Università degli Studi di Trieste
Dipartimento di Ingegneria e Architettura

Corso di Laurea Triennale in Ingegneria dell'Informazione

Rapporto tecnico

Realizzazione di un Add-on per Google Docs per estrazione interattiva di pattern sintattici

Candidato:
Lorenzo Gasparini

Relatore:
Prof. Alberto Bartoli

Correlatore:
Prof. Eric Medvet

Anno Accademico 2015–2016

INDICE

ELENCO DELLE FIGURE

ELENCO DELLE TABELLE

INTRODUZIONE

Lo scopo di questo documento è quello di fornire un rapporto tecnico dell'attività di laboratorio svolta dal candidato come prova finale.

Il lavoro consiste nella realizzazione di un componente aggiuntivo per la piattaforma Google Docs, implementando un algoritmo di estrazione interattiva di entità dai documenti di testo.

L'algoritmo è stato sviluppato nel laboratorio di Machine Learning ed è descritto nel documento *Real-time Interactive Syntax-based Entity Extraction* con l'identificatore SJ-MC. Esso, basandosi su alcuni esempi forniti dall'utente di entità da estrarre e da non estrarre, li generalizza creando un estrattore corrispondente al pattern ricercato.

- Il primo capitolo è un manuale d'uso rivolto agli utenti finali dell'Add-on, che ne descrive le funzionalità e il flusso di funzionamento da un punto di vista esterno.
- Il secondo capitolo tratta i dettagli tecnici della realizzazione e le decisioni progettuali prese durante lo sviluppo del software.

1 MANUALE D'USO

Il primo passo per l'utilizzo dell'Add-on è l'apertura dello stesso dal menu contestuale del documento al quale è collegato. Il menu presente nell'editor Google Docs comprende infatti una voce *Componenti aggiuntivi* che contiene una lista degli Add-on disponibili per il documento corrente.

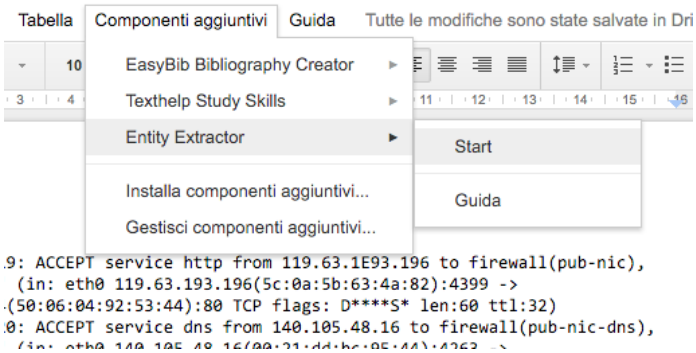

Figura 1: Apertura dell'Add-on dal menu.

Cliccando su *Start* l'Add-on viene avviato e compare la sidebar relativa a lato del documento. L'interazione con l'Add-on avviene tramite la sidebar.

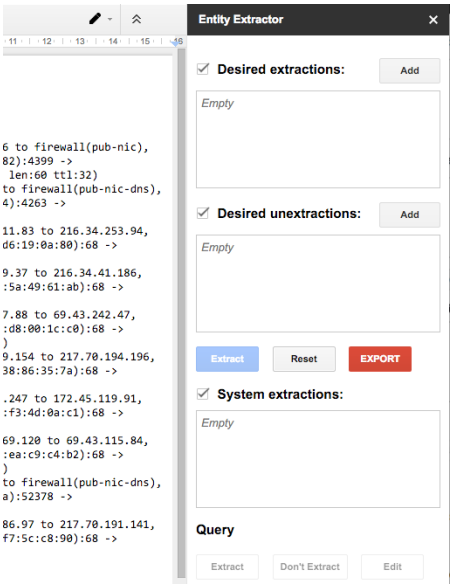

Figura 2: Stato iniziale della sidebar dopo l'apertura.

La sidebar è divisa in 5 sezioni:

1. La prima sezione è quella delle *Desired extractions*, e mostra l'insieme attuale delle stesse, permettendo di aggiungerne di nuove e di disattivare la loro evidenziatura cliccando nella checkbox di fianco al titolo.

