi presenti le misure di valutazione e le statistiche aggregate relative alla performance dell'annotatore. A partire da questa schermata, riportata in Figura 3.5 è possibile analizzare nel dettaglio i singoli documenti. Tale opportunità è resa possibile da un plot navigabile, il quale permette di effettuare un'iniziale suddivisione dei documenti in cinque gruppi, calcolati sulla base dei loro valori crescenti di F1, precision o recall. All'interno di ognuno dei cinque gruppi è possibile analizzare i documenti ad esso corrispondenti ordinandoli per lunghezza e numero di annotazioni crescente, oltre che per le tre metriche usate al livello precedente. Ognuno dei documenti visualizzati a questo livello è selezionabile per la visione dettagliata.

3.5.2 Singolo documento

La visione dettagliata di un documento mostra le statistiche dettagliate ad esso relative. In questa tipologia di visualizzazione, la struttura della pagina è organizzata in maniera simile a quella globale riportata Figura 3.5, sebbene divisa in quattro aree principali, la prima delle quali, come avviene per la controparte globale, contiene le **metriche**, analoghe a quelle calcolate a livello globale ma relative in questo caso al singolo documento, che vedono quindi la distinzione di macro- e micro- F1, precision e recall lasciare il passo al concetto classico di F1, precision e recall. Proseguendo nella lettura del resoconto è possibile visualizzare un **dettaglio sul documento**, utile per indagare al meglio sul comportamento di un annotatore nel caso di un determinato testo. Questa sezione mette infatti a disposizione una visualizzazione side-by-side che permette di analizzare sia il documento annotato con il golden standard, e corredato di informazioni sulle NE ri-

Experiment Type: Sa2KB - **Matching:** STRONG_ANNOTATION_MATCH - **Annotator:** WAT - **Dataset:** MSNBC

Metrics							
Micro F1	0.689712	Macro F1	0.69622374	True Positives	419	Total Gold Annotations	650
Micro Precision	0.74158294	Macro Precision	0.7612515	False Negatives	231	Total Stanford NEs	1836
Micro Recall	0.6446154	Macro Recall	0.6537063	False Positives	146	Retrieved Annotations	565



Entity Type Performance stats	CHARACTER	EVENT	LOCATION	ORGANIZATION	PERSON	PRODUCT	THING
Entity Type Correct	0	0	126	59	157	6	71
Entity Type Errors: Actual	0	0	13	3	4	0	15
Entity Type Errors: Golden	0	1	14	10	3	0	7
Entity Type Missing	0	0	27	41	59	1	68
Entity Type Excess	0	0	46	8	24	4	29

NER Performance stats	LOCATION	MISC	ORGANIZATION	PERSON
NE Correct	123	16	93	146
NE Missing	21	11	35	44
NE Error	21	2	5	3

Entity Type-NE Errors Actual	LOCATION	MISC	ORGANIZATION	PERSON
CHARACTER	0	0	0	0
EVENT	0	0	0	0
LOCATION	12	1	0	0
ORGANIZATION	0	0	3	0
PERSON	0	0	0	2
PRODUCT	0	0	0	0
THING	9	1	2	1

Entity Type-NE Errors Golden	LOCATION	MISC	ORGANIZATION	PERSON
CHARACTER	0	0	0	0
EVENT	1	0	0	0
LOCATION	11	0	1	0
ORGANIZATION	6	1	3	0
PERSON	0	0	0	1
PRODUCT	0	0	0	0
THING	3	1	1	2

Figura 3.5: Vista globale dei risultati di un'annotazione

trovate dallo Stanford recognizer, che il documento così come è stato annotato durante l'esperimento, avendo la possibilità di scegliere quali annotazioni vedere in base alla loro tipologia: Correct, Error, Missing, Excess. Un esempio di tale visualizzazione è mostrato

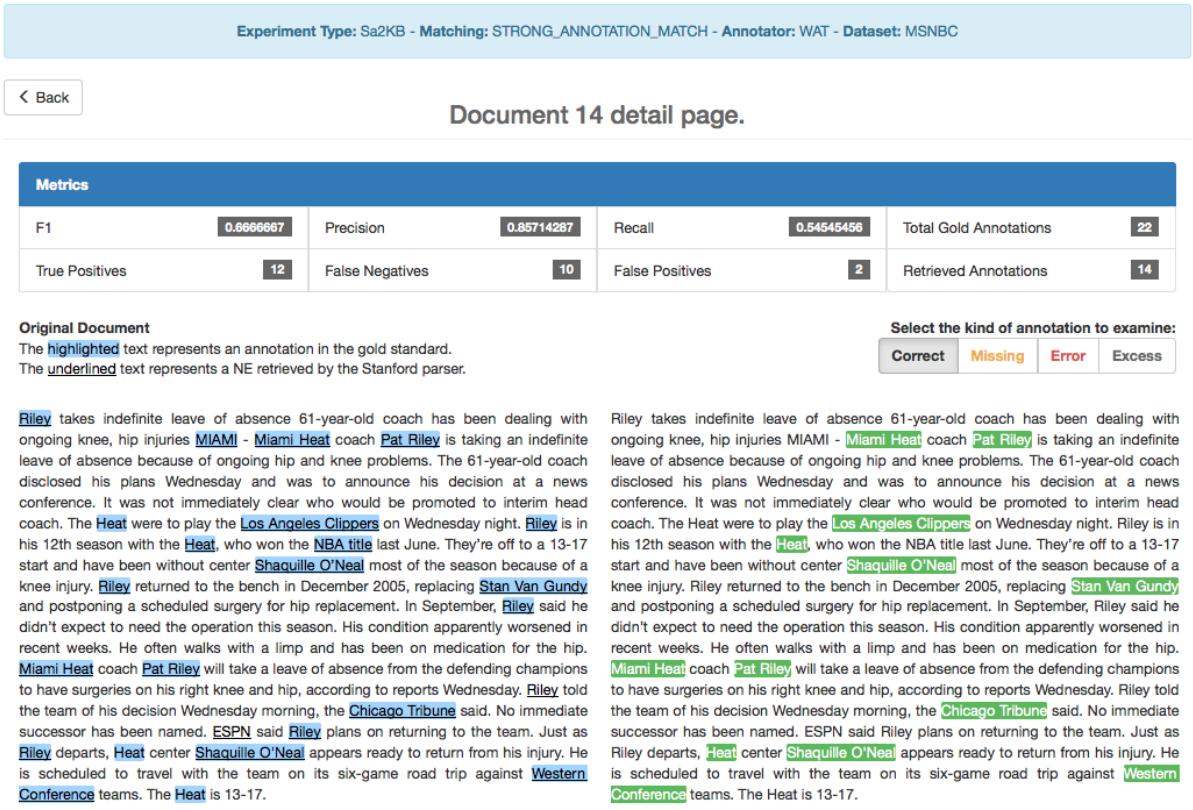


Figura 3.6: Dettaglio visualizzazione singolo documento

in Figura 3.6. Come è possibile notare, al di sotto della tabella contenente le metriche relative al documento, è presente la sezione dedicata al testo del documento. La colonna di sinistra rappresenta il golden standard per il documento: sono evidenziate in azzurro le porzioni di testo corrispondenti ad un'annotazione, e sottolineate le porzioni riconosciute come named entity dallo Stanford Parser. La colonna a destra rappresenta invece l'output dell'annotatore, la visualizzazione di default mostra le annotazioni correttamente individuate, ed è possibile con l'utilizzo degli appositi pulsanti visualizzare le altre tipologie di annotazione prodotte. Per ognuna delle annotazioni, siano esse golden o actual, è sufficiente un passaggio del mouse per visualizzare le corrispondenti proprietà (i.e. Classe DBpedia, ID Wikipedia, NE Label qualora si tratti di una named entity). Come nella visualizzazione d'insieme è possibile avere una panoramica del comportamento dell'annotatore in relazione alle categorie di appartenenza delle annotazioni del documento attraverso le **statistiche**, punto di aggregazione delle statistiche precedentemente presentate, calcolate sulla base del singolo documento, le quali aggregate sommando i vari campi danno origine alle statistiche relative all'intero task.

È infine presente a disposizione dell'utente una sequenza di **informazioni sulle annotazioni**, la quale attraverso due grafici offre una rappresentazione del posizionamento

delle annotazioni golden all'interno del documento. In entrambi i grafici la lunghezza del documento viene normalizzata a 100 per calcolare, a seconda della posizione nel documento e del grafico considerato, il numero di annotazioni e la percentuale delle annotazioni totali del documento presenti in ogni posizione del testo.

3.5.3 Statistiche sul dataset

Questa tipologia di statistiche è totalmente indipendente dall'esperimento di annotazione svolto, e mira esclusivamente a fornire informazioni generiche sulla struttura del dataset in oggetto. L'idea alla base di questa analisi è di fornire gli strumenti necessari a collegare determinati comportamenti dell'annotatore a particolari caratteristiche della struttura del dataset (e.g. distribuzione delle annotazioni tra le classi DBpedia, ecc.), oltre che di esprimere caratteristiche base relative ai documenti e alle annotazioni in esso contenute.

La prima tipologia di statistiche calcolate per il dataset (Figura 3.7) comprende valori che offrono una misura delle caratteristiche fondamentali del dataset: numero complessivo di documenti, numero complessivo di annotazioni contenute nel golden standard, conteggio delle NE trovate dallo Stanford Parser all'interno dei documenti, lunghezza media di un documento e di una annotazione, numero medio di annotazioni per documento. Un'altra interessante statistica permette di avere un colpo d'occhio sulla distribuzione delle classi di DBpedia tra le annotazioni golden e delle label tra le NE individuate dallo Stanford Parser. Per entrambe le tipologie di categorizzazione è infatti presente un conteggio delle istanze contenute nel dataset divise per classe/label.

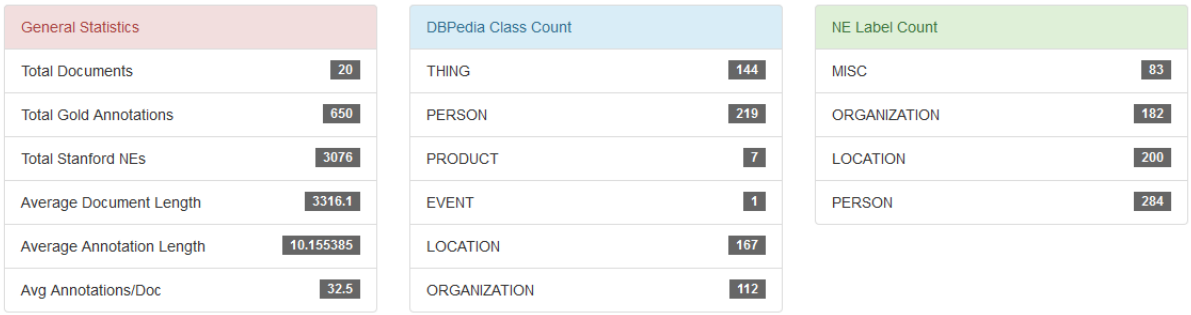


Figura 3.7: Statistiche sul dataset

Infine è possibile visualizzare tre grafici, riportati in Figura 3.8, relativi alle annotazioni golden contenute nei documenti: due di essi sono equivalenti a quelli mostrati per ogni singolo documento, e mostrano la distribuzione delle annotazioni nei vari documenti normalizzandone la lunghezza a 100 e aggregando il conteggio su ogni singolo documento per offrire una vista globale del numero di annotazioni e della percentuale di annotazioni totali espressi in funzione della posizione nel testo. Il terzo plot mostra per ogni documento il numero di annotazioni golden in esso contenuto.

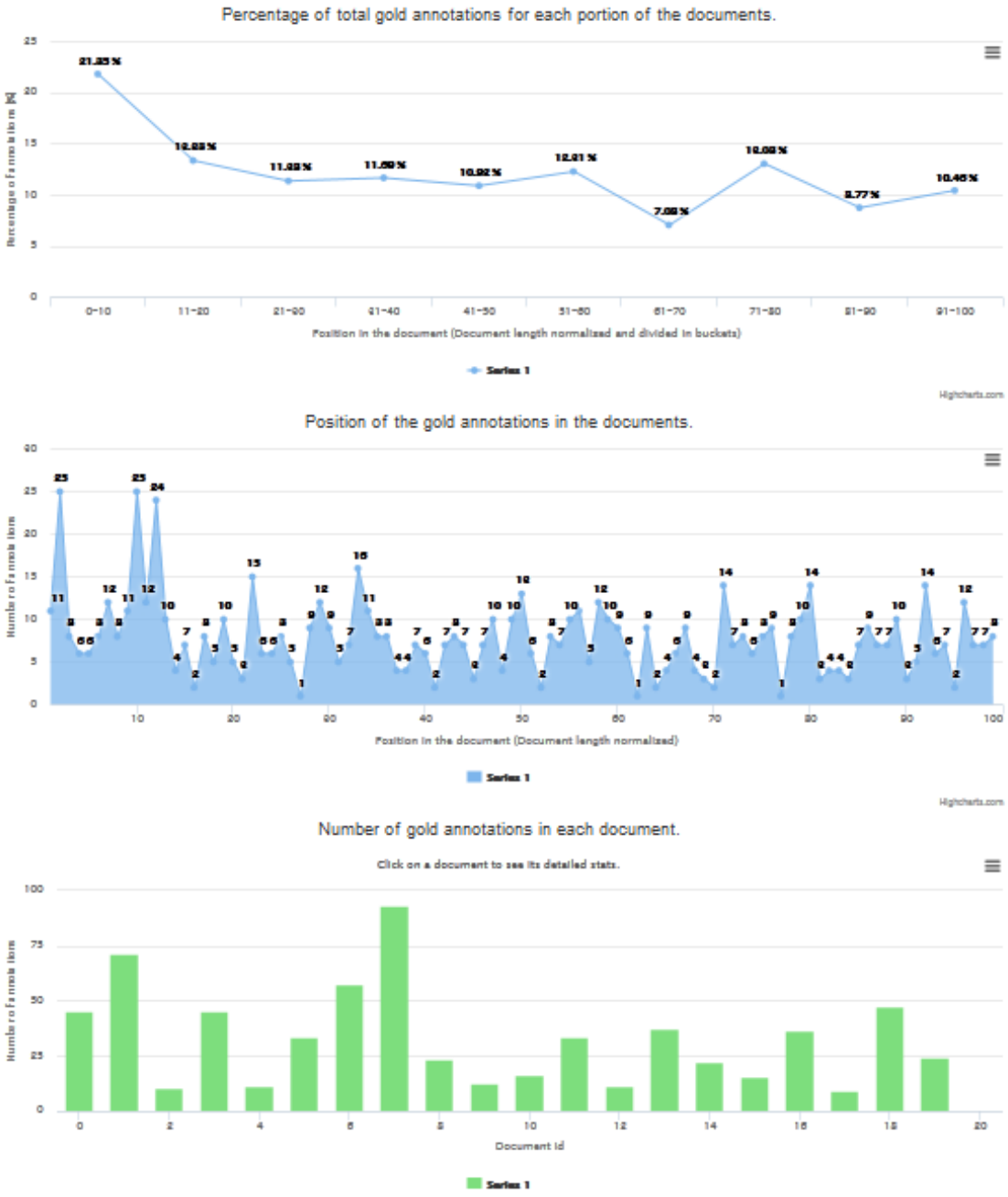


Figura 3.8: Grafici sulla distribuzione delle annotazioni nel dataset

3.5.4 Analisi FP/FN/Missing ed Excess

L'ultima categoria di statistiche presenti nell'error report è relativa all'analisi dettagliata di False Positive *(fp)*, False Negative *(fn)*, Missing ed Excess. Per quanto riguarda falsi positivi e falsi negativi, il resoconto mostra, per ogni documento, la lista dei suoi fp ed fn, riportando le frasi del testo originale in cui le annotazioni oggetto dell'analisi sono contenute (si veda Figura 3.9). Diversa è invece la prospettiva utilizzata nell'analizzare

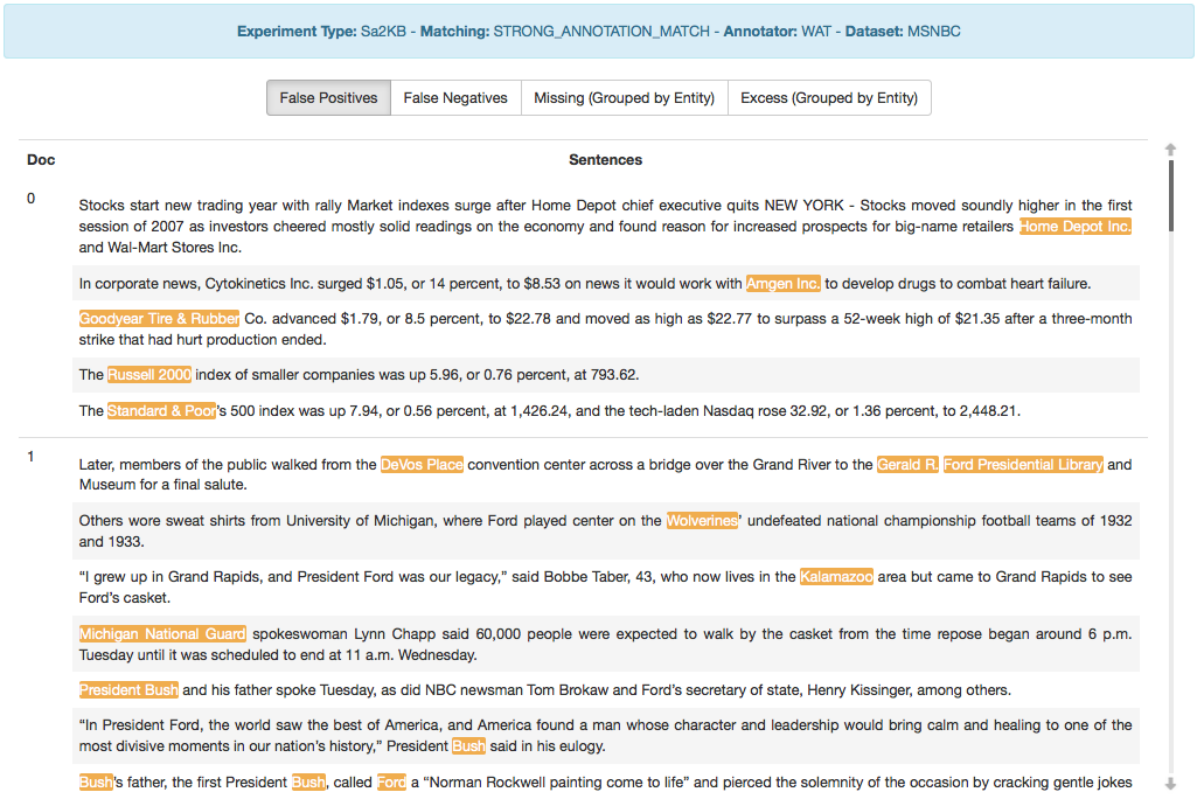


Figura 3.9: Analisi dei falsi positivi divisi per documento

Missing ed Excess, essi sono infatti raggruppati per entità, e mostrano, per ogni pagina di Wikipedia corrispondente ad un'annotazione Missing o Excess, l'elenco delle mention ad essa collegate, corredato di conteggio delle occorrenze di tale mention all'interno del dataset. L'utilità di tale visualizzazione è, ad esempio nel caso delle missing annotations (Figura 3.10, di verificare numericamente e in maniera rapida la tipologia di annotazione sfuggita all'annotatore in modo da individuare eventuali pattern di errore e fronteggiarli adeguatamente, migliorando le performance del sistema.