2. La seconda sezione è analoga alla prima, ma è relativa alle *Desired unextractions*.

3. La terza sezione comprende i pulsanti:

   EXTRACT per effettuare l'estrazione delle entità dal documento usando l'insieme attuale di annotazioni;

   RESET per reimpostare l'Add-on allo stato iniziale;

   EXPORT per effettuare l'estrazione delle entità, come *Extract*, ed esportarle in un nuovo documento una per riga.

4. La quarta sezione è quella delle *System extractions*, e mostra l'insieme delle stesse una volta che l'estrazione è avvenuta. Permette inoltre di disabilitare la loro evidenziatura nel documento.

5. La quinta sezione è quella della *Query*. Una volta che l'estrazione è avvenuta, l'aggiunta manuale di nuove annotazioni viene disabilitata e l'algoritmo di *Active learning* sceglie una query da proporre all'utente, il quale può rispondere in tre modi:

   EXTRACT annota la query attuale come da estrarre, i.e. la aggiunge all'insieme delle Desired extractions ed esegue nuovamente l'estrazione;

   DO NOT EXTRACT annota la query attuale come da non estrarre, i.e. la aggiunge all'insieme delle Desired unextractions ed esegue nuovamente l'estrazione;

   EDIT riattiva la possibilità di aggiungere manualmente ulteriori Desired extractions e Desired unextractions, rimuovendo il vincolo di risposta binaria alla query per proseguire.

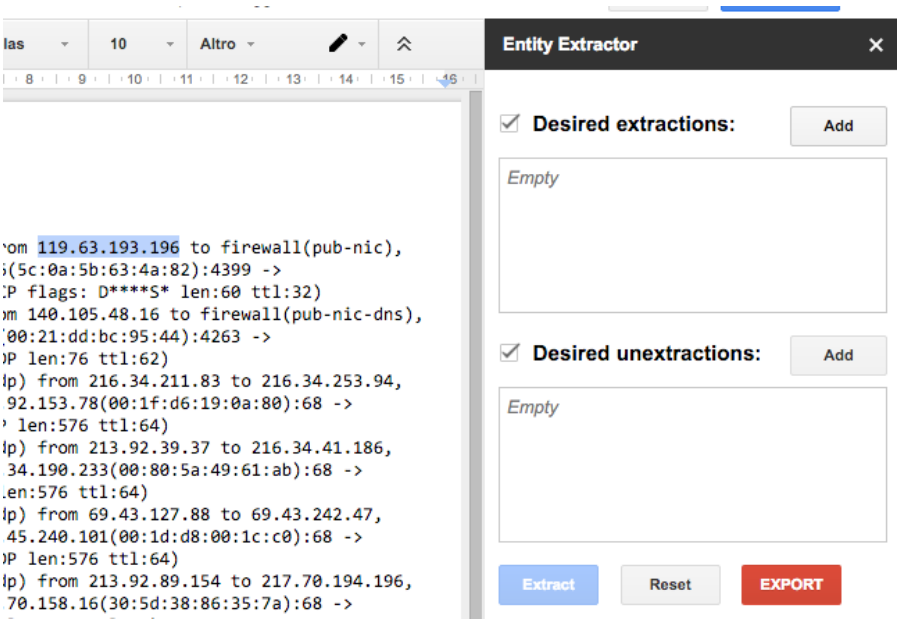

Figura 3: Aggiunta di una annotazione come desired extraction o desired unextraction.

Per aggiungere una nuova annotazione è sufficiente selezionare una porzione di testo del documento, e cliccare il pulsante *Add* relativo all'insieme al

quale si vuole aggiungere l'annotazione. Una volta aggiunta l'annotazione all'insieme corrispondente, essa verrà evidenziata nel documento con il suo colore, verde per le Desired extraction e rosso per le Desired unextraction (se l'evidenziatura è abilitata). Le annotazioni sono ordinate in base alla loro posizione nel documento.