Analogo è il caso delle excess annotation, che permette di individuare quelle situazioni in cui l'annotatore è tratto in inganno annotando porzioni di testo che dovrebbe invece trascurare.

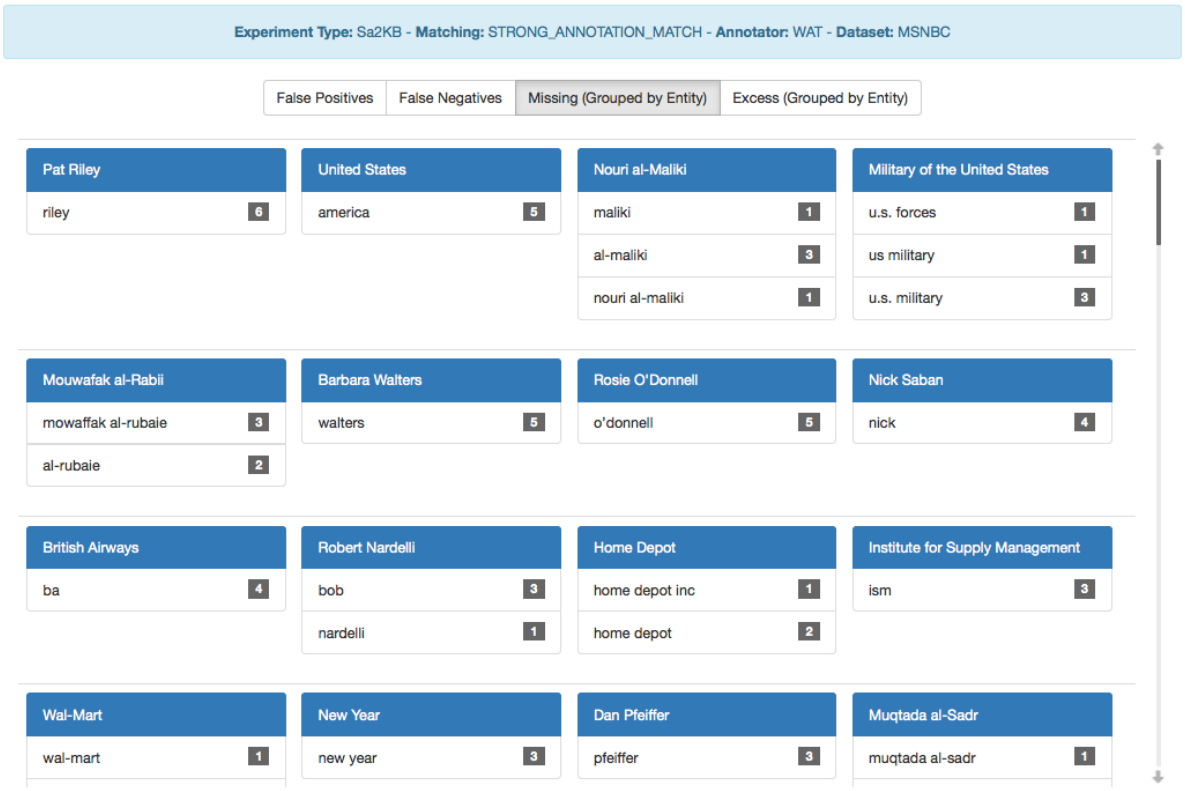


Figura 3.10: Analisi dei missing raggruppati per entità (L'intestazione di ogni blocco è il titolo dell'entità Wikipedia indicata nel golden standard per le menzioni riportate nella tabella, per ognuna delle quali è indicato il numero di occorrenze.)

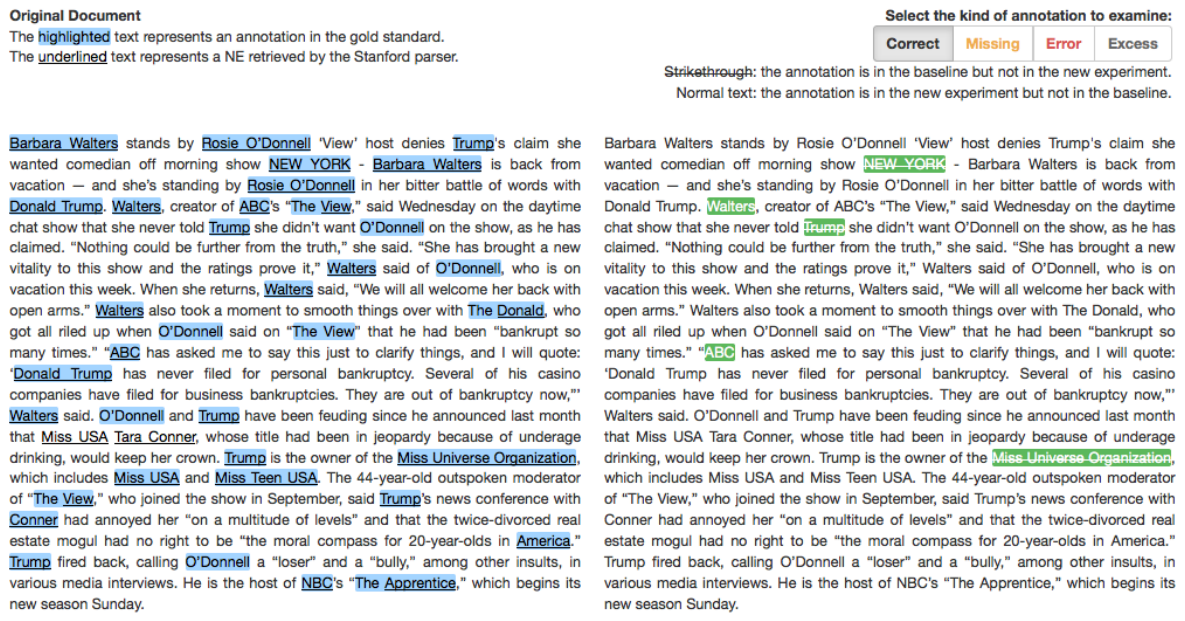


Figura 3.11: Differenza tra annotazioni correct di un documento

3.5.5 Differenza tra due report

Per rendere più facile l'individuazione delle differenze presenti tra due report relativi a differenti task di annotazione, è stata implementata una funzionalità di calcolo della differenza tra due report. Come rappresentato in Figura 3.1, alla tabella dei risultati di GERBIL, oltre a quello necessario alla visualizzazione del report, è stato aggiunto un ulteriore pulsante che permette di raggiungere una schermata la quale consente di scegliere, per un dato task, un altro tra quelli presenti nel database che condividono le sue stesse caratteristiche (i.e. problema, matching, dataset) in modo da poterlo confrontare con quello di partenza. Il risultato è una pagina web identica nelle sue parti a quella utilizzata per un singolo report, ma nella quale i valori presenti sono espressi come differenza (aritmetica quando non diversamente specificato) tra i valori del task di partenza (indicato anche come *baseline*) e il nuovo task. I valori numerici relativi a metriche e statistiche, sia globali che relative ai singoli documenti, sono espressi come differenza aritmetica tra i corrispondenti valori contenuti nei due report, ad esempio dati due valori per la metrica F1, $F1_b$ (Il valore di F1 dell'esperimento baseline) ed $F1_n$ (Il valore di F1 dell'esperimento con cui si sta effettuando il confronto), il valore espresso nel campo F1 della differenza sarà $F1_b - F1_n$ ad esprimere un aumento o una diminuzione di quel valore nel nuovo task rispetto alla baseline. All'interno della schermata di valutazione globale del task è possibile dividere i documenti in cinque gruppi come nella versione base del report, utilizzando come criterio di divisione le metriche F1, precision e recall sia della baseline che del nuovo task. All'interno dei cinque sottoinsiemi è possibile ordinare anche per valori crescenti delle differenze tra le tre metriche, oltre che per i valori standard precedentemente presentati. Le informazioni contenute nelle sezioni del resoconto aventi a che fare con annotazioni, quali dettaglio del documento annotato, dettaglio fp ed fn, sono elaborate come segue: dati due insiemi di annotazioni fanno parte della differenza le annotazioni presenti nel primo ma non nel secondo (indicate col testo sbarrato nella grafica del report) e quelle non presenti nel primo ma presenti nel secondo (rappresentate come normali annotazioni). Non sono disponibili le differenze per il dettaglio di missing ed excess essendo concettualmente poco significative nell'ambito della differenza tra due esperimenti. In figura 3.11 è possibile osservare la rappresentazione scelta per la differenza tra due output di annotazione. La struttura, mutuata dalla visualizzazione precedentemente illustrata, mantiene nella colonna di sinistra la rappresentazione del golden standard per il documento considerato, riportando evidenziate in azzurro le annotazioni golden e sottolineate le named entity individuate dallo Stanford Parser. La sezione destra mira invece, mantenendo la visualizzazione separata nelle quattro classi Correct, Missing, Error ed Excess, a concentrarsi esclusivamente sulle differenze tra due report. Si prenda ad esempio la visualizzazione delle annotazioni Correct mostrata in figura, tenendo presente che il medesimo discorso si applica anche alle rimanenti categorie; in tale modalità di visualizzazione non sono riportate le annotazioni presenti in entrambi i report confrontati ma soltanto quelle ottenute filtrando gli output con il criterio precedentemente menzionato: fanno parte della differenza soltanto le annotazioni presenti nel primo risultato (la baseline) ma non nel secondo, che sono rappresentate mediante il testo sbarrato, e quelle presenti nel secondo risultato ma non nella baseline, rappresentate come normali annotazioni.

Capitolo 4

TAGME

Sviluppato nel 2010 nel dipartimento di Informatica dell'Università di Pisa, TAGME è un annotatore testuale nato con lo scopo ben preciso di specializzare la sua attività su documenti molto brevi, tipicamente composti da poche decine di termini.

4.1 Caratteristiche

L'idea iniziale alla base della creazione di questo strumento era quella di applicare le tecniche dell'annotazione a testi di natura poco strutturata quali tweet, notizie flash ecc.. Questa tipologia di documento porta con se due importanti problematiche, la risoluzione delle quali era alla base dell'idea di sviluppo di TAGME: la necessità di effettuare annotazioni on-the-fly, data l'impossibilità di preprocessare i testi a causa della loro natura volatile e di fruizione a tempo di query; la necessità di nuove tecniche algoritmiche in grado di fare a meno delle analisi statistiche caratterizzanti la gran parte degli strumenti già presenti nella letteratura. Partendo dai principi e dagli obiettivi appena presentati è stato progettato un sistema di annotazione facente utilizzo di articoli di Wikipedia come entità e che sfrutta la struttura di àncore e link insita nell'organizzazione a grafo delle pagine Wikipedia per individuare i possibili spot del testo e i loro possibili significati. L'idea principale era quindi di realizzare un annotatore che, facendo pieno uso della struttura della base di conoscenza scelta per l'annotazione, fosse in grado di produrre risultati di buona qualità su testi particolarmente ardui da annotare, in tempi di esecuzione ristretti.

4.2 Fasi dell'annotazione

Il processo di annotazione di TAGME si articola in quattro fasi, delle quali una preliminare di indicizzazione e tre di annotazione vera e propria. Nella fase preliminare vengono creati indici che sfruttano informazioni contenute in Wikipedia da utilizzare in seguito nelle tre fasi dell'annotazione. L'annotazione di un documento da parte di TAGME vede una prima fase di parsing del testo iniziale, seguita dalla fase di disambiguazione e termina con una fase di potatura (pruning) dei risultati trovati.

4.2.1 Creazione degli indici

Allo scopo di agevolare la procedura di annotazione, TAGME si avvale di una serie di indici, creati a partire da uno snapshot di Wikipedia, che fungono da struttura di supporto all'annotazione.

Anchor dictionary La creazione di questa struttura inizia con la selezione di tutte le àncore presenti nelle pagine di Wikipedia, che vengono arricchite con i titoli delle pagine alle quali fanno riferimento. Tale dizionario di àncore viene in seguito ripulito di tutte quelle istanze le cui àncore risultano composte da caratteri singoli o da numeri e di quelle in cui una determinata àncora appare collegata ad una determinata pagina in un numero esiguo di volte, in quanto sfruttare questo genere di informazioni potrebbe portare a deviazioni nel processo di annotazione;

Page catalog Il catalogo delle pagine viene creato a partire da tutte le pagine di Wikipedia, fatta eccezione per quelle di disambiguazione, le liste e le redirezioni, in quanto non rappresentando un'entità univoca, il loro utilizzo non risulta adatto agli scopi dell'annotazione testuale;

In link graph Si tratta di un grafo orientato i cui nodi sono le pagine contenute nel catalogo delle pagine, e i cui archi sono i collegamenti tra queste pagine ricavati dalla struttura di Wikipedia.

È bene notare che la creazione di queste strutture di supporto non è direttamente legata alla fase di annotazione vera e propria, essa avviene infatti a monte di ogni task di annotazione, ed è eseguita una tantum.

4.2.2 Parsing

La prima attività svolta dal sistema dopo la ricezione di un testo in input è quella di tokenizzarlo, ossia dividerlo in porzioni più piccole, dette token, costituite da una determinata sequenza di caratteri del testo e utilizzate come unità base per tutte le fasi successive. I token ottenuti a partire dal testo iniziale vengono quindi analizzati alla ricerca di àncore che potrebbero diventare menzioni da annotare nelle fasi successive.

4.2.3 Disambiguazione delle àncore

Una volta individuate all'interno del testo le àncore che possono essere collegate ad entità presenti nella Wikipedia, ha inizio la fase cruciale del processo di annotazione: ad ogni àncora viene collegata un'entità univoca che la descrive, scelta tra tutti i significati possibili ad essa collegati. Una delle principali novità algoritmiche introdotte da TAGME rispetto ai sistemi di annotazione pre-esistenti è costituita proprio dalla innovativa strategia di disambiguazione dei possibili significati delle àncore. Dato un insieme di àncore individuate all'interno di un testo, TAGME assegna ad ognuna di esse il signifcato più opportuno, calcolando per ogni senso possibile un punteggio, che esprime la pertinenza di una determinata pagina rispetto all'àncora considerata. Tale punteggio è calcolato per

mezzo di un cosiddetto *voting scheme*, nel quale ogni àncora esprime un voto per ogni altra annotazione prodotta a partire dal documento iniziale. Supponendo di voler calcolare quindi il punteggio relativo all'annotazione di un'àncora a con una determinata pagina p_a , si fa in modo che ogni altra àncora b presente nell'insieme delle àncore, esprima il suo voto sull'annotazione. Dal momento che ogni àncora può avere più di un senso accettabile, il voto è calcolato come correlazione media (*average relatedness*) tra ogni possibile senso p_b dell'àncora b e il senso p_a che si sta cercando di associare all'àncora a. La misura della correlazione tra due pagine Wikipedia p_a e p_b è espressa dalla formula:

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

nella quale *in(p)* è l'insieme di pagine di Wikipedia che puntano alla pagina p e W è il numero di pagine di Wikipedia. Pertanto il voto che l'àncora b esprime in relazione all'annotazione tra a e p_a è esprimibile come:

$$vote_b(p_a) = \frac{\Sigma_{p_b \in Pg(b)} rel(p_b, p_a) \cdot Pr(p_b|b)}{|Pg(b)|}$$

Terminata la procedura di votazione, il voto complessivo dell'annotazione in oggetto è calcolato come la somma dei voti dati da tutte le àncore presenti nel testo e non coinvolte nella presente annotazione. Tale voto è utilizzato ai fini della disambiguazione in concomitanza con un'altra misura, la cosiddetta *commonness*, che indica quanto frequente sia per una certà entità essere utilizzata quale significato di una determinata àncora. Dei numerosi approcci plausibili per combinare la commonness e il voto espresso per un'annotazione, all'interno di TAGME ne sono presenti due: uno che fa utilizzo di un classificatore che, partendo dalle due misure, calcola un valore rappresentante la probabilità che la disambiguazione sia stata effettuata correttamente collegando l'àncora a all'entità p_a; un altro che invece evita l'utilizzo di un classificatore e individua invece il senso migliore nel seguente modo: per prima cosa determina il senso p_{best} che ha la più alta relatedness con l'àncora a, dopodiché seleziona, tra tutti i possibili sensi di a, quelli che raggiungono valori simili di relatedness con l'àncora, usando una soglia predefinita ϵ. Infine annota l'àncora a con il senso p_a che ottiene la più alta *commonness* $Pr(p_a|a)$ tra i migliori sensi precedentemente individuati. Va sottolineato come la procedura di votazione abbia impatto esclusivamente sulle àncore con più di un significato, infatti qualora un àncora abbia un unico senso esso sarà quello prodotto dalla disambiguazione.

4.2.4 Potatura delle àncore

La fase di disambiguazione delle àncore porta alla produzione di un insieme di annotazioni, contenente un'annotazione per ogni àncora individuata nel testo fornito in input. Per fare in modo che esclusivamente le annotazioni significative facciano parte del risultato finale. La tecnica adottata da TAGME per individuare le annotazioni di scarso interesse è quella di calcolare un punteggio basato su due soli dati: la probabilità che la menzione a considerata linki alla pagina p_a (*link probability*) e la coerenza tra l'annotazione considerata e le altre annotazioni presenti nel testo. Lo scopo di questa fase è mantenere nel risultato finale quelle annotazioni la cui link probability sia alta o il cui senso sia particolarmente coerente con quelli assegnati alle altre annotazioni rilevanti.

4.3 Prestazioni

Un test intensivo che confrontasse le prestazioni di TAGME con quelle degli altri annotatori presenti all'epoca è stato realizzato in concomitanza con la creazione del BAT-Framework. In tale occasione sono stati organizzati in tutto tre esperimenti, differenziati tra loro dalla tipologia dei documenti da annotare: notizie, tweet e pagine web.

Primo esperimento: Notizie

I dataset a disposizione per questo genere di input presentavano documenti brevi in alcuni casi e più lunghi in altri. Sebbene il punto di forza di TAGME siano i testi brevi, le misure calcolate durante l'esperimento hanno sorprendentemente mostrato la sua efficacia anche su documenti più lunghi. La valutazione dei risultati dell'esperimento è stata condotta concentrando le attenzioni sulla metrica micro-F1, calcolata sia con vincoli di matching strong che weak. I risultati globali hanno mostrato come TAGME fosse l'annotatore con le migliori prestazioni in entrambe le tipologie di matching e di documento, sebbene sia stata rilevata una generalizzata flessione delle prestazioni in concomitanza con l'utilizzo del più restrittivo dei criteri di matching, segno di un problema diffuso ad identificare in maniera perfetta come menzioni le porzioni di testo presenti nel golden standard.

Secondo esperimento: Tweet

L'obiettivo di questo esperimento era di testare i vari sistemi di annotazione nel caso in cui i testi da annotare fossero molto brevi e di bassa qualità, caratteristiche tipiche dei tweet. Tali caratteristiche costituiscono una difficoltà per i riconoscitori di named entity, le cui performance subiscono quindi delle flessioni quando applicati a questo genere di documento. Per questo motivo è facile immaginare come le performance complessive fossero prevedibilmente inferiori rispetto al precedente esperimento. Considerato anche che dei competitor TAGME era l'unico ad essere progettato specificatamente per questa tipologia di analisi, l'esperimento serviva più che da misura assoluta di raffronto, da esperimento della flessibilità dei vari tool di annotazione nell'affrontare tipologie di dati ad essi poco congeniali. Il risultato finale dell'esperimento ha visto TAGME classificarsi secondo, subito dietro a WikipediaMiner, sistema caratterizzato da un ottima misura di recall, che contribuisce notevolmente all'incremento del valore della metrica considerata nel confronto, la micro-F1.

Terzo esperimento: Pagine Web

Il terzo ed ultimo test messo in pratica nel raffronto intensivo tra i diversi sistemi di annotazione, aveva come obiettivo la verifica delle prestazioni nell'ambito di testi lunghi estratti da pagine web. Le premesse lasciavano ovviamente presagire un calo della performance di TAGME, data la sua natura di annotatore di testi di scarsa lunghezza. La misura di coerenza utilizzata durante la fase di potatura dei risultati rischia infatti di compromettere la buona riuscita dell'annotazione in documenti lunghi, che possono presentare, anche per annotazioni golden, una variazione di concetti che ne abbassano il valore. I risultati di questo esperimento, come previsto, hanno visto primeggiare sistemi

progettati per l'annotazione di porzioni di testo lunghe, con TAGME quarto classificato, a precedere, nonostante le sue caratteristiche non fossero pienamente assecondate dalla tipologia di task, sistemi sulla carta più adatti a tale genere di annotazione.

Vale la pena sottolineare che, oltre alla qualità del risultato espressa mediante le misure di valutazione, riveste importanza nell'analisi delle prestazioni anche l'efficienza di esecuzione. È infatti facilmente intuibile come un'annotazione eseguita rapidamente sia più facilmente inseribile in contesti classici dell'Information Retrieval che necessitano di un'interazione a query-time con l'utente. Con questo concetto in mente è stato quindi realizzato, oltre agli esperimenti precedentemente presentati, anche un test temporizzato che consentisse di valutare le prestazioni in termini di tempo dei vari annotatori. Tale test è stato svolto cercando di rimuovere, quanto più fosse possibile, ogni possibile latenza nell'esecuzione, compito non facile vista la diversa struttura procedurale dei vari sistemi, che ha richiesto opportune contromisure valutate caso per caso perché i tempi calcolati non risultassero influenzati dall'ambiente circostante. Il responso dell'esperimento ha messo in evidenza che TAGME era l'annotatore più veloce tra quelli testati, con una velocità 10 volte superiore a quella del secondo classificato.

Al termine della fase di testing risultò quindi che TAGME superava, sia in termini di qualità del risultato, che in termini di tempo di esecuzione, gli annotatori precedentemente pubblicati, caratterizzandosi così come stato dell'arte tra i software di annotazione testuale.

Capitolo 5

Da TAGME a WAT

Sfruttando la tecnologia algoritmica introdotta in TAGME, nel 2014 è stato sviluppato, nel dipartimento di Informatica dell'Università di Pisa, WAT, un nuovo annotatore che rappresenta il successore di TAGME, sfruttandone i punti di forza e introducendo innovazioni in tutte le fasi dell'annotazione, cercando di migliorarne ulteriormente le prestazioni. Tra le novità introdotte da WAT è interessante annoverare la rinnovata struttura del software, reingegnerizzata allo scopo di creare un sistema modulare e più efficiente del suo predecessore, nonché il miglioramento dell'attività di annotazione, ottenuto grazie all'intera riscrittura delle tre componenti principali di TAGME: spotter, disambiguatore e pruner. Tale riscrittura è stata portata avanti mediante attività specifiche di analisi di ogni singola componente, integrate dall'implementazione di nuovi algoritmi, testati su tutti i dataset di annotazione disponibili pubblicamente.

5.1 Fasi dell'annotazione

Come già anticipato, tutte e tre le componenti principali di TAGME sono state riscritte durante la creazione di WAT, si vedrà ora nel dettaglio quali siano le novità introdotte in questo nuovo sistema di annotazione.

5.1.1 Spotting

La prima fase dell'annotazione prevede il parsing del documento da annotare, allo scopo di invididuare i cosiddetti spot, porzioni di testo candidate ad essere annotate. L'insieme di tutte le possibili menzioni riconoscibili da WAT è ottenuto, come nel caso di TAGME, da un pre-processing del grafo di Wikipedia. Oltre ai già visti concetti di àncore, titoli e redirezioni, WAT introduce in associazione ad ogni mention contenuta nel database dei possibili spot, un attributo denominato **link probability** (lp), il quale, in relazione ad una mention m viene calcolato come $lp(m) = link(m)/freq(m)$, dove con *freq(m)* si indica il numero di volta che la menzione m appare dentro Wikipedia, mentre *link(m)* denota il numero di volte che la menzione m appare come àncora nelle pagine di Wikipedia. Per ogni menzione m, il database degli spot mette a disposizione una lista di entità E_m papabili, ordinate utilizzando come criterio il numero di volte che la menzione m è utilizzata per linkare l'entità e, misura questa definita come **commonness**. In aggiunta agli attributi

classici, ogni menzione contenuta nel database degli spot in WAT è corredata da un insieme di nuove statistiche utilizzate da WAT per il training di un classificatore binario (opzionale) utilizzato per migliorare le prestazioni della fase di individuazione degli spot contenuti nel testo. Tra i nuovi dati introdotti è annoverabile un insieme di caratteristiche strettamente legate alle caratteristiche sintattiche di una mention, oltre ad una sequenza di valori relativi alla classe di appartenza della mention, alle pagine contenenti la mention e alla posizione della mention considerata rispetto ad altre possibili mention presenti nel documento.

5.1.2 Disambiguazione

All'interno di WAT è presente un vasto insieme di algoritmi di disambiguazione, utilizzati per assegnare ad una mention la più pertinente delle entità ad essa associate nel database degli spot. I metodi di disambiguazione implementati in WAT possono essere distinti in due categorie: quelli basati sul voting scheme introdotto in TAGME e una nuova categoria, basata sul cosiddetto *Mention-Entity graph*. Entrambe le tipologie di disambiguazione vengono arricchite dall'introduzione del concetto di finestra concettuale, ossia un contesto circostante una menzione m, utilizzato per migliorare la fase di disambiguazione tenendo conto della porzione di documento situata nelle vicinanze della menzione considerata.

5.1.2.1 Algoritmi voting-based

Si è precedentemente visto, a proposito di TAGME, come sia concettualmente strutturato un meccanismo di votazione utilizzabile per la disambiguazione delle entità. Al termine di tale meccanismo ad ogni entità appartenente ad E_m viene associato un punteggio s_e, e l'entità col punteggio più alto risulta quella associata alla menzione m in quanto considerata la più rappresentativa della menzione tra tutte quelle candidate.

In aggiunta al metodo base, già utilizzato in TAGME, WAT aggiunge tre varianti che utilizzano per derivare un punteggio per l'entità, rispettivamente la *trigram-similarity* tra il titolo dell'entità e e la mention m, il numero n di voti positivi ricevuti dall'entità, o entrambi i dati, in aggiunta alla correlazione semantica tra e ed e'.

Nome	Tecnica	Algoritmo Base
base	Voti ricevuti	Voting scheme
base-t	Trigram-similarity	Voting scheme
base-n	Numero di voti positivi	Voting scheme
base-nt	Trigrammi e numero di voti	Voting scheme

Tabella 5.1: Metodi di disambiguazione basati sul voting scheme

5.1.2.2 Algoritmi graph-based

La seconda categoria di metodi di disambiguazione presente in WAT è basata su quello che prende il nome di *Mention-Entity graph*, un grafo nel quale i nodi corrispondenti alle

menzioni sono collegati ad un numero di entità accettabili, e nel quale il ruolo di tale collegamento è quello di indicare i possibili significati di una certa menzione. I nodi relativi alle entità sono tra loro connessi mediante archi pesati che indicano la similarità semantica tra le entità ai due estremi dell'arco. La costruzione del grafo è svolta nel modo seguente: l'insieme delle menzioni m è utilizzato per creare i nodi relativi alle menzioni, mentre le entità E_m ad esse collegate rilevate dallo spotter sono utilizzate per creare i nodi entità del grafo. Gli archi da una menzione ad un'entità possono essere pesati utilizzando una delle possibile funzioni di similarità tra menzioni: **identità**, il cui valore è sempre 1, **commonness**, oppure la **similarità basata sul contesto**, un punteggio di similarità calcolato tra il testo attorno alla menzione e ogni entità di Wikipedia (intesa come documento testuale) collegata alla menzione considerata. Una strategia di assegnazione dei punteggi simile a questa è applicata anche per la creazione degli archi pesati tra vertici di tipo entità, in questo caso i pesi degli archi sono calcolati utilizzando una delle misure di correlazione che verranno presentate nel dettaglio nei paragrafi successivi.

Nome	Similarità	Algoritmo Base
pr	identity	PageRank
ctx-pr	context	PageRank
comm-pr	commonness	PageRank
ppr	identity	Personalized PageRank
ctx-ppr	context	Personalized PageRank
comm-ppr	commonness	Personalized PageRank
ppr-uniform	identity	Personalized PageRank
ctx-ppr-uniform	context	Personalized PageRank
comm-ppr-uniform	commonness	Personalized PageRank
hits-auth	identity	HITS - Authority score
ctx-hits-auth	context	HITS - Authority score
comm-hits-auth	commonness	HITS - Authority score
hits-hub	identity	HITS - Hub score
ctx-hits-hub	context	HITS - Hub score
comm-hits-hub	commonness	HITS - Hub score
salsa-auth	identity	SALSA - Authority score
ctx-salsa-auth	context	SALSA - Authority score
comm-salsa-auth	commonness	SALSA - Authority score
salsa-hub	identity	SALSA - Hub score
ctx-salsa-hub	context	SALSA - Hub score
comm-salsa-hub	commonness	SALSA - Hub score

Tabella 5.2: Metodi di disambiguazione basati sul Mention-Entity graph

Una volta terminata la costruzione del grafo Mention-Entity, esso viene reso oggetto di una procedura di disambiguazione, avente come obiettivo finale quello di raffinare le possibili entità per ogni menzione, giungendo ad avere per ogni nodo di tipo menzione un solo arco che lo colleghi al vertice relativo all'entità individuata come miglior senso possibile per la menzione in oggetto. Il criterio utilizzato per individuare il miglior senso

possibile tra tutti quelli candidati è quello di applicare uno dei classici algoritmi di analisi dei grafi: PageRank, PageRank personalizzato, HITS e SALSA. Particolari precisazioni meritano gli ultimi due, per comprendere i quali è bene essere a conoscenza dei concetti di **hub** e **authority**. Una authority è un nodo che riveste importanza all'interno del grafo, caratteristica individuata in base al numero di nodi ad essa collegati; un hub è invece un nodo collegato ad un alto numero di authority. Il calcolo di tale classificazione è mutuamente ricorsivo, con l'**hub score** calcolato come somma dell'authority score dei nodi collegati ad un determinato nodo e l'**authority** score di un nodo calcolato come somma dell'hub score dei nodi ad esso collegati. La struttura duale di questi due metodi fa sì che siano due i punteggi di probabilità ottenuti con la loro applicazione, uno relativo alle autorità e l'altro relativo agli hub.

5.1.2.3 Ottimizzazione

In seguito alla fase di disambiguazione, esiste la possibilità in WAT di attivare il cosiddetto *ottimizzatore*, che esegue in sostanza una seconda fase di disambiguazione utilizzando il metodo basato sul voting scheme con uso di *trigram-similarity* tra il titolo dell'entità e e la mention m, e il numero n di voti positivi ricevuti dall'entità ma con una variazione: la procedura di voto viene effettuata considerando come votanti soltanto le entità associate alle annotazioni derivate dalla prima passata, invece che tutte le entità associate alla menzione.

5.1.3 Misure di correlazione

Per tentare di contrastare le possibili limitazioni della funzione di correlazione tra due entità implementata in TAGME, durante lo sviluppo di WAT si è cercato di introdurre un nuovo set di misure che fossero in grado si esprimere il grado di relazione tra due entità. Una funzione di correlazione assume valori compresi tra 0, che indica una assoluta mancanza di qualsivoglia correlazione, e 1, che sottende invece una forte correlazione tra le due entità in oggetto.

Tra le numerose misure di correlazione implementate in WAT, alcune delle quali saranno viste più approfonditamente nella sezione riguardante il refactoring eseguito sul sistema nell'ambito del lavoro di tesi, quattro sono state incluse nella pubblicazione del sistema, mentre le altre sono state messe da parte in quanto portatrici di prestazioni non particolarmente brillanti. Le quattro misure in questione sono:

1. *Jaccard*: ottenuta calcolando il rapporto tra l'intersezione e l'unione dei collegamenti in entrata delle due entità e_1 ed e_2: $\frac{|in(e_1) \cap in(e_2)|}{|in(e_1) \cup in(e_2)|}$;
2. *Milne-Witten*: definita come $1 - \frac{max(log|in(e_1)|, log|in(e_2)|) - log|in(e_1) \cap in(e_2)|}{|W| - min(log|in(e_1)|, log|in(e_2)|)}$;
3. *Cosine similarity* tra due vettori ottenuti applicando la tecnica LSI alle entità in oggetto;
4. Una misura empirica ottenuta dall'analisi delle distribuzioni delle intersezioni tra i link in entrata di due pagine, effettuata tramite una precomputazione di tali intersezioni per ogni coppia di entità, usata come base per il calcolo delle distribuzioni

basato sulle frequenze, le quali sono poi utilizzate per valutare la relatedness tra due entità.

5.1.4 Pruning

Come affermato in precedenza, l'obiettivo della fase di pruning è quella di ripulire il risultato dell'annotazione di tutte le annotazioni prodotte ritenute poco pertinenti, migliorando così la *precision* del sistema. In aggiunta al meccanismo di pruning basato sul valore soglia, ereditato da TAGME, WAT mette a disposizione anche la possibilità di utilizzare nel pruner un classificatore binario, allenato su una serie di dati relativi alla mention e all'entità col punteggio più alto, che classifica ogni annotazione prodotta nelle fasi precedenti come pertinente o non pertinente.

5.2 Prestazioni note di WAT

La fase di testing effettuata dal team di sviluppo di WAT è stata suddivisa in tre diversi esperimenti, allo scopo di verificare separatamente le varie caratteristiche del sistema. Tutti i risultati riportati fanno riferimento ai risultati ottenuti da WAT sul dataset ERD, sebbene test congiunti siano stati condotti su altri dataset quali AIDA/CoNNL, AQUAINT, IITB e MSNBC, ottenendo risultati comparabili a quelli che saranno di seguito descritti. Il primo esperimento, chiamato *spotting coverage*, prevedeva di effettuare la fase di spotting sui documenti passati in input, e di confrontare poi il set di menzioni presenti nel golden standard con quelle individuate nella fase di parsing appena svolta. Il secondo esperimento era invece istanza del problema D2W, con le menzioni presenti nel golden standard trasmesse in input e il set di entità disambiguate restituito come risultato e in seguito confrontato con le entità collegate alle menzioni nel golden standard. L'ultimo esperimento eseguito ha visto invece l'utilizzo dell'annotatore completo, con un tipico problema di annotazione con il testo come input e un set di annotazioni come risultato finale.

In seguito all'esecuzione dei tre esperimenti le misure di F1, precision e recall sono state calcolate, sia utilizzando nel processo di confronto con il golden standard il matching di tipo weak che quello di tipo strong.

Primo esperimento: Spotting

È facile fornire una descrizione di questo tipo di esperimento: il documento viene fornito in input, la parte dell'annotatore che si occupa del parsing viene eseguita sul testo e l'elenco delle menzioni individuate viene confrontato con l'elenco delle menzioni contenute nel golden standard. In questo caso sono considerate *true positive* le annotazioni presenti sia nel golden standard che nelle menzioni individuate dall'annotatore, *false positive* le menzioni che sono presenti nell'elenco prodotto dallo spotter ma non nel golden standard, sono invece definiti i *false negative* come le menzioni presenti nel golden standard ma non identificate nella fase di parsing. Partendo dalla definizione di questi concetti base è facile derivare le classiche misure di F1, precision e recall utilizzando per la loro computazione le formule viste in precedenza.

Per quanto riguarda il settaggio delle opzioni dello spotter è possibile seguire tre vie, nella prima si concentrano le attenzioni sulla massimizzazione del recall, affidandosi al pruner per la rimozione dei risultati spuri, un'altra opzione è quella di massimizzare invece la precisione dello spotter, correndo il rischio di tralasciare menzioni generando quindi falsi negativi, l'ultima opzione è quella di massimizzare l'F1, sebbene questa scelta implichi la sicurezza di avere a disposizione un efficiente algoritmo di disambiguazione in combinazione con un pruner relativamente semplice. Per questo motivo è consigliabile migliorare il recall, a meno che lo spotter non sia in grado di produrre una F1 vicina al valore massimo.