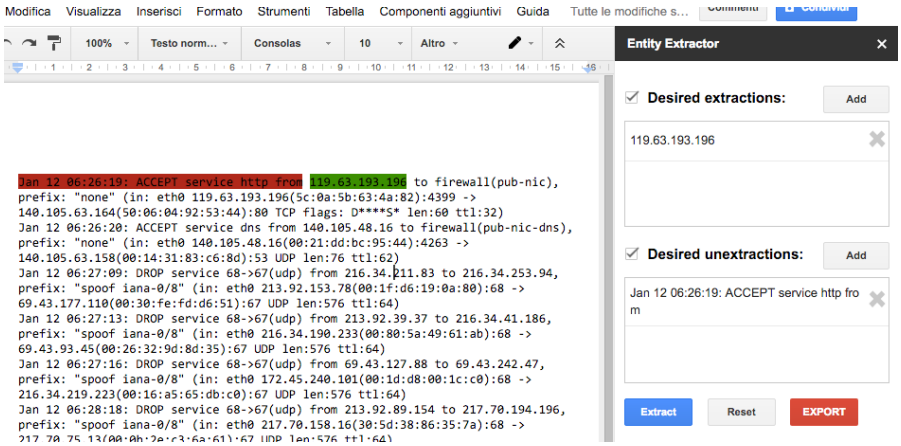

Figura 4: Annotazioni aggiunte come desired extraction e unextraction.

Cliccando su un elemento delle due liste, verrà selezionata nel documento la porzione di testo corrispondente. Se il testo in questione non è visibile il documento scorrerà fino al punto in cui esso compare.

Ogni annotazione a fianco ha un pulsante, rappresentato da una croce, che permette di eliminarla.

È bene ricordare che le Desired extractions e Desired unextractions non si possono sovrapporre, se si prova infatti ad aggiungere un'annotazione che si sovrappone con un'altra esistente si visualizzerà un messaggio d'errore.

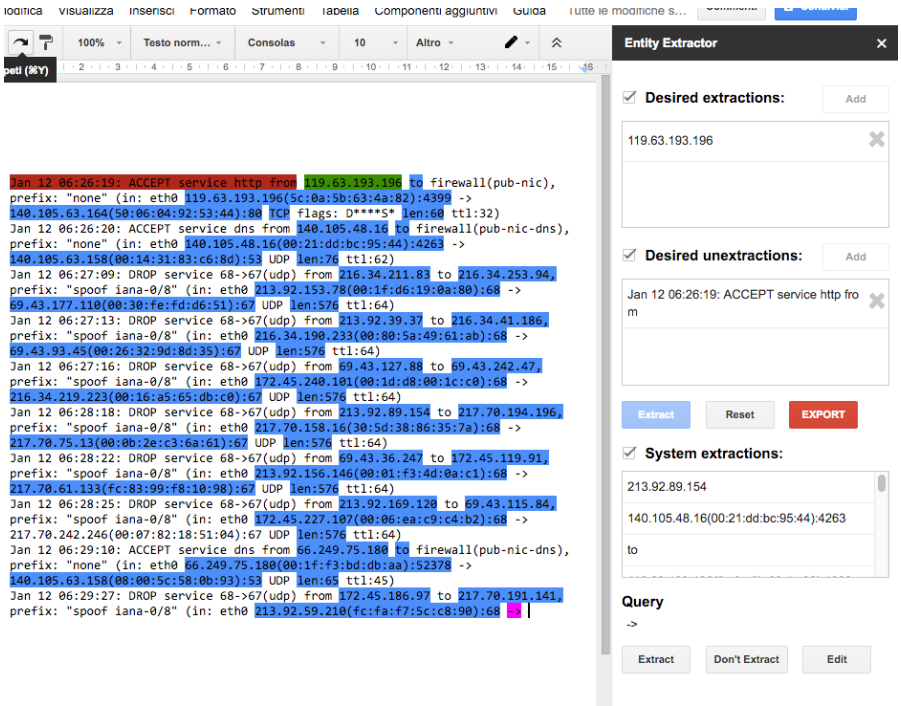

Figura 5: Risultato dell'algoritmo di estrazione.

Una volta aggiunte le annotazioni volute, si può eseguire l'algoritmo di estrazione. È necessario ci siano presenti almeno una Desired extraction ed una Desired unextraction.

Eseguito l'algoritmo, nel documento verranno evidenziate in blu le porzioni di testo corrispondenti alle System extractions, e verrà popolata la relativa lista nella sidebar. Anche in questo caso cliccando su un elemento della lista viene selezionata nel documento la porzione di testo a cui esso corrisponde.

Si è scelto, in contrasto a quanto descritto nell'articolo, di non tenere in considerazione le System extractions che si sovrappongono con Desired extractions o Desired unextractions; questo per migliorare la chiarezza visiva.