Nella tabella 5.3 sono riportati i valori ottenuti da WAT nell'esperimento eseguito, confrontati con quelli ottenuti dallo Stanford NER parser, considerato lo stato dell'arte nell'ambito della ricerca delle entità di un testo, e utilizzato come spotter da numerosi sistemi di annotazione.

Spotter	Soglia	P	R	F1
WAT	0.266	0.467	0.588	0.493
WAT	0.000	0.076	0.919	0.136
Stanford	-	0.296	0.381	0.312

Tabella 5.3: Spotting sul dataset ERD con strong match e soglia su $lp(m)$

Sebbene in termini assoluti di misurazioni, la soluzione implementata dall'Università di Stanford abbia offerto prestazioni di livello elevato, lo spotter utilizzato da WAT si è rivelato più efficiente in termini di rapporto tra tempo di esecuzione e qualità dell'output fornito, oltre ad aver dimostrato maggiore flessibilità e capacità di adattamento a dataset contenenti documenti di differente tipologia.

Secondo esperimento: Disambiguazione

Per questo esperimento ogni algoritmo di disambiguazione implementato in WAT è stato testato singolarmente, utilizzandolo per la risoluzione di un problema di tipo Disambiguate to Wikipedia. L'esperimento è stato organizzato nel modo seguente: l'input fornito all'annotatore era costituito dall'insieme delle menzioni contenute nel golden standard, a tali menzioni è stata applicata la procedura di disambiguazione, lasciando al disambiguatore la possibilità di scegliere se usare o meno altre menzioni o caratteristiche lessicali del testo originale.

L'analisi degli algoritmi di disambiguazione ha rivelato particolari ignorati sino a quel momento: essi offrivano infatti prestazioni paragonabili tra di loro, lasciando quindi intuire che il problema della fase di annotazione non fosse in realtà la fase di disambiguazione in se quando più la presenza tra le menzioni da disambiguare di falsi positivi, introdotti quindi erroneamente durante la fase di spotting. Allo scopo di provare questa intuizione è stata predisposta una seconda fase nell'esperimento, nella quale è stato trasmesso al disambiguatore, oltre all'elenco delle menzioni contenute nel golden standard, un insieme di altre menzioni individuate dallo spotter, le quali pur non potendo far parte del risultato finale, concorrevano nella disambiguazione delle restanti menzioni influenzando quindi il

risultato della procedura. Come previsto è risultato che tutti gli algoritmi risentissero della presenza del rumore introdotto sui dati, sebbene delle due tipologie di disambiguazione, quella basata sul grafo Mention-Entity risultasse più robusta all'introduzione di valori spuri nell'input.

I risultati dell'esperimento hanno dimostrato da un lato la effettiva robustezza e resistenza al rumore degli algoritmi di disambiguazione implementati in WAT, dall'altro la necessità di utilizzare uno spotter più preciso allo scopo di limitare l'introduzione di falsi positivi, costituenti una complicazione sia per la fase di disambiguazione che per la successiva fase di pruning, affrontata a partire da un insieme di annotazioni ricco di risultati non pertinenti.

Terzo esperimento: Sa2W to D2W

Il terzo esperimento effettuato nella fase di testing preliminare di WAT è stata la cosiddetta riduzione da Sa2W a D2W, la quale vede un testo completamente annotato dall'annotatore senza utilizzo di alcune informazioni ausiliarie e nel quale esclusivamente le menzioni aventi match (weak) con quelle del golden standard sono utili ai fini della valutazione finale del risultato.

I risultati di quest'ultimo test evidenziano prestazioni e informazioni in linea con quelle dei precedenti due esperimenti, mostrando un'innalzamento nell'intervallo tra l'1 e il 9% delle prestazioni rispetto a quelle di TAGME.

5.3 Considerazioni sull'analisi delle prestazioni

Dall'analisi effettuate e appena presentate, è facile evincere come il punto critico del nuovo sistema, nonostante le prestazioni siano state notevolmente migliorate rispetto a quello che era considerato lo stato dell'arte nell'annotazione, fossero le fasi preliminari e quella finale. Infatti nonostante nella comunità degli annotatori testuali gran parte degli studi siano concentrati sullo sviluppo di nuove ed innovative tecniche di disambiguazione che consentano una sempre crescente precisione nell'assegnazione della corretta entità ad ogni menzione, è evidente che la necessità reale sarebbe quella di implementare una fase di spotting in grado di fornire allo step successivo risultati di maggiore qualità. Tale necessità è resa evidente dall'uniformità della resa dei vari esperimenti verificata dal secondo degli esperimenti effettuati, e in particolare dalla flessione delle performance degli algoritmi di disambiguazione all'atto dell'introduzione nell'output di menzioni non pertinenti. Ciò ha reso evidente che il reale problema dell'annotazione non risiede attualmente nella capacità di assegnare ad ogni menzione la più pertinente entità, compito assolto egregiamente dalle tecnologie già esistenti, ma nella capacità della fase preliminare di individuare le porzioni di testo realmente interessanti, permettendo così alla fase di disambiguazione di operare soltanto su risultati pertinenti e limitando la necessità di un'azione di pruning intensa a causa della grande percentuale di falsi positivi introdotti dallo spotter.

È su questi importanti concetti che si è deciso di basare la fase di refactoring oggetto di questa tesi, volta a minimizzare la presenza di falsi positivi nel risultato finale, garantendo comunque una copertura ottimale delle menzioni presenti nel testo, introducendo

nuove tecniche di spotting e disambiguazione che, con il supporto di nuove strategie, permettessero di migliorare ulteriormente le prestazioni di un sistema già affermatosi come nuovo stato dell'arte.

Capitolo 6

Il nuovo WAT

6.1 Prestazioni degli annotatori

Prima di entrare nel dettaglio delle analisi effettuate su WAT allo scopo di individuarne debolezze e trovare ad esse eventuale rimedio, verrà effettuata una rapida descrizione delle prestazioni di partenza del sistema da utilizzare come baseline per i successivi confronti. Saranno inoltre analizzate in termini di metriche le prestazioni degli annotatori presenti su GERBIL per poter effettuare una comparazione tra essi e WAT in seguito alle modifiche apportate.

6.1.1 Dataset

Tutti gli esperimenti presentati sia in questo capitolo che nei successivi sono stati eseguiti su otto dataset, tutti inclusi in GERBIL: AIDA/CoNLL-Complete (d'ora in avanti abbreviato in AIDA o AIDA/CoNLL), MSNBC, IITB, AQUAINT, DBpedia Spotlight, N3-Reuters-128, N3-RSS-500 e KORE50. Tali dataset sono particolarmente eterogenei sia per tipo di documenti presenti che per la natura delle annotazioni contenute nel golden standard. Come è possibile notare dal riassunto riportato nella tabella 6.1, risulta molto variabile oltre alla lunghezza media dei documenti, anche il numero medio di annotazioni in essi presenti, con dataset riccamente annotati, come nel caso di IITB, o più parchi di annotazioni, quali AIDA. Tra quelli presenti in GERBIL sono stati esclusi i dataset Microposts in quanto non redistribuibile e quindi non incluso nell'installazione del sistema, ACE2004 il cui golden standard è vuoto per il 38% dei documenti e quindi ritenuto poco indicativo e infine Meij il quale non offre un golden standard per il tipo di esperimenti svolto. È facile intuire la difficoltà insita nel compito di individuare una procedura di annotazione che sia in grado di offrire prestazioni costanti in presenza di una varietà così ricca di tipologie di documenti e annotazioni.

6.1.2 Analisi delle prestazioni di WAT

L'analisi preliminare delle prestazioni del sistema ha avuto come prima fase la ricerca del più efficiente metodo di disambiguazione e della miglior misura di correlazione, in modo da

Dataset	# Docs	# Anns	AVG |Doc|	AVG |Ann|	AVG Anns/Doc
AIDA/CoNLL	1393	27815	1130,12	8,996	19,97
AQUAINT	50	727	1415,98	11,55	14,54
DBpedia Spotlight	58	330	173,34	8,56	5,69
IITB	103	11242	3879,37	8,81	109,15
KORE 50	50	143	74,6	6,31	2,86
MSNBC	20	650	3316,1	10,16	32,5
N3 Reuters-128	128	621	752,37	12,15	4,85
N3 RSS-500	500	495	170,93	11,15	0,99

Tabella 6.1: Statistiche sui dataset coinvolti nell'esperimento. Di ogni dataset sono riportati il nome, il numero di documenti, il numero di annotazioni nel golden standard, la lunghezza media di un documento, la lunghezza media in caratteri di un'annotazione e il numero medio di annotazioni per documento.

individuare la migliore combinazione possibile da utilizzare come baseline per i successivi raffronti.

Gli esperimenti condotti hanno mostrato, in accordo con quanto appurato dagli sviluppatori di WAT in fase di pubblicazione, che il più efficiente dei metodi di disambiguazione è quello base, basato sul voting scheme inizialmente introdotto in TAGME. Tale metodo sarà pertanto quello utilizzato in tutti gli esperimenti illustrati nei prossimi capitoli.

Anche per quanto riguarda la misura di relatedness in grado di garantire la maggiore efficacia gli esperimenti condotti sono giunti alle medesime conclusioni precedentemente raggiunte dagli sviluppatori del sistema, individuando nella misura di Jaccard la più efficiente tra quelle disponibili. Essa rimarrà pertanto la misura di default negli esperimenti condotti.

6.1.3 Valutazione iniziale

Al fine di stabilire una base di confronto, è stato eseguito con GERBIL un esperimento di tipo Sa2KB, valutato con Strong Match Annotation, sugli otto dataset precedentemente illustrati, utilizzando WAT e gli altri annotatori disponibili out of the box in GERBIL. I risultati di tali esperimenti sono riportati nella tabella 6.2, la quale, oltre ai risultati di WAT (nella sua versione iniziale), contiene, per ogni dataset, i risultati dei tre migliori annotatori disponibili. Come è possibile notare dai dati presentati, WAT deteneva già in partenza il primato per quanto riguarda i dataset AIDA, MSNBC, N3-Reuters-128 e N3-RSS-500, e il suo punteggio era tra i migliori tre per i dataset AQUAINT e KORE50. Per poter meglio individuare i punti critici di ognuno degli stage della pipeline di annotazione, è stata messa a punto un'analisi approfondita che tenesse in considerazione singolarmente ognuno degli aspetti chiave del sistema. Le due fasi chiave dalle quali dipende gran parte delle prestazioni di un sistema di annotazione come WAT sono la fase di spotting e quella di disambiguazione, ed è su queste che si è concentrata l'analisi preliminare in oggetto.

Dataset	Annotatore	F1	Precision	Recall
AIDA/CoNLL	WAT	**0.662**	0.733	0.605
	TAGME	0.562	0.553	0.571
	Wikipedia Miner	0.470	0.403	0.565
	NERD-ML	0.446	0.489	0.410
AQUAINT	TAGME	**0.458**	0.412	0.514
	Wikipedia Miner	0.400	0.296	0.616
	WAT	0.351	0.364	0.340
	DBpedia Spotlight	0.235	0.172	0.367
DBpediaSpotlight	Wikipedia Miner	**0.512**	0.551	0.479
	TAGME	0.466	0.397	0.564
	DBpedia Spotlight	0.461	0.428	0.500
	WAT	0.228	0.622	0.139
IITB	NERD-ML	**0.477**	0.482	0.472
	Wikipedia Miner	0.463	0.452	0.474
	DBpedia Spotlight	0.420	0.411	0.430
	WAT	0.211	0.458	0.137
KORE50	Babelfy	**0.552**	0.552	0.552
	WAT	0.493	0.519	0.469
	TAGME	0.459	0.404	0.532
	Wikipedia Miner	0.317	0.289	0.350
MSNBC	WAT	**0.627**	0.717	0.557
	TAGME	0.465	0.428	0.509
	NERD-ML	0.404	0.444	0.371
	Babelfy	0.398	0.327	0.511
N3-Reuters-128	WAT	**0.490**	0.558	0.436
	TAGME	0.323	0.368	0.288
	NERD-ML	0.294	0.346	0.256
	Wikipedia Miner	0.224	0.229	0.219
N3-RSS-500	WAT	**0.410**	0.385	0.438
	TAGME	0.378	0.370	0.387
	Babelfy	0.278	0.227	0.359
	Wikipedia Miner	0.246	0.307	0.206

Tabella 6.2: Prestazioni ottenute nell'esperimento Sa2KB con strong annotation match dagli annotatori sui dataset considerati. La versione di WAT utilizzata è quella originaria, senza alcuna delle modifiche apportate nel corso del lavoro di tesi.

6.2 La fase di spotting

Come visto in precedenza lo spotter è quella componente dell'annotatore che esegue sul testo la fase preliminare di individuazione delle menzioni che dovranno essere successivamente annotate. La qualità della fase di spotting pone chiaramente un upper bound sulla qualità delle successive fasi dell'annotazione: è infatti evidente che, pur avendo a disposizione l'algoritmo di disambiguazione ideale, sarebbe comunque impossibile massimizzare le prestazioni utilizzando uno spotter non efficiente; menzioni che risultano essere pertinenti potrebbero infatti non essere state individuate dalla fase di spotting e pertanto non verrebbero annotate limitando la frazione di annotazioni pertinenti correttamente effettuate.

La prima fase dell'analisi del sistema è quindi coincisa con l'organizzazione di un esperimento volto a mettere in chiaro la qualità della procedura di spotting presente in WAT, confrontandola con possibili alternative delle quali si esamineranno a fondo le prestazioni nel contesto generale dell'annotazione nella prossima sezione. Il problema risolto in questa fase è così formulabile: dato in input un testo, esso viene sottoposto a procedura di spotting e l'annotatore restituisce una lista di menzioni (indici che individuano una porzione di testo che dovrebbe in seguito essere annotata), la quale viene confrontata con la lista delle menzioni contenute nel golden standard, ottenuta considerando soltanto le posizioni delle annotazioni in esso contenute e non le entità collegate. Tale confronto può essere effettuato utilizzando le metriche di valutazione classiche e utilizzando per effettuare il confronto tra due menzioni lo Strong Mention Match, che considera due menzioni come equivalenti quando condividono sia l'indice di partenza all'interno del testo che la lunghezza. In seguito a tale procedura di matching è possibile definire le classiche misure di precision, recall ed F1 considerando come *true positive* le menzioni trovate dallo spotter e presenti nel golden standard, *false positive* le menzioni trovate dallo spotter ma senza una corrispondenza nel golden standard, e come *false negative* le menzioni non individuate dallo spotter sebbene siano presenti nel golden standard (si veda la sezione 2.1.3 per una più precisa definizione). L'esperimento appena descritto è stato eseguito utilizzando lo spotter di WAT sulle istanze degli otto dataset presentati in precedenza, integrandolo via via con possibili componenti alternative mirate ad aumentare la copertura dello spotting sul testo, e considerate qui esclusivamente in relazione alle loro prestazioni nell'ambito dell'individuazione delle corrette menzioni, lasciando ad una fase successiva il loro test in condizioni normali di annotazione.

Lo spotter di WAT è realizzato in modo da sfruttare a proprio vantaggio le informazioni provenienti dello step di parsing e tokenizzazione effettuato sul testo prima di iniziare l'annotazione vera e propria. Diversi tokenizer pubblicamente disponibili offrono infatti oltre alle classiche informazioni su inizio e fine di un token, anche altre informazioni, denominate *part-of-speech label (POS)* e *named entity label (NE)*. Le POS label offrono informazioni di tipo grammaticale sui token, indicando cosa rappresentino all'interno della frase (nomi propri, nomi comuni, aggettivi, verbi), mentre le NE label individuano, per quelle classificate come entità, la classe che meglio le descrive tra un insieme ristretto di classi possibili. Sulla base di queste informazioni, lo spotter di WAT stabilisce dove andare a cercare le mention da annotare concentrando la ricerca su quelle che sono state identificate come entità. Risulta quindi evidente che gran parte del successo dell'attività

di spotting derivi dalla qualità delle informazioni elaborate in fase di tokenizzazione. È quindi chiave il ruolo del tokenizer scelto. Sono in totale tre i tokenizer messi a confronto in questa fase:

1. **OpenNLP**: il tokenizer presente nella versione di partenza di WAT, offerto dalla suite OpenNLP di Apache;
2. **Stanford Parser**: considerato lo stato dell'arte dell'analisi testuale e sulla carta foriero delle migliori prestazioni;
3. **Factorie**: un toolkit per la creazione di modelli probabilistici nell'ambito dell'analisi di testi in grado di offrire prestazioni migliori in termini di tempo rispetto a quelle dello Stanford Parser;

Per quanto riguarda lo Stanford Parser e Factorie, sono state testate per ciascuno di essi differenti configurazioni, che saranno via via descritte, allo scopo di aumentare la percentuale di testo correttamente individuato dallo spotter e massimizzare il recall dell'esperimento.

OpenNLP

L'esperimento condotto su questo tokenizer ha visto l'utilizzo di una configurazione identica a quella usata in partenza su WAT, in modo da poter contemporaneamente valutarne le prestazioni e costituire una baseline sulla quale definire una classifica delle configurazioni possibili. Le metriche ottenute dallo svolgimento dell'esperimento utilizzando lo spotter basato su OpenNLP sono riportate in tabella 6.3.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.715	0.705	0.726
AQUAINT	0.334	0.293	0.388
DBpedia Spotlight	0.267	0.730	0.164
IITB	0.260	0.543	0.171
KORE 50	0.872	0.885	0.860
MSNBC	0.675	0.651	0.700
N3 Reuters-128	0.569	0.491	0.676
N3 RSS-500	0.435	0.289	0.877

Tabella 6.3: Spotting con OpenNLP

Stanford Parser

Nella sperimentazione di questo tokenizer sono state realizzate diverse configurazioni, avvalendosi delle caratteristiche insite nel sistema utilizzato. Lo Stanford Parser produce infatti per i token individuati sia le *NE label*, che le *POS label*. Nella normale procedura di spotting, WAT utilizzerebbe come suggestion esclusivamente le porzioni di testo etichettate dal tokenizer come *Named entity*, ed è questa la prima delle configurazioni testate.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.802	0.685	0.967
AQUAINT	0.318	0.264	0.400
DBpedia Spotlight	0.234	0.510	0.152
IITB	0.227	0.434	0.153
KORE 50	0.806	0.846	0.769
MSNBC	0.619	0.545	0.715
N3 Reuters-128	0.508	0.395	0.712
N3 RSS-500	0.395	0.260	0.826

Tabella 6.4: Spotting con Stanford Parser

Poiché è stato possibile notare che spesso la procedura di tokenizzazione non classifica come entità porzioni del documento che sarebbe interessante annotare, si è proceduto con l'aggiunta alla lista delle suggestions, dei token aventi POS label relativa a nomi propri.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.781	0.652	0.974
AQUAINT	0.322	0.255	0.435
DBpedia Spotlight	0.248	0.487	0.167
IITB	0.254	0.431	0.180
KORE 50	0.858	0.830	0.888
MSNBC	0.620	0.521	0.765
N3 Reuters-128	0.503	0.378	0.750
N3 RSS-500	0.387	0.249	0.874

Tabella 6.5: Spotting con Stanford Parser (con nomi propri)

Factorie

Sulla scia dell'organizzazione seguita nel testing dello Stanford Parser, le prime due configurazioni sono state utilizzate anche nel caso di Factorie. I risultati ottenuti nella configurazione base, nella quale solo le entità possono essere menzioni, sono riportati in tabella 6.6. La seconda configurazione è stata anche in questa occasione quella ottenuta con l'aggiunta alle suggestion dei token etichettati come nomi propri. In aggiunta agli esperimenti già effettuati, utilizzando Factorie si è proceduto con due ulteriori step di configurazione. Il primo ha visto l'aggiunta alle suggestion oltre che delle entità e dei nomi propri anche dei token etichettati come nomi comuni, mentre nel secondo sono stati suggeriti anche i token relativi agli aggettivi. La scelta di testare l'aggiunta di nomi comuni e propri soltanto in relazione all'utilizzo di Factorie ma non dello Stanford Parser è giustificata dall'analisi dei risultati ottenuti dai due sistemi negli esperimenti precedentemente svolti, che ha sancito una sostanziale equivalenza prestazionale, lasciando però a vantaggio di

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.899	0.862	0.939
AQUAINT	0.349	0.318	0.388
DBpedia Spotlight	0.214	0.597	0.130
IITB	0.243	0.561	0.155
KORE 50	0.771	0.906	0.671
MSNBC	0.761	0.733	0.792
N3 Reuters-128	0.624	0.577	0.680
N3 RSS-500	0.421	0.296	0.725

Tabella 6.6: Spotting con Factorie

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.807	0.697	0.959
AQUAINT	0.322	0.259	0.424
DBpedia Spotlight	0.262	0.577	0.170
IITB	0.283	0.530	0.193
KORE 50	0.906	0.903	0.909
MSNBC	0.719	0.626	0.843
N3 Reuters-128	0.576	0.475	0.731
N3 RSS-500	0.388	0.250	0.862

Tabella 6.7: Spotting con Factorie (con nomi propri)

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.483	0.323	0.960
AQUAINT	0.222	0.134	0.640
DBpedia Spotlight	0.550	0.467	0.670
IITB	0.485	0.396	0.624
KORE 50	0.726	0.590	0.944
MSNBC	0.301	0.182	0.855
N3 Reuters-128	0.183	0.105	0.731
N3 RSS-500	0.185	0.104	0.864

Tabella 6.8: Spotting con Factorie (con nomi propri e comuni)

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.444	0.288	0.961
AQUAINT	0.215	0.127	0.682
DBpedia Spotlight	0.527	0.412	0.733
IITB	0.481	0.369	0.690
KORE 50	0.694	0.549	0.944
MSNBC	0.269	0.159	0.855
N3 Reuters-128	0.161	0.090	0.733
N3 RSS-500	0.167	0.092	0.864

Tabella 6.9: Spotting con Factorie (con nomi propri, nomi comuni e aggettivi)

Factorie la miglior performance a livello di tempo di esecuzione, portando pertanto le analisi successive a concentrarsi esclusivamente su quest'ultimo sistema.

È evidente come l'eterogeneità dei dataset giochi un ruolo chiave nell'influenzare le prestazioni dello spotter, se infatti l'aggiunta di nomi comuni e aggettivi costituisce per alcuni casi un semplice aumento del rumore e non offre sostanziali miglioramenti in termini di copertura del testo, per altri, come ad esempio IITB, è l'unica procedura tra quelle testate in grado di innalzare il valore del recall. Tale fenomeno è conseguenza diretta delle scelte effettuate in fase di realizzazione dei dataset, che vedono in alcuni di essi la presenza di annotazioni golden in esclusiva concomitanza con le *named entity* contenute nel testo, e in altri la presenza di annotazioni su qualunque parte del discorso, e, nel caso particolare di dataset quali IITB e AQUAINT, in particolar modo su aggettivi e nomi comuni, i quali difficilmente vengono marcati come entità dai tokenizer.

La chiave di lettura dei risultati appena riportati, riassunti in Tabella 6.10, è quella di stabilire quanto, in presenza di un algoritmo di disambiguazione in grado di collegare ogni singola menzione individuata all'entità corretta, sia possibile ottenere dal sistema di annotazione utilizzando un determinato metodo di spotting. Risulta chiaro pertanto che, ottenuto in fase di spotting un determinato valore di recall, non ci si può aspettare in fase di annotazione di migliorare tale valore, dal momento che è evidente che l'annotazione trascurerà completamente determinate porzioni di testo in quanto non segnalate dallo spotter.

In complementarità con l'esperimento appena presentato, è utile ai fini di classificare correttamente le potenzialità e la perfettibilità di un annotatore effettuare un secondo tipo di test ad esso strettamente correlato. Il problema da risolvere è così formulato: dato all'annotatore un testo, esso lo sottopone alla procedura di spotting e restituisce questa volta oltre alla lista di menzioni, la lista di entità candidate ad essere senso di ognuna delle menzioni spottate. La valutazione dei risultati consiste nel verificare, per ogni menzione ritenuta true positive al passo precedente, se tra le entità candidate sia presente quella corretta indicata dal golden standard.

Tale dato permette di calcolare delle metriche ipotetiche su quali sarebbero i risultati dell'esperimento di annotazione se in fase di disambiguazione ogni menzione true positive venisse annotata con la giusta entità, tenendo conto anche della presenza di falsi positivi (le menzioni annotate ma senza alcuna corrispondenza nel golden standard) ed errori

Dataset	Tokenizer	F1	Precision	Recall
AIDA/CoNLL	Factorie	**0.899**	**0.862**	0.939
	Stanford (NP)	0.781	0.652	**0.974**
AQUAINT	Factorie	**0.349**	**0.318**	0.388
	Factorie (NP, NN, JJ)	0.215	0.127	**0.682**
DBpediaSpotlight	Factorie (NP, NN)	**0.550**	0.467	0.670
	OpenNLP	0.267	**0.730**	0.164
	Factorie (NP, NN, JJ)	0.527	0.412	**0.733**
IITB	Factorie (NP, NN)	**0.485**	0.396	0.624
	OpenNLP	0.260	**0.543**	0.171
	Factorie (NP, NN, JJ)	0.481	0.369	**0.690**
KORE50	Factorie (NP)	**0.906**	0.903	0.909
	Factorie	0.771	**0.906**	0.671
	Factorie (NP, NN)	0.726	0.590	**0.944**
MSNBC	Factorie	**0.761**	**0.733**	0.792
	Factorie (NP, NN)	0.301	0.182	**0.855**
N3-Reuters-128	Factorie	**0.624**	**0.577**	0.680
	Stanford (NP)	0.503	0.378	**0.750**
N3-RSS-500	OpenNLP	**0.435**	**0.289**	**0.877**

Tabella 6.10: Riepilogo delle prestazioni dello spotter. Per ogni dataset sono riportati gli spotter che ottengono il miglior risultato in almeno una delle metriche F1, Precision e Recall, indicati in grassetto. (NP = nomi propri, NN = nomi comuni, JJ = aggettivi)

Dataset	Missing	Excess	Correct	Disambiguabili	Non Disambiguabili
AIDA/CoNLL	1095	65900	26720	26339	381
AQUAINT	231	3398	496	483	13
DBpedia Spotlight	88	346	242	238	4
IITB	3463	13190	7719	6592	1127
KORE 50	8	111	135	133	2
MSNBC	94	2933	556	533	23
N3 Reuters-128	166	4588	455	402	53
N3 RSS-500	67	4199	427	403	24

Tabella 6.11: Statistiche sullo spotting con Factorie (con nomi propri, nomi comuni e aggettivi), configurazione che massimizza il recall sulle menzioni individuate. Missing, Excess e Correct fanno riferimento alle menzioni individuate dallo spotter.

(le menzioni per le quali non è presente tra i candidati il senso corretto). Avvalendosi della struttura di classi di GERBIL, si è svolto l'esperimento appena presentato, e sono state calcolate le metriche ipotetiche creando un risultato di annotazione fittizio avente come annotazioni quelle corrispondenti alle menzioni individuate dallo spotter, annotate, qualora fosse possibile, con l'entità presente nel golden standard. In aggiunta alle metriche standard è stato calcolato anche il numero in termini assoluti di menzioni correttamente spottate per le quali non sarebbe possibile individuare il senso corretto a causa di una mancanza di tale entità nel sistema. I risultati sono stati calcolati per ognuna delle possibili configurazioni utilizzate durante l'esperimento di spotting, e quelli relativi alla strategia di spotting rivelatasi più efficiente in termini di recall sono riassunti nella tabella 6.11. Il senso di tali risultati è, come nel caso precedente, quello di stabilire quali risultati sarebbe possibile ottenere utilizzando il perfetto disambiguatore, avendo alla base uno degli spotter presenti in WAT. In aggiunta però a questa interpretazione dei risultati, viene in questo caso introdotto un ulteriore concetto, quello di capacità di annotazione, che è necessario a comprendere quanti degli eventuali problemi di disambiguazione siano relativi ad una mancanza dei sensi corretti all'interno del sistema di annotazione. I dati sulle menzioni prodotte dallo spotting mostrano come, anche nel caso di un algoritmo di spotting rivelatosi il più pervasivo tra quelli testati, come visibile nell'apposita tabella, il numero delle menzioni impossibili da spottare per alcuni dataset è particolarmente alto, come alto è il numero di annotazioni in eccesso le quali contribuiscono al deterioramento delle prestazioni generali.

Per completezza verranno riportati anche i valori ottenuti utilizzando la strategia di spotting ritenuta migliore come performance complessiva, in quanto meno falsati dall'incremento degli excess e quindi maggiormente utilizzabili come riferimento nelle fasi successive di analisi sul sistema reale di disambiguazione. È bene tenere conto del fatto

Dataset	F1	Prec	Rec	Missing	Correct	Disamb	Non Disamb
AIDA/CoNLL	0.886	0.850	0.926	1702	26113	25753	360
AQUAINT	0.346	0.314	0.384	445	282	279	3
DBpedia Spotlight	0.214	0.597	0.130	287	43	43	0
IITB	0.229	0.530	0.146	9451	1731	1635	96
KORE 50	0.763	0.896	0.664	47	96	95	1
MSNBC	0.733	0.706	0.763	135	515	496	19
N3 Reuters-128	0.562	0.520	0.612	199	422	380	42
N3 RSS-500	0.397	0.280	0.684	136	358	338	20

Tabella 6.12: Statistiche sullo spotting con Factorie (versione semplice, come da tabella 6.6), metriche F1, precision e recall ipotetiche, ossia calcolate assumendo che uno spot individuato correttamente porti sicuramente alla scelta della entità corretta. Missing, Excess e Correct fanno riferimento alle menzioni individuate dallo spotter, F1, Precision e Recall sono invece calcolati in relazione alle entità. Disamb = entità corretta presente tra quelle possibili per la menzione, Non disamb = tra le entità collegate alla menzione non esiste quella corretta.

che i dati sui quali fare maggiormente affidamento sono quelli relativi alle menzioni dal

momento che, sebbene generalmente in linea con quelle ottenuti in esperimenti reali che saranno mostrati nelle prossime sezioni, le metriche proiettate rischiano di risultare falsate da molti fattori, dal momento che sono calcolate sulla base dell'ipotesi poco realistica che tutte le menzioni individuate vengano poi annotate con una soglia di pertinenza sufficientemente alta da far parte del risultato finale, e che quindi quelle per le quali esiste il giusto senso siano annotate con tale entità mentre tutte quelle per le quali il senso non esiste siano annotate con un'entità sbagliata. Al contrario i dati relativi alle menzioni spottate riportati in tabella mostrano quanto bassa sia la percentuale di menzioni correttamente spottate ma che non possono essere disambiguate correttamente perché nel sistema non è presente il senso corretto, sottolineando ancora una volta l'importanza di un robusto algoritmo di spotting, come dimostrato dal numero di menzioni *missing*, molto spesso nettamente superiore a quelle correct per le quali non è però presente un senso ammissibile. Discorso a parte merita il caso particolare del dataset IITB, che, come visibile nella tabella 6.12, ottiene punteggi in controtendenza rispetto agli altri dataset. È infatti possibile notare come il numero di menzioni che lo spotter non è stato in grado di individuare è molto alto rispetto all'analogo valore negli altri casi. Tale discrepanza è dovuta con tutta probabilità alla particolare struttura del golden standard di IITB, le cui annotazioni spesso coincidono con parti del testo difficili da classificare come entità (aggettivi, nomi comuni) e che pertanto lo spotter non è in grado di individuare, facendo sì che su questo particolare dataset, nessun algoritmo di disambiguazione sarebbe in grado, senza una differente strategia di spotting, di eseguire correttamente l'annotazione.

6.3 La fase di disambiguazione

Il secondo aspetto preso in considerazione nell'analisi è quello della disambiguazione, riconducibile al problema generalmente indicato come D2W o D2KB. Il problema consiste nel trasmettere all'annotatore un testo insieme alla lista delle menzioni relative alle annotazioni presenti nel golden standard, le quali dovranno essere sottoposte a procedura di disambiguazione e annotate quindi con l'entità più pertinente. Se l'esperimento precedente aveva la capacità di individuare come il sistema si potrebbe comportare abbinando ad uno spotter imperfetto un algoritmo di disambiguazione perfetto, questo esperimento indaga il problema complementare: quali prestazioni l'annotatore sarebbe in grado di garantire se lo spotter fosse perfetto. Per condurre questo test è stato utilizzato lo strumento per la risoluzione del problema D2KB presente in GERBIL, usando WAT nella sua versione base per eseguire la disambiguazione, utilizzando l'algoritmo con voting scheme ereditato da TAGME, precedentemente individuato come il più performante tra gli algoritmi di disambiguazione inclusi in WAT. I risultati ottenuti sono riportati nella tabella 6.13. Come è possibile notare, l'insieme dei dataset che maggiormente hanno messo in difficoltà lo spotter durante il precedente esperimento si interseca in gran parte con quello dei dataset nei quali la disambiguazione ha prestazioni meno soddisfacenti. Da tale intuizione è possibile trarre la conclusione che alcuni tra i dataset considerati risultano a priori difficili da annotare, sia per quanto riguarda il problema dell'individuare le corrette menzioni, che per la fase di disambiguazione, risultata poco efficiente anche in caso di spotting perfetto. Quanto detto va a rafforzare ulteriormente la più volte rimarcata centralità delle

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.812	0.846	0.780
AQUAINT	0.740	0.819	0.674
DBpediaSpotlight	0.636	0.689	0.591
IITB	0.592	0.658	0.537
KORE50	0.552	0.566	0.539
MSNBC	0.756	0.819	0.702
N3-Reuters-128	0.645	0.755	0.562
N3-RSS-500	0.690	0.713	0.668

Tabella 6.13: Risultati dell'esperimento D2KB

differenze strutturali tra i dataset. Come è possibile dedurre dalla panoramica sui dataset riportata nella sezione 2.2.2.3 e dai dati statistici in tabella 6.1, non esiste omogeneità strutturale tra i dataset, essi infatti differiscono non solo per la lunghezza dei loro documenti, ma anche e soprattutto per le scelte operate in fase di annotazione, esse sono state infatti condotte indipendentemente le une dalle altre, senza adoperare una linea di azione comune che portasse ad ottenere dataset comparabili tra loro. Oltre al dato evidente del numero di annotazioni, chiaramente visibile dal numero medio di annotazioni per documento riportato in tabella, è mutevole anche la relazione tra menzione ed entità linkata. Per alcuni dei dataset infatti, si è scelto di annotare esclusivamente le menzioni ritenute importanti, ed esse sono generalmente coincidenti con le named entity rilevate dai più comuni tokenizer. Altri dataset, quale ad esempio IITB, hanno invece optato per una strategia diametralmente opposta, che ha visto l'inserimento nel golden standard di un gran numero di annotazioni, molto spesso associate ad entità difficili da disambiguare correttamente. Come deducibile dai risultati ottenuti nelle fasi di spotting e disambiguazione, una scelta di questo tipo introduce complicazioni su entrambi i versanti: se dal punto di vista dello spotting è difficile per il sistema individuare come menzioni porzioni di testo che nella maggior parte dei dataset non sono considerati tali, dal punto di vista della disambiguazione è spesso difficile associare ad un testo l'entità corretta, trattandosi molto spesso di componenti della frase come aggettivi o nomi comuni, di frequente strettamente legate al contesto del documento stesso. Alla luce di tali considerazioni risulta chiara la difficoltà nello sviluppo di una tecnologia di spotting e disambiguazione in grado di offrire prestazioni comparabili su tutti i dataset. Ciononostante è chiaro come, con una procedura di spotting perfetta, il sistema sarebbe in grado di associare alle menzioni il senso ritenuto più pertinente con percentuali di successo decisamente superiori a quelle mostrate negli esperimenti di annotazione precedentemente illustrati. Questo dato contribuisce a sottolineare ancora una volta la primaria importanza delle prestazioni della fase di spotting per avere garanzie di efficiente conclusione dell'attività di annotazione, pur senza negare la perfettibilità di una procedura quale la disambiguazione le cui metriche lasciano ampio spazio al miglioramento, essendo in alcuni casi piuttosto lontane dalla perfezione.