All'utente sarà inoltre proposta una query, e sarà disabilitata l'aggiunta di nuove annotazioni. La query sarà evidenziata nel documento con il colore magenta.

Come già detto, ci sono tre modi in cui l'utente può rispondere alla query.

Cliccando su *Extract* la query viene aggiunta all'insieme delle Desired extractions e viene eseguita una nuova iterazione dell'algoritmo.

Cliccando su *Do not Extract* la query viene aggiunta all'insieme delle Desired unextractions e, anche in questo caso, l'algoritmo di estrazione viene eseguito nuovamente.

Cliccando su *Edit* viene riattivata la possibilità di aggiungere annotazioni manualmente. Inoltre, l'evidenziatura delle System extractions viene automaticamente disabilitata per migliorare la chiarezza visiva.

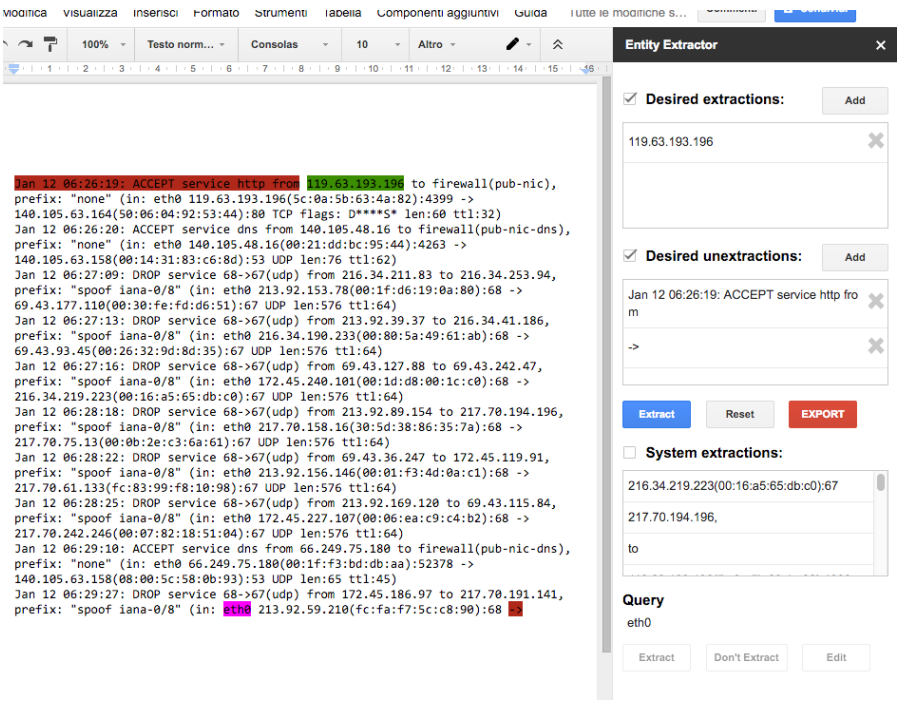

Figura 6: Modalità Edit

Quando l'utente è soddisfatto dell'attuale insieme di System extractions, può esportarlo in un nuovo documento cliccando il pulsante Export.

Le System extractions verranno salvate in un nuovo documento di testo, una per riga. Verrà aperta inoltre una finestra modale contenente un link al nuovo documento creato.

Figura 7: Esportazione delle system extractions attuali in un nuovo documento.

Cliccando il link si aprirà il documento con le System extractions.

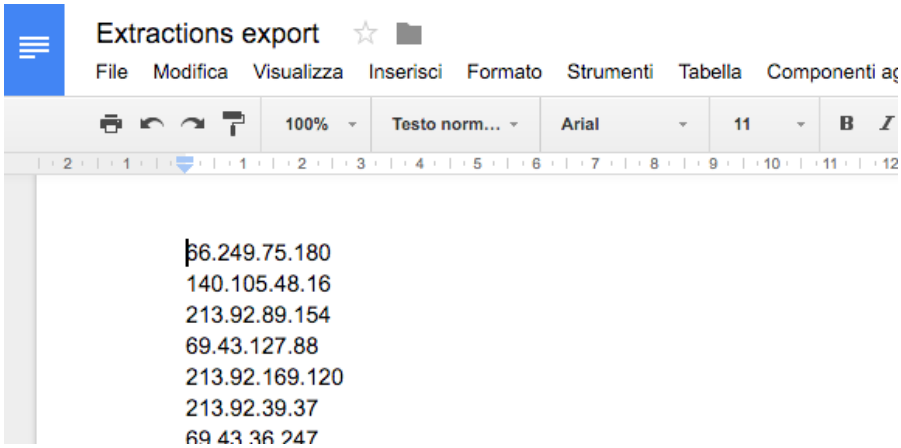

Figura 8: Il nuovo documento creato dall'esportazione delle system extractions, contenente una estrazione per riga.