6.4 Modifiche a WAT

In seguito all'analisi dei report delle annotazioni effettuate sulla versione iniziale di WAT e alle sperimentazioni preliminari svolte, si è deciso di procedere con alcune modifiche ad ognuno degli stage della pipeline di annotazione, nel tentativo di contrastare o arginare l'incidenza dei problemi riscontrati. A differenza dei dati presentati in precedenza, i quali facevano riferimento alle diverse fasi dell'annotazione esaminate una alla volta, gli esperimenti svolti in questa fase sono relativi all'applicazione di quanto evidenziato dai precedenti dati all'intero processo annotativo, e sono volti a misurare l'influenza in termini di risultato finale dell'annotazione, delle varie migliorie suggerite durante le analisi iniziali.

6.4.1 Spotter

È bene, prima di descrivere le modifiche applicate allo spotter di WAT, riassumerne brevemente la struttura e il funzionamento. La prima fase dello spotting prevede la tokenizzazione del testo in input mediante uno strumento di parsing (presentato nel dettaglio nelle sezioni precedenti). Questa tipologia di strumenti fornisce, per ogni token, informazioni aggiuntive, quali indicazione di appartenenza di un token ad una named entity e informazioni di tipo part-of-speech. Lo spotter di WAT effettua nel testo la ricerca delle menzioni da annotare utilizzando come indicazioni sul dove effettuare tale operazione una lista di cosiddetti suggerimenti, nella versione base dello spotter tali suggerimenti sono costituiti esclusivamente dalle porzioni di testo classificate come named entity, ma in alcuni dei casi visti in precedenza sono stati aggiunti ai suggerimenti anche nomi propri, nomi comuni e aggettivi. La particolare struttura dello spotter di WAT fa sì che, qualora una porzione di testo sia stata precedentemente classificata come poco rilevante (ossia non sia stata inclusa tra i suggerimenti), essa non possa essere classificata come menzione candidata all'annotazione. Per quanto concerne questa fase dell'annotazione, sono stati testati in un esperimento di tipo Sa2KB i tre spotter (nelle loro differenti configurazioni) presentati nell'ambito dell'analisi dello spotting, al fine di valutare quale fosse la combinazione ideale che permettesse la massimizzazione delle metriche sull'esperimento di annotazione, valutando quanto il variare della qualità dello spotting impattasse sui risultati finali di un'annotazione.

OpenNLP

Trattandosi in questo caso del tokenizer di default presente in WAT, per stabilire una base da usare come riferimento nell'analisi dei tokenizer alternativi è stato effettuato tramite GERBIL un esperimento di tipo Sa2KB sugli stessi dataset utilizzati in tutti i test visti in precedenza. I risultati sono stati ottenuti applicando lo Strong annotation match, e sono riportati nella tabella 6.14.

Stanford Parser

Come visto in precedenza, l'analisi effettuata dallo Stanford Parser produce sia POS label che NE label, ed entrambe tali informazioni sono state sfruttate nelle due fasi che

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.662	0.733	0.605
AQUAINT	0.351	0.364	0.340
DBpedia Spotlight	0.228	0.622	0.139
IITB	0.211	0.458	0.137
KORE 50	0.493	0.519	0.469
MSNBC	0.627	0.717	0.557
N3 Reuters-128	0.490	0.558	0.436
N3 RSS-500	0.410	0.385	0.438

Tabella 6.14: Risultati Sa2KB ottenuti sull'intera pipeline di annotazione con tokenizer OpenNLP e disambiguazione con voting scheme

hanno contraddistinto l'analisi di questo sistema. Tali fasi ricalcano l'approccio mantenuto durante la fase di testing dello spotter.

Prima fase: Le entità

Il primo approccio con il nuovo tokenizer è avvenuto nel modo classico: di tutti i token prodotti dallo Stanford Parser, solo quelli marcati come entità, sono stati segnalati come suggerimenti allo spotter, in modo da dirigere l'analisi verso quelle porzioni di testo ritenute più influenti nel contesto del documento. Allo scopo di svolgere un confronto tra le prestazioni di WAT con il nuovo tokenizer e le precedenti è stato svolto un altro esperimento, analogo al precedente che ha fornito i risultati riportati nella tabella 6.15.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.759	0.763	0.755
AQUAINT	0.348	0.367	0.330
DBpedia Spotlight	0.185	0.469	0.115
IITB	0.191	0.530	0.116
KORE 50	0.449	0.491	0.413
MSNBC	0.640	0.492	0.570
N3 Reuters-128	0.518	0.571	0.473
N3 RSS-500	0.392	0.333	0.477

Tabella 6.15: Risultati Sa2KB ottenuti sull'intera pipeline di annotazione con tokenizer Stanford Parser e disambiguazione con voting scheme

Seconda fase: Aggiunta dei nomi propri

Analizzando i risultati ottenuti nella fase precedente si è notato che alcune delle annotazioni tralasciate dal sistema facevano riferimento a menzioni che, pur non essendo rilevate come entità dal tokenizer, erano facilmente riconducibili ad entità annotabili. Allo scopo

di aumentare il recall del sistema, tentando di evitare perdite in termini di precision, si è pertanto pensato di sfruttare le part-of-speech label prodotte dal parser, e in particolare quelle relative ai nomi propri, inserendole nella lista dei suggerimenti insieme alle entità già presenti, portando lo spotter ad analizzare anche parti del testo non espressamente legate ad un'entity. I risultati ottenuti, riportati nella tabella seguente, sono stati solo parzialmente in linea con le aspettative, nonostante infatti sia stato rilevato l'auspicato aumento del recall, seppur non in maniera massiccia, si è verificato anche un calo della precision che, pur non impedendo in alcuni casi un aumento delle prestazioni complessive, espresse in termini di F1, ha portato a valutare come rischiosa l'aggiunta dei nomi propri alla lista dei suggerimenti, in quanto si è ritenuto che il rischio di introdurre nel risultato finale una folta lista di annotazioni non pertinenti sia sovradimensionato rispetto agli eventuali vantaggi complessivi portati da una procedura di spotting comprensiva dei nomi propri.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.771	0.747	0.796
AQUAINT	0.361	0.350	0.371
DBpedia Spotlight	0.198	0.442	0.127
IITB	0.208	0.503	0.132
KORE 50	0.506	0.540	0.476
MSNBC	0.653	0.694	0.615
N3 Reuters-128	0.519	0.694	0.615
N3 RSS-500	0.394	0.326	0.497

Tabella 6.16: Risultati Sa2KB ottenuti sull'intera pipeline di annotazione con tokenizer Stanford Parser (con nomi propri) e disambiguazione con voting scheme

Factorie

L'ultimo dei tokenizer testato è stato quello basato su Factorie, il quale ha evidenziato durante l'analisi dello spotting prestazioni in linea con quelle garantite dallo Stanford Parser, ma senza pagare lo scotto di un rallentamento nell'esecuzione come verificatosi con quest'ultimo. Per mantenere coerente il confronto con le sperimentazioni effettuate nel caso del precedente tokenizer, lo studio di Factorie ha visto due fasi iniziali corrisponenti a quelle previste dalla sperimentazione dello Stanford Parser, alle quali ne sono state aggiunte due ulteriori la cui natura è stata espressa in precedenza nella fase di analisi dello spotting.

Prima fase: Le entità

Come nell'analoga situazione vista in precedenza, in questa prima fase si è stabilito che lo spotter ricevesse come suggerimenti per l'analisi esclusivamente quei token individuati come named entity, aspettandosi di ottenere prestazioni paragonabili a quelle offerte dall'utilizzo del tool di Stanford. Tali aspettative non sono state disattese, come è possibile

evincere dalle metriche riportate in tabella 6.17. Le performance ottenute sono state in generale in linea con quelle mostrate nella prima fase dell'annotazione eseguita utilizzando lo Stanford Parser, ma col vantaggio di essere ottenute con un sistema più efficiente sul piano del tempo di esecuzione.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.778	0.775	0.781
AQUAINT	0.342	0.352	0.332
DBpedia Spotlight	0.196	0.565	0.118
IITB	0.202	0.544	0.124
KORE 50	0.402	0.472	0.350
MSNBC	0.685	0.736	0.640
N3 Reuters-128	0.520	0.545	0.498
N3 RSS-500	0.380	0.363	0.400

Tabella 6.17: Risultati Sa2KB ottenuti sull'intera pipeline di annotazione con tokenizer Factorie (Standard) e disambiguazione con voting scheme

Seconda fase: Aggiunta dei nomi propri

Come lecito aspettarsi, anche con l'utilizzo di Factorie, è emersa dalle analisi dei report, una tendenza dell'annotatore a tralasciare entità giudicate facili da annotare, ma evidentemente non suggerite allo spotter a causa del loro mancato riconoscimento come entità. Si è pertanto provveduto ad aggiungere alla lista dei suggerimenti, anche in questo caso, quei token marcati come nomi propri dal tokenizer. In questo caso ancor più che nell'analogo relativo allo Stanford Parser, i risultati non sono stati giudicati totalmente in linea con le aspettative. Come è possibile notare dalla tabella che segue, ad un marcato calo della precision non è corrisposto un adeguato innalzamento del recall, portando nel complesso a misure di F1 in linea con quelle ottenute indicando allo spotter di effettuare la ricerca delle mention solamente in concomitanza con le entità, al prezzo però di una precisione molto inferiore, segno della presenza nelle annotazioni prodotte di numerosi elementi non pertinenti.

Fasi aggiuntive: Aggiunta dei nomi comuni e degli aggettivi

Visti i risultati dell'analisi dello spotting su alcuni dei dataset presi in considerazione si è deciso di procedere con l'aggiunta ai suggerimenti dei token etichettati come nomi comuni dal tokenizer, oltre a quelli propri aggiunti al passo precedente. In ultimo, visto il netto aumento del recall dello spotting in alcuni dataset osservato durante le sperimentazioni preliminari, si è proceduto con l'aggiungere ai suggerimenti anche i token relativi agli aggettivi.

I dati relativi a tali ultime due configurazioni non vengono riportati in quanto ritenuti non particolarmente indicativi, infatti come già evidenziato durante l'analisi approfondita dello spotter, l'aggiunta di queste categorie lessicali all'insieme delle menzioni da

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.764	0.738	0.792
AQUAINT	0.344	0.319	0.374
DBpedia Spotlight	0.222	0.453	0.149
IITB	0.225	0.453	0.149
KORE 50	0.518	0.533	0.504
MSNBC	0.674	0.677	0.672
N3 Reuters-128	0.500	0.498	0.502
N3 RSS-500	0.338	0.366	0.313

Tabella 6.18: Risultati Sa2KB ottenuti sull'intera pipeline di annotazione con tokenizer Factorie (con nomi propri) e disambiguazione con voting scheme

disambiguare porta ad un incremento delle prestazioni esclusivamente su alcuni dataset, portando ad un calo invece sui restanti, con incrementi tali da non giustificare le corrispondenti decrescite introdotte.

Scelta del tokenizer

Da un confronto tra le metriche ottenute durante gli esperimenti di annotazione effettuati è possibile delineare un scenario ampiamente in linea con le conclusioni tratte durante l'analisi generale dello spotting. È infatti evidente come la natura dei dataset influenzi pesantemente la qualità delle annotazioni prodotte, e come il tentativo di migliorare le prestazioni su un determinato tipo di documento finisca con il peggiorare i valori ottenuti su un altro tipo, mostrando la necessità, in attesa di tecniche più raffinate che permettano un discernimento preciso delle annotazioni grazie ad un criterio ben fondato di pertinenza, di avere un sistema il più generico possibile, perché sia in grado di affrontare le diverse tipologie di problema in maniera equilibrata senza snaturarne le prestazioni per farle perfettamente aderire ad un caso specifico.

Considerate le metriche prodotte dalle varie configurazioni possibili dei tokenizer testati, si è scelto di assumere come nuova tecnica di tokenizzazione base quella basata su Factorie, utilizzando come suggerimenti per lo spotting solamente le porzioni di testo identificate come named entity durante la procedura di parsing. Questa scelta è sostenuta da diverse motivazioni: innanzitutto il rischio che si corre ad accomunare nella disambiguazione entità "certe" classificate dal tokenizer e entità "non certe" quali nomi e aggettivi è di influenzare negativamente la procedura di ricerca del corretto senso di una menzione. Va infatti ricordato che nel voting scheme utilizzato sono le menzioni stesse a votare di volta in volta per le altre presenti nel documento. Dando validità uguale ad entrambi i tipi di entità il rischio è che entità "non certe" devino la procedura di votazione portandola lontano dalla giusta direzione. Altra questione a svantaggio di queste configurazioni è quella relativa alle prestazioni: è infatti evidente che pur aumentando il recall, a tale aumendo corrisponda un drastico calo della precision, che richiederebbe un intervento decisamente più profondo in fase di pruning pena un calo generalizzato delle prestazioni; senza contare che l'aumento di recall, salvo casi specifici, non è sufficientemente consistente da poter

accettare un calo di precisione, tantopiù considerando che dovendo disambiguare un numero più alto di menzioni aumenta vertiginosamente il tempo di annotazione, senza però ottenere un'adeguata contropartita in termini prestazionali.

I risultati ottenuti con i tre diversi tokenizer, divisi per F1, precision e recall, sono presentati nei grafici riportati in figura 6.1, nei quali si riportano per brevità soltanto i risultati ottenuti da Stanford e Factorie nella loro versione standard, senza quindi le differenti configurazioni testate, le quali vanno comunque tenute in considerazione come fondamenta di eventuali sviluppi futuri. Dall'analisi dei risultati, e in particolare dei valori di F1, è facile notare come tutti e tre i nuovi metodi considerati abbiano migliorato le prestazioni del sistema, giustificando la scelta effettuata di utilizzare Factorie come tokenizer mostrando come, salvo in rari casi, nei dataset in cui le metriche ottenute da Factorie non siano le migliori, esse siano comunque in linea con quelle dei concorrenti.

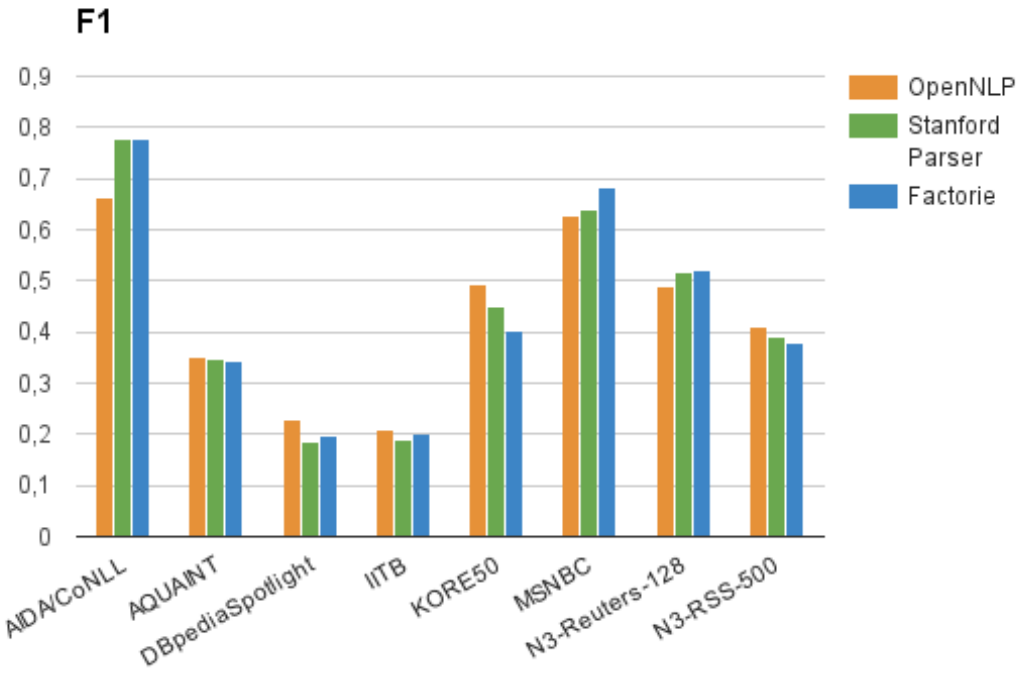


Figura 6.1: Confronto tra i valori di F1, precision e recall ottenuti utilizzando i tre tokenizer sull'intera pipeline di annotazione con disambiguazione basata sul voting scheme

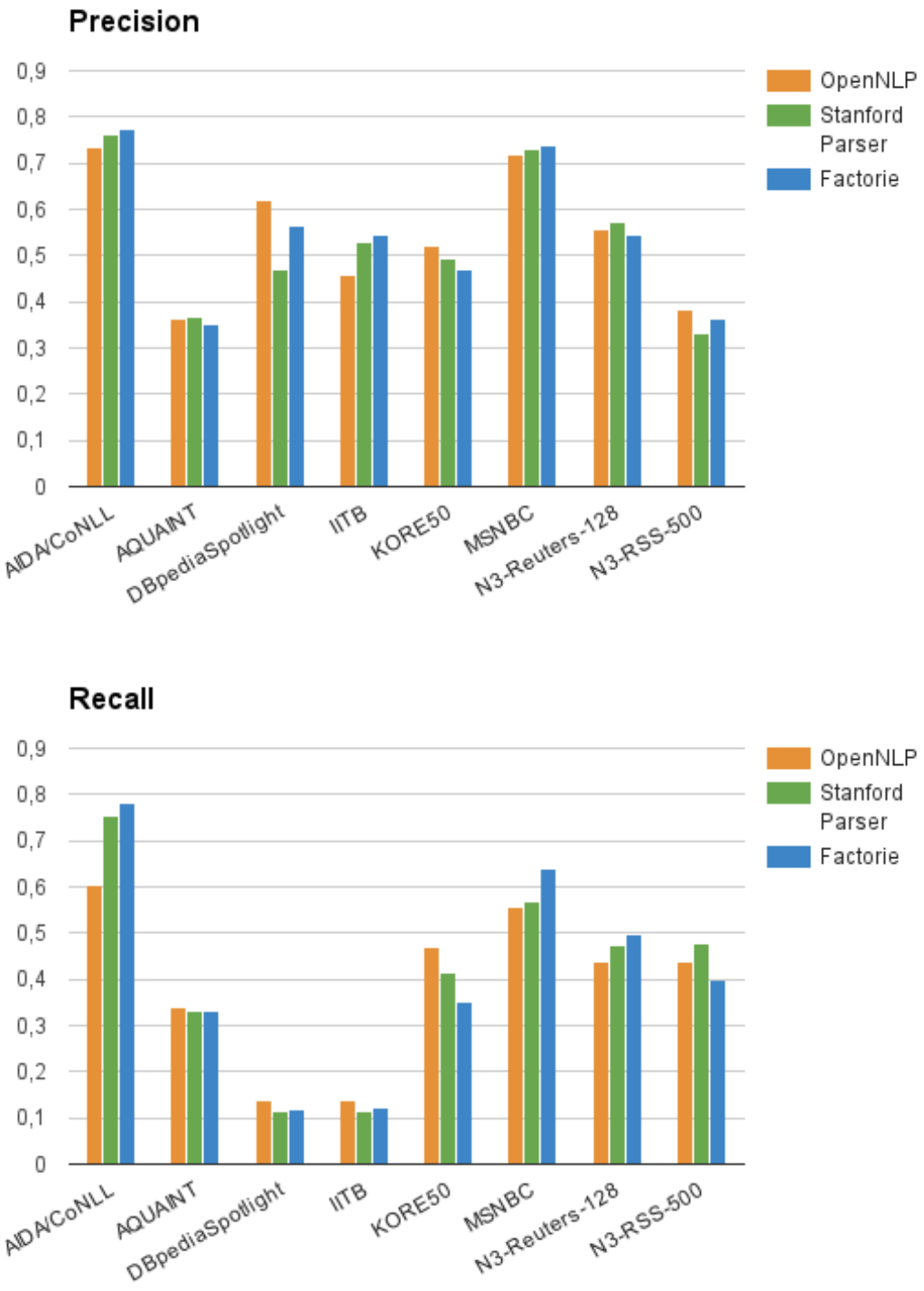


Figura 6.1: Confronto tra i valori di F1, precision e recall ottenuti utilizzando i tre tokenizer sull'intera pipeline di annotazione con disambiguazione basata sul voting scheme

6.4.2 Disambiguatore

Alla luce dei miglioramenti ottenuti attraverso la sola modifica della fase di tokenizzazione con l'utilizzo dei parser attualmente disponibili, si è passati ad analizzare la fase immediatamente successiva, durante la quale ad ogni menzione individuata dallo spotter viene assegnata la più pertinente delle pagine di Wikipedia che la descrive.

6.4.2.1 Disambiguazione con convergenza a step

Memori delle problematiche riscontrate durante la fase precedente, si è cercato di ovviare in questa fase al problema delle annotazioni mancanti invididuato dai report in precedenza e che si era tentato di arginare mediante l'aggiunta ai suggerimenti per lo spotter dei nomi propri e comuni e degli aggettivi. Dalle analisi più approfondite esposte nella sezione precedente, era emerso come oltre che alla categoria dei nomi propri la gran parte delle entità mancate fosse riconducibile alle categorie dei nomi comuni e degli aggettivi, si è di conseguenza teorizzata una strategia di disambiguazione che permettesse di aggiungere alle annotazioni prodotte normalmente quelle ottenute tramite la disambiguazione di queste menzioni. L'innovazione che sarà di seguito presentata è stata applicata al metodo base, ossia sul voting scheme originario di TAGME, il quale ha dimostrato in precedenza di essere quello più efficiente tra quelli proposti da WAT. Il concetto che è stato introdotto è quello di disambiguazione con convergenza a step. La fase preliminare è stata di introdurre il concetto di peso per i suggerimenti, realizzato sotto forma di suggerimenti *strong* e *light*. Durante la fase di tokenizzazione, i suggerimenti relativi alle named entity sono stati marcati come *strong suggestion*, mentre quelli relativi a nomi propri, nomi comuni e aggettivi sono stati marcati come *light suggestion*. La fase di disambiguazione è stata trasformata in un processo iterativo, nel quale ad ogni step la disambiguazione fosse relativa a suggerimenti di peso via via decrescente. Il principio base era di disambiguare inizialmente le menzioni strong, in modo da ottenere l'insieme standard di annotazioni già prodotto nelle precedenti strategie, e di disambiguare, procedendo con l'iterazione, le menzioni light, utilizzando come kernel per la votazione le menzioni già disambiguate in precedenza. L'introduzione di questa tecnica era supportata dall'idea che, aggiungendo alla lista delle menzioni prodotte dallo spotter tutte quelle relative a nomi e aggettivi, si sarebbe aumentata massivamente la percentuale di annotazioni corrette trovate, portando così ad un aumento generale delle prestazioni, seppur in concomitanza con un aumento del numero di annotazioni in eccesso. Le aspettative prevedevano la massimizzazione del recall, eventualmente a discapito della precision, in modo tale da poter demandare ad una successiva fase di pruning più stringente l'eliminazione delle meno pertinenti tra le annotazioni prodotte. I risultati delle sperimentazioni condotte sono risultati in parziale contrasto con le previsioni. Come si può osservare dal riepilogo nella tabella 6.19 è infatti possibile notare come questa strategia di disambiguazione abbia introdotto vantaggi limitati solo ad alcuni dataset specifici e non invece generalizzati. Tale problematica, già ampiamente sottolineata nelle fasi di studio precedenti è determinata dall'eterogeneità dei dataset, che si pone ancora una volta ad ostacolo di una variazione uniforme delle prestazioni, sottolineando la necessità di un algoritmo più sofisticato in grado di adattarsi alla situazione relativa al documento che sta annotando, e mostrando come unica alternativa al momento quella di operare una scelta sulla direzione nella quale concentrare gli

Dataset	$F1_{old}$	F1	Precision	Recall
AIDA/CoNLL	0.778	0.670	0.641	0.701
AQUAINT	0.342	0.365	0.346	0.385
DBpediaSpotlight	0.196	0.399	0.438	0.367
IITB	0.202	0.355	0.381	0.332
KORE50	0.402	0.430	0.402	0.462
MSNBC	0.685	0.514	0.568	0.469
N3-Reuters-128	0.520	0.417	0.519	0.348
N3-RSS-500	0.380	0.318	0.403	0.263

Tabella 6.19: Risultati convergenza a step (Con $F1_{old}$ è indicato il valore ottenuto prima dell'utilizzo della tecnica di convergenza a step)

sforzi: è infatti evidente come nel tentativo di migliorare le performance su determinati dataset si finisca con il peggiorare quelle ottenute sui restanti. Va sottolineato, a parziale spiegazione della discrepanza tra i dati relativi alle misure di qualità dello spotting precedentemente presentati e quelli relativi all'esperimento appena svolto, e in particolare al recall vistosamente più basso nel caso dell'annotazione vera e propria, che esso è alterato rispetto ai valori puri dello spotting da due fattori concomitanti. In primis anche ammettendo che tutte le menzioni siano correttamente individuate dallo spotter, l'imperfezione dell'algoritmo di disambiguazione non impedisce che esse vengano annotate con un'entità sbagliata, creando quello che viene classificato come error e che per ovvi motivi impedisce al recall di aumentare (si ricorda che il recall è la frazione delle annotazioni rilevanti *correttamente* individuata dall'annotatore, e che un'annotazione la cui menzione sia corretta non è considerata corretta a meno che non esista corrispondenza anche con le relative entità). Il secondo fattore che altera le misure è la natura stessa del problema Sa2KB per come è implementato nei sistemi di benchmarking utilizzati. Come già sottolineato durante la descrizione del problema, l'annotatore associa ad ogni annotazione prodotta un valore, compreso tra 0 e 1 che esprime la correlazione tra la menzione annotata e il senso attribuitole. Durante la fase di valutazione delle prestazioni, il sistema di benchmarking opera un'azione di sogliatura, nella quale facendo variare un valore soglia nell'intervallo tra 0 e 1, elimina tutte le annotazioni aventi punteggio inferiore a tale soglia e calcola di conseguenza le metriche, mantenendo come metriche finali dell'esperimento le migliori tra tutte quelle calcolate. È pertanto plausibile che durante tale operazione di sogliatura annotazioni con punteggio più basso prodotte dall'annotatore siano state tralasciate perché al di sotto del valore di soglia ideale. A puro scopo dimostrativo è stato realizzato lo stesso esperimento appena presentato facendo sì che il punteggio assegnato all'annotazione fosse uguale ad 1 per ogni annotazione prodotta. I valori ottenuti mostrano un netto innalzamento del recall, come previsto, ma un significativo peggioramento della precision, con una F1 finale estremamente bassa.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.380	0.248	0.810
AQUAINT	0.181	0.108	0.565
DBpediaSpotlight	0.353	0.278	0.482
IITB	0.303	0.235	0.427
KORE50	0.357	0.283	0.483
MSNBC	0.224	0.133	0.702
N3-Reuters-128	0.119	0.067	0.531
N3-RSS-500	0.130	0.072	0.662

Tabella 6.20: Risultati convergenza a step con soglia fissata a 1

6.4.2.2 Filtro sulle votazioni

Un dato emerso dall'analisi dei report ottenuti durante le sperimentazioni precedenti, tra le quali quelle sulla convergenza a step, ha rivelato che una delle concause del mancato aumento delle misure di recall nei test effettuati fosse l'alta incidenza di errori di disambiguazione prodotti nell'ambito delle menzioni strong. La seconda modifica apportata è stata pertanto orientata alla rimozione, per quanto possibile, di qualcuno di questi errori, i quali sono stati individuati come frequenti anche nel resto degli esperimenti effettuati. Avendo notato nei report che spesso gli errori di disambiguazione occorrevano in concomitanza di menzioni immerse in un determinato contesto disambiguate su entità non coerenti con il contesto stesso, si è provveduto ad analizzare alla radice il problema. Si è quindi notato che il tokenizer in molti dei casi in oggetto, era stato in grado di identificare correttamente il tipo di entità che era stato poi annotato in maniera sbagliata. Sulla base di quanto emerso da questa analisi, si è sviluppato un filtro sulle votazioni. Avvalendosi delle classificazioni sulle entità presenti in DBpedia già precedentemente presentate nell'ambito delle statistiche sviluppate nella fase di error reporting, e andando ad agire come al passo precedente sull'algoritmo base con voting scheme, si è creato un controllo basato sul confronto tra Entity Type DBpedia e NE label del tokenizer. Tale filtro fa in modo che, nel momento in cui per una menzione ambigua si sta votando per un determinato senso possibile, si verifichi quale classe DBpedia assegni a tale senso, e quale label il tokenizer avesse assegnato alla mention. Qualora le due classi siano compatibili, il valore dei voti espressi a favore del senso corrente viene aumentato, penalizzando quindi i sensi che facciano riferimento a classi differenti da quella individuata dal parser. L'obiettivo di questa modifica era quello di favorire, in fase di disambiguazione, i sensi più corretti rispetto al contesto generale del documento, sfruttando il fatto che il tokenizer è in grado di individuare a quale categoria appartenga uno stesso termine nel preciso contesto in cui è utilizzato. Si pensi ad esempio ad un termine quale "England", a seconda della struttura del documento, il tokenizer è in grado di attribuire al termine l'etichetta di *Location*, qualora esso sia utilizzato in un contesto che fa riferimento al senso geografico, o l'etichetta di *Organization*, qualora si tratti invece ad esempio di un documento in cui si parla di sport facendo riferimento alla nazionale inglese. Tale tipologia di informazione rimane normalmente inutilizzata nel classico processo di disambiguazione, nonostante risulti es-

sere utile in certe situazioni per contrastare il problema della polisemia. Dall'introduzione di questo filtro ci si aspettava un aumento delle prestazioni, seppur di non grande entità. Tali aspettative non sono state disattese e, come mostrano i dati riportati di seguito, il miglioramento è relativo, sia sul fronte della precision che di recall e F1, a pressoché tutti i dataset testati salvo rari casi nei quali le prestazioni sono in linea con quelle ottenute senza il filtro.

Dataset	$F1_b$	$F1_f$	$Prec_b$	$Prec_f$	Rec_b	Rec_f
AIDA/CoNLL	0.778	0.785	0.775	0.782	0.781	0.789
AQUAINT	0.342	0.341	0.352	0.386	0.332	0.305
DBpediaSpotlight	0.196	0.201	0.565	0.58	0.118	0.121
IITB	0.202	0.201	0.544	0.542	0.124	0.124
KORE50	0.402	0.402	0.472	0.472	0.350	0.350
MSNBC	0.685	0.690	0.736	0.742	0.640	0.645
N3-Reuters-128	0.520	0.515	0.545	0.535	0.498	0.496
N3-RSS-500	0.380	0.382	0.363	0.378	0.400	0.387

Tabella 6.21: Risultati del filtro EntityType-NE Label. I valori relativi alla versione base dell'algoritmo (Factorie standard e voting scheme) sono indicati con il pedice *b*, mentre quelli ottenuti utilizzando il filtro sono indicati con il pedice *f*.

6.5 Sulle prestazioni del nuovo WAT

Si riportano infine, per fini di chiarezza e completezza, i confronti tra le prestazioni di WAT prima delle operazioni di aggiornamento e quelli ottenuti tramite la modifica della procedura di spotting e il filtro sulle entità precedentemente illustrati, rimandando al prossimo capitolo le conclusioni sulle prestazioni generali di WAT in relazione agli altri annotatori disponibili.

Come è possibile notare dai grafici riportati in figura 6.2, i risultati ottenuti seguono l'andamento generale osservato in tutti gli esperimenti presentati in precedenza, mostrando come gli incrementi di performance ottenuti su determinati dataset corrispondano a decrementi su altri. Al netto delle problematiche strettamente legate alla natura dei dataset impiegati nei test, è comunque presente un incremento generale di prestazioni, e, nei casi in cui tale incremento non sia stato ottenuto, i valori delle metriche sono in generale rimasti in linea con quelli della precedente versione.

A completamento dell'analisi svolta si è ritenuto opportuno offrire una visione d'insieme dei punti critici dell'annotatore, approfondire gli esperimenti sullo spotting effettuati nella prima parte di questo capitolo, concentrando l'attenzione sulle caratteristiche del sistema che possono costituire un problema per la corretta riuscita del processo di annotazione. Si è proceduto con due analisi, una effettuata utilizzando lo spotter così come configurato nella soluzione proposta come *nuovo WAT*, ossia quello basato sull'utilizzo di Factorie in versione standard, e l'altra utilizzando lo spotter che si è precedentemente

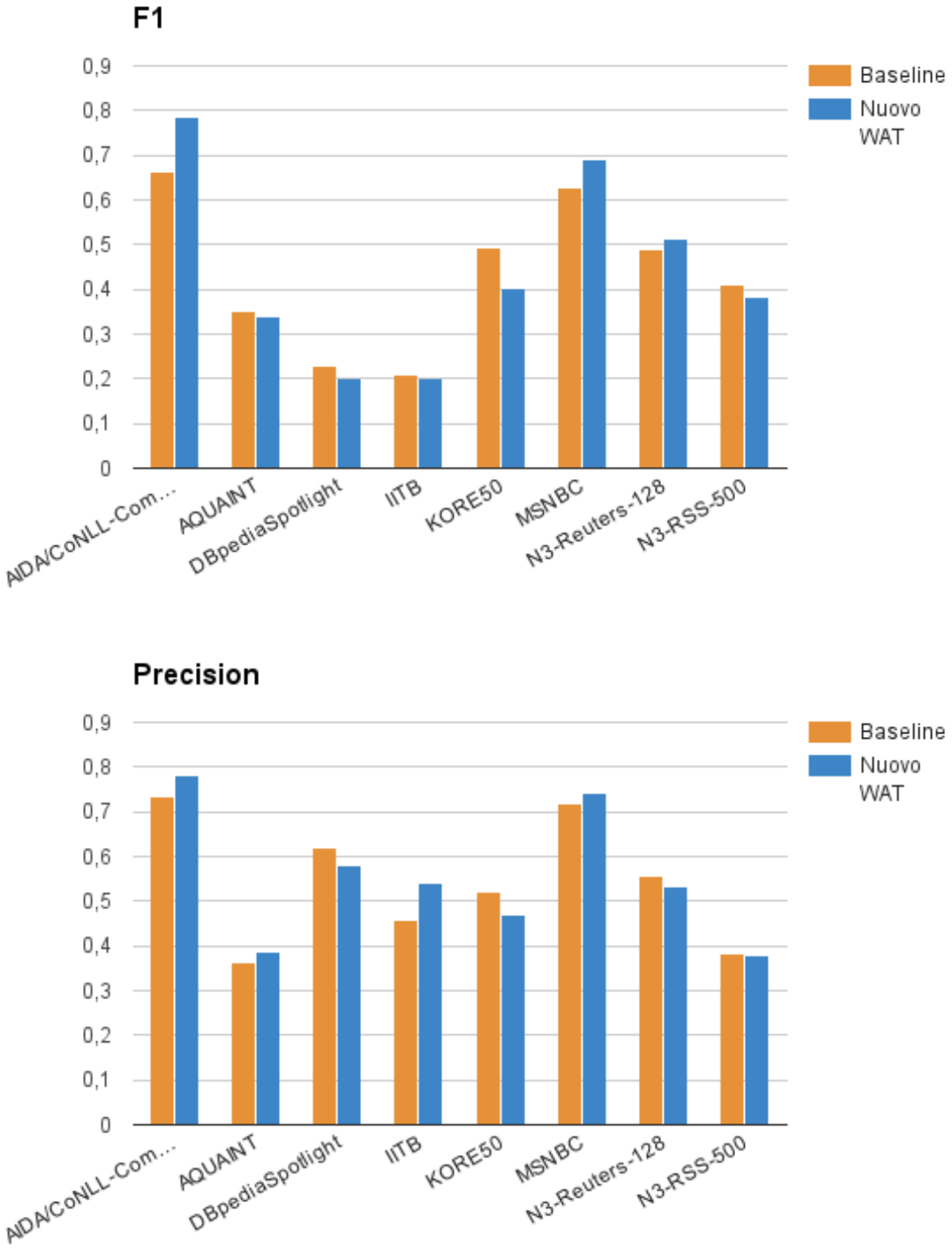


Figura 6.2: Confronto tra i valori di F1, precision e recall ottenuti rispetto alla baseline

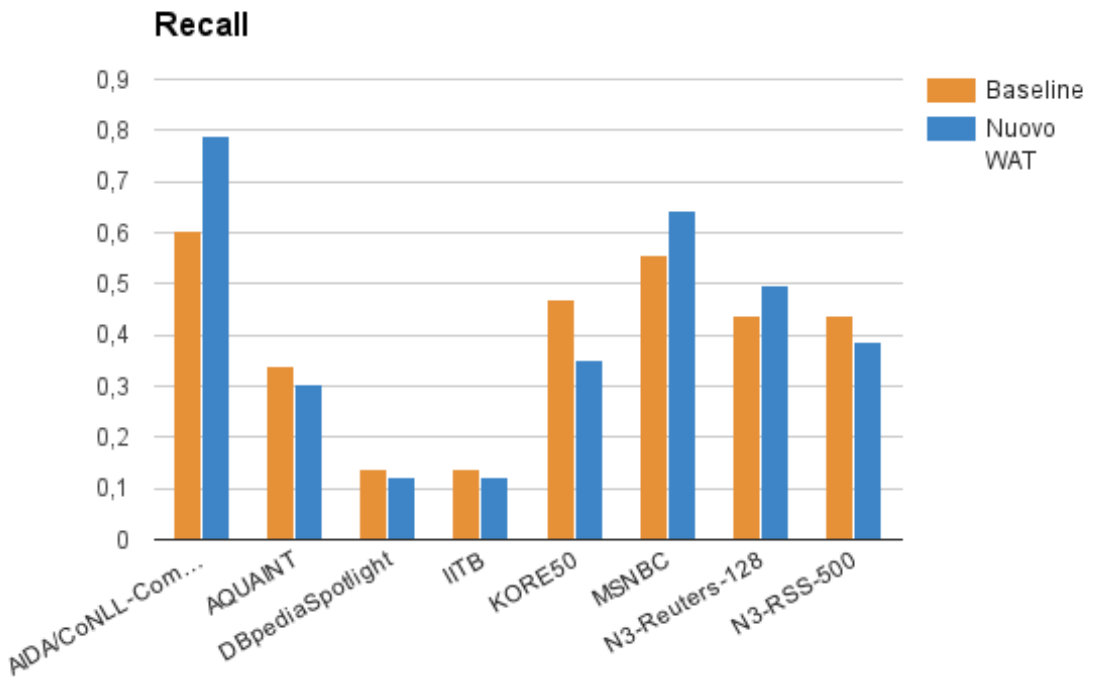


Figura 6.2: Confronto tra i valori di F1, precision e recall ottenuti rispetto alla baseline

indicato come quello in grado di offrire le maggiori prestazioni in termini di recall, ossia quello che si avvale di Factorie con l'aggiunta nei suggerimenti di nomi propri, nomi comuni e aggettivi. Per motivi di brevità si riportano esclusivamente i risultati ottenuti sui dataset AIDA/CoNNL, AQUAINT e MSNBC, ritenuti i più indicativi, nonché quelli più spesso citati nella letteratura. Per ognuno dei dataset è stata eseguita con WAT la fase di spotting, confrontando le menzioni individuate dal sistema con quelle presenti nei golden standard. Le menzioni analizzate nel dettaglio sono state quelle classificate come *correct* e *missing*, per ognuna delle annotazioni corrispondenti si è effettuata in WAT la ricerca dei sensi possibili, al fine di verificare se l'introduzione di tecniche di spotting e disambiguazione più efficienti sarebbe sufficiente a garantire un incremento prestazionale o se al contrario si andrebbe comunque incontro ad una mancanza di miglioramenti causati dall'insufficienza dei dati ricavati dall'analisi di Wikipedia.

Le menzioni possono appartenere ad una delle seguenti tipologie:

- **Disambiguabili**: tra le entità presenti in WAT in relazione a tale menzione esiste quella corretta indicata nel golden standard;
- **Non disambiguabili**: non esiste tra le entità associate da WAT alla menzione quella indicata dal golden standard;
- **Inesistenti**: WAT non dispone di questa menzione e quindi non è in grado di associarle alcuna entità.

Le informazioni ricavabili da tale suddivisione delle menzioni correct e missing sono molteplici: i dati sulle menzioni correttamente individuate dallo spotter mettono in luce

quanti degli errori di disambiguazione sarebbero evitabili modificando la procedura di disambiguazione (menzioni disambiguabili) e quanti invece verrebbero commessi anche in presenza di un disambiguatore perfetto, a causa del fatto che nella struttura di Wikipedia sfruttata da WAT non esiste modo di associare ad una menzione il senso corretto (menzione non disambiguabile) o addirittura non è presente alcun senso possibile (menzioni inesistenti) perché il testo della menzione non è mai utilizzato come àncora in Wikipedia. I dati sulle menzioni missing mostrano invece il risultato complementare: se lo spotter fosse perfetto e in grado quindi di individuare il 100% delle menzioni del golden standard, quante di esse sarebbero poi disambiguabili correttamente e quante invece non sarebbe possibile annotare con la corretta entità perché mancante tra i sensi possibili (menzione non disambiguabile) o non presente nell'archivio delle possibili àncore di Wikipedia?

Dataset	Correct	Disambiguabili	Non Disambiguabili	Inesistenti
AIDA/CoNLL	26113	25753	360	0
AQUAINT	282	279	3	0
MSNBC	515	496	19	0

Dataset	Missing	Disambiguabili	Non Disambiguabili	Inesistenti
AIDA/CoNLL	1702	1495	31	176
AQUAINT	445	372	21	52
MSNBC	135	107	14	14

Dataset	$F1_d$	$Precision_d$	$Recall_d$
AIDA/CoNLL	0.886	0.850	0.926
AQUAINT	0.346	0.314	0.384
MSNBC	0.733	0.706	0.763

Dataset	$F1_{dp}$	$Precision_{dp}$	$Recall_{dp}$
AIDA/CoNLL	0.955	0.986	0.926
AQUAINT	0.553	0.989	0.384
MSNBC	0.852	0.963	0.763

Tabella 6.22: Statistiche sullo spotting con Factorie (versione standard, come da tabella 6.6). Missing e Correct fanno riferimento alle menzioni individuate dallo spotter. Disambiguabili = entità corretta presente tra quelle possibili per la menzione, Non disambiguabili = tra le entità collegate alla menzione non esiste quella corretta, Inesistenti = la menzione non appare come àncora in Wikipedia e pertanto non esistono sensi possibili. Metriche F1, Precision e Recall ideali, calcolate in relazione all'intero processo di annotazione. Con il pedice d si indicano le prestazioni calcolate in presenza di disambiguazione perfetta ma nessuna operazione di pruning dei risultati, mentre il pedice dp fa riferimento ai risultati ottenuti introducendo nel sistema oltre al disambiguatore perfetto anche una tecnica di pruning in grado di rimuovere dal risultato tutte le annotazioni in eccesso.

La suddivisione delle menzioni in tre tipologie aiuta pertanto ad effettuare un'attribuzione delle responsabilità nel caso di prestazioni poco soddisfacenti: è infatti impossibile porre rimedio in caso di menzione inesistente in quanto in tali casi non è l'annotatore ad essere poco accurato, bensì la struttura di Wikipedia a non fornire le necessarie informazioni. Qualora invece esista nel sistema la possibilità di annotare le menzioni non individuate dallo spotter con la corretta entità, risulta chiara la necessità di intervenire sulla procedura di spotting al fine di innalzare le prestazioni mediante l'annotazione delle menzioni al momento mancanti. Infine la presenza per le menzioni correttamente individuate delle corrette entità corrispondenti, lascia intendere la possibilità di miglioramento attraverso l'introduzione di più sofisticate tecniche di disambiguazione.

Dataset	Correct	Disambiguabili	Non Disambiguabili	Inesistenti
AIDA/CoNLL	26720	26639	381	0
AQUAINT	496	483	13	0
MSNBC	556	533	23	0

Dataset	Missing	Disambiguabili	Non Disambiguabili	Inesistenti
AIDA/CoNLL	1095	909	10	176
AQUAINT	231	168	11	52
MSNBC	94	70	23	14

Dataset	$F1_d$	$Precision_d$	$Recall_d$
AIDA/CoNLL	0.437	0.284	0.947
AQUAINT	0.209	0.124	0.664
MSNBC	0.258	0.153	0.820

Dataset	$F1_{dp}$	$Precision_{dp}$	$Recall_{dp}$
AIDA/CoNLL	0.966	0.986	0.947
AQUAINT	0.790	0.974	0.664
MSNBC	0.884	0.959	0.820

Tabella 6.23: Statistiche sullo spotting con Factorie (con nomi propri, nomi comuni e aggettivi), configurazione che massimizza il recall sulle menzioni individuate. Missing e Correct fanno riferimento alle menzioni individuate dallo spotter. Disambiguabili = entità corretta presente tra quelle possibili per la menzione, Non disambiguabili = tra le entità collegate alla menzione non esiste quella corretta, Inesistenti = la menzione non appare come àncora in Wikipedia e pertanto non esistono sensi possibili. Metriche F1, Precision e Recall ideali, calcolate in relazione all'intero processo di annotazione. Con il pedice d si indicano le prestazioni calcolate in presenza di disambiguazione perfetta ma nessuna operazione di pruning dei risultati, mentre il pedice dp fa riferimento ai risultati ottenuti introducendo nel sistema oltre al disambiguatore perfetto anche una tecnica di pruning in grado di rimuovere dal risultato tutte le annotazioni in eccesso.

Nella tabella 6.22 sono riportati i valori relativi allo spotting effettuato utilizzando la configurazione proposta come migliore al termine dell'analisi presentata, mentre nella tabella 6.23 sono riportati i risultati raggiunti utilizzando lo spotter che ha dimostrato in precedenza di offrire il recall più elevato. Si ricorda che le prestazioni ottenute dall'annotatore progettato in questa tesi sono state indicate in tabella 6.21 e usano lo stesso spotter di tabella 6.22.

Le metriche F1, Precision e Recall riportate nelle tabelle 6.22 e 6.23 stimano le prestazioni che sarebbe possibile ottenere in un'annotazione completa se il sistema fosse dotato di un disambiguatore perfetto, in grado di assegnare la corretta entità ad ogni menzione individuata dallo spotter della quale si possiede il giusto senso e sbagliasse tutte quelle per le quali non esiste il senso corretto nel sistema, non annotando quelle che non appaiono come àncora in Wikipedia. Nella tabella 6.22 sono riportate le prestazioni ideali dello spotter che utilizza Factorie nella sua versione standard (come da tabella 6.6), mentre nella tabella 6.23 sono mostrate quelle dello spotter Factorie (con nomi propri, nomi comuni e aggettivi) che garantisce il massimo recall. Per ognuno degli spotter considerati sono presentati due gruppi di metriche, relativi a due differenti configurazioni del sistema al quale si riferiscono. Il primo dei due set di misure fa riferimento ad un sistema nel quale allo spotter considerato sia abbinato un disambiguatore perfetto ma senza che sia applicata alcuna procedura di pruning dei risultati, aggiungendo pertanto al risultato finale tutte le menzioni in eccesso individuate, determinando un enorme numero di falsi positivi che comporta un significativo drop in Precision/F1. Il secondo gruppo di metriche fa invece riferimento ai risultati che sarebbe possibile ottenere abbinando al disambiguatore perfetto precedentemente illustrato anche un pruner perfetto, ossia in grado di rimuovere dal risultato tutte le annotazioni in eccesso, limitando quindi il numero di falsi positivi.

Capitolo 7

Conclusioni

Durante le analisi approfondite svolte sull'annotatore WAT è stato possibile mettere in luce diversi aspetti rivelatisi fondamentali nell'approccio al problema dell'annotazione testuale. Il primo dato emerso, nonostante non direttamente legato al problema dell'annotazione in sé è quello relativo all'influenza che i dataset utilizzati per le procedure di benchmarking hanno sulla totalità del processo di valutazione. Come si ha avuto modo di ripetere in più situazioni, la profonda eterogeneità dei dataset utilizzati come banco di prova per la topic annotation contribuisce a rendere particolarmente complicata la messa a punto di un sistema in grado di ottenere prestazioni competitive su tutti i fronti. Viene naturale chiedersi se, senza partire da un qualche insieme di linee guida definite a monte per la costruzione dei dataset abbia valore reale la valutazione di sistemi tramite problemi quali A2KB o Sa2KB, vista la quasi impossibilità, a partire dal testo grezzo, di operare una procedura di spotting così efficace da poter individuare tutte le menzioni da annotare. Tale interrogativo è avallato anche dall'alternanza di risultati ottenuti dagli altri annotatori sui quali sono stati svolti i test.

Entrando più a fondo nei risultati dell'analisi svolta sulle componenti principali di WAT, è possibile individuare due punti chiave i quali risultano complementari per una buona riuscita di un processo annotativo.

La fase di spotting si è confermata fondamentale premessa ad un efficiente lavoro di annotazione e i test svolti hanno dimostrato come essa sia necessariamente da migliorare, soprattutto in relazione a determinate tipologie di testo. E' stato infatti possibile verificare come le prestazioni ipotetiche pronosticate nell'ambito della verifica delle capacità di annotazione del sistema costituiscano un realistico upper bound sulle performance dell'annotatore, e come queste siano per ovvi motivi strettamente correlate alle prestazioni dello spotter. In riferimento a casi particolari di dataset, quali IITB, è emersa la necessità di diversificare la strategia di spotting, avvalendosi di un parser più sofisticato o espandendo in vari modi la copertura dello spotting sul testo, come visto nell'analisi dei tokenizer utilizzabili in WAT.

Come già anticipato durante la presentazione degli studi effettuati su WAT dal team che si è occupato del suo sviluppo, è stato possibile confermare il ruolo primario della fase di spotting, nonostante la maggiore attenzione tipicamente riservata agli algoritmi di disambiguazione. E' infatti stato dimostrato dagli esperimenti di tipo D2KB come in presenza di un ipotetico spotting perfetto le prestazioni dell'annotatore sarebbero sensi-

bilmente migliori rispetto a quelle ottenute in condizioni reali con un'annotazione iniziata dal testo semplice. Tuttavia questa conclusione non deve far pensare che, sebbene forieri di performance sopra la media, gli algoritmi di disambiguazione inclusi in WAT non lascino spazio di manovra per quanto riguarda il loro perfezionamento. I dati riportati nella tabella 6.13 mostrano infatti chiaramente come, soprattutto nel caso di alcuni tra i dataset più problematici, le prestazioni ottenute anche in presenza di spotting perfetto siano tutt'altro che ottimali.

Partendo dai risultati delle analisi effettuate sui primi due stage dell'annotazione in WAT e incrociandoli con quelli contenuti negli error report inclusi in GERBIL durante la prima fase del lavoro è stato possibile ideare alcune contromisure atte a cercare di contrastare i problemi più di frequente riscontrati migliorando così le prestazioni del sistema.

Innanzitutto, vista l'importanza rivestita dalla fase di spotting si è provveduto all'individuazione di un nuovo sistema di tokenization che permettesse una maggiore copertura del testo in fase di parsing. Dal punto di vista della disambiguazione si sono invece tentate differenti strade, mirate alla minimizzazione del rumore introdotto utilizzando una fase di spotting più lasca rispetto a quella di default.

Le modifiche apportate al sistema hanno infine contribuito ad innalzare, seppure in maniera non sempre marcata, le prestazioni rispetto a quelle originarie di WAT, consolidandone ulteriormente la posizione di stato dell'arte dell'annotazione testuale. È infatti possibile, confrontando i risultati ottenuti in seguito alle modifiche apportate, riportati in tabella 7.1, con quelli relativi agli altri annotatori presenti in GERBIL, riportati in tabella 6.2, notare come siano state mantenute le iniziali gerarchie, contribuendo però ad incrementare il vantaggio rispetto agli annotatori concorrenti su quasi tutti i dataset. Rimangono tuttavia esclusi dai miglioramenti prestazionali i dataset che più hanno risentito delle debolezze del sistema in fase di spotting e disambiguazione, in maniera particolare IITB e DBpediaSpotlight, la cui particolare struttura richiede verosimilmente l'introduzione di nuove procedure algoritmiche per l'analisi del testo, in quanto mal si adatta, come visto in precedenza, alle attuali caratteristiche strutturali di WAT.

Dataset	F1	Precision	Recall
AIDA/CoNLL	0.785	0.782	0.789
AQUAINT	0.341	0.386	0.305
DBpediaSpotlight	0.201	0.580	0.121
IITB	0.201	0.542	0.124
KORE50	0.402	0.472	0.350
MSNBC	0.690	0.742	0.645
N3-Reuters-128	0.515	0.535	0.496
N3-RSS-500	0.382	0.378	0.387

Tabella 7.1: Risultati del sistema dopo le modifiche

Sviluppi futuri

Sebbene le operazioni di refactoring abbiano portato miglioramenti rispetto alla base di partenza, in particolare nella fase di spotting, rimane la necessità di individuare una tecnica di spotting più sofisticata, per la quale sarebbe probabilmente necessario un parser più avanzato di quelli al momento disponibili, che permetta di massimizzare quanto più possibile la quantità di menzioni corrette individuate in fase di spotting, facendo contemporaneamente fronte all'inevitabile rumore aggiunto all'annotazione dalla presenza di annotazioni spurie mediante una procedura di potatura più robusta. Dal punto di vista della disambiguazione sono state gettate le basi per una possibile implementazione di un nuovo algoritmo che, partendo dal concetto di disambiguazione a step introdotto in precedenza e declinandolo in maniera più efficiente, permetta di ridurre, in collaborazione con il pruner auspicato in precedenza, la quantità di annotazioni irrilevanti prodotta, massimizzando così i vantaggi ottenuti da una fase di parsing più ricca di risultato. In ultimo l'analisi dell'errore effettuata tramite le funzionalità ora incluse in GERBIL, ha fatto sì che emergesse la necessità di introdurre in fase di disambiguazione un qualche concetto di "contesto", il quale agevoli la disambiguazione di una menzione scegliendo il senso più adatto al contesto nel quale essa si trova. Uno strumento di tale genere sarebbe in grado di risolvere gran parte dei casi di errore nell'annotazione al momento commessi da WAT, contribuendo in maniera sostanziale al consolidamento del suo primato nel campo dell'annotazione testuale.

Bibliografia

[1] P. Ferragina and U. Scaiella. *Tagme: on-the-fly annotation of short text fragments (by Wikipedia entities).* In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10), 1625-1628, 2010

[2] F. Piccinno and P. Ferragina. *From TagME to WAT: a new entity annotator.* In ERD'14, Proceedings of the First International Workshop on Entity Recognition & Disambiguation, hosted by ACM SIGIR Conference, 55-62, 2014

[3] M. Cornolti, P. Ferragina, and M. Ciaramita. *A framework for benchmarking entity-annotation systems.* In Proceedings of the 22nd World Wide Web Conference, 249-260, 2013.

[4] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann. *GERBIL – General Entity Annotation Benchmark Framework.* In Proceedings of the 24th WWW conference, 1133-1143, 2015.

[5] C. D. Manning, P. Raghavan, and H. Schütze *Introduction to Information Retrieval*, Cambridge University Press, 2008.