2 NOTE REALIZZATIVE

Un componente aggiuntivo[1] di Google Docs è costituito da un insieme di file `.html` e `.gs`. I file `.gs` sono script scritti nel linguaggio *Google Apps Script*[2], il quale è un linguaggio di scripting basato su Javascript che viene eseguito nei server di Google.

L'Add-on realizzato è composto dai seguenti file:

Nome file	Righe
EntityExtractor.gs	561
Main.gs	507
Set.gs	54
Sidebar.css.html	67
Sidebar.html	61
Sidebar.js.html	322
Store.gs	53
TextRange.gs	34

Tabella 1: Struttura dei file del progetto.

2.1 STRUTTURE DATI

I dati che rappresentano lo stato dell'Add-on sono salvati lato server, in modo da sopravvivere alla chiusura e riapertura dello stesso.

Le liste di desired extractions e desired unextractions sono rappresentate ognuna come `Array` Javascript di oggetti `TextRange`, la quale definizione è contenuta in `TextRange.gs`, e vengono memorizzate dal server.

La lista di system extractions è rappresentata come un `Array` di oggetti `TextRangeWithScore`, la quale definizione si trova in `TextRangeWithScore.gs`, e non viene memorizzata dal server ma calcolata al momento del bisogno. Lo stesso vale per la query che è un oggetto di tipo `TextRangeWithScore`. Si è scelto di adottare questa soluzione dal momento che l'esecuzione dell'algoritmo non è onerosa dal punto di vista computazionale, e Google impone dei limiti restrittivi in merito allo spazio di memorizzazione disponibile lato server.

Le strutture dati di supporto all'algoritmo durante la sua esecuzione non sono memorizzate in modo permanente ma vengono create e distrutte ogni volta che esso viene eseguito. Tra le strutture usate è presente anche `Set`, definita in `Set.gs`, che rappresenta un generico insieme di elementi.

Sono salvate lato server anche le preferenze di evidenziazione dei tre tipi di annotazioni.

2.2 LATO CLIENT

2.2.1 Sidebar.html

Questo file è il sorgente della Sidebar che viene aperta all'apertura dell'Add-on, e ne rappresenta di conseguenza la struttura. La Sidebar è il meccanismo di interazione dell'utente con il componente aggiuntivo.

Il codice è scritto nel linguaggio HTML5, e viene servito dai server di Google all'interno di un Iframe.

È stato utilizzato come foglio di stile quello suggerito dalle linee guida di Google, allo scopo di di mantenere uno stile uniforme agli altri Add-on.

Grazie al meccanismo del templating[3] è possibile scrivere codice Apps Script all'interno delle pagine HTML; questa funzione è stata usata per includere i file contenenti il codice Javascript e il codice CSS della Sidebar.

2.2.2 Sidebar.css.html

Contiene la veste grafica della sidebar. È scritto nel linguaggio CSS (Cascading Style Sheets).

2.2.3 Sidebar.js.html

Rappresenta il codice Javascript eseguito nel browser dell'utente. È stata usata la libreria jQuery[4], come suggerito dalle linee guida di Google, che ha lo scopo di semplificare la manipolazione e la gestione degli eventi delle pagine HTML.

La prima istruzione che viene eseguita all'apertura della Sidebar corrisponde alla funzione `restoreSavedStatus()`, la quale si occupa di recuperare dal server lo stato dell'Add-on e di reimpostare la Sidebar in base allo stato recuperato.

Successivamente vengono registrati i gestori degli eventi corrispondenti al click dei pulsanti e delle checkbox.

Per far eseguire una determinata funzione al server viene usato il metodo `google.script.run.nomeFunzione()`, il quale comprende la possibilità dei registrare dei callback in caso di successo e fallimento dell'esecuzione, rispettivamente con i metodi `withSuccessHandler(callback)` e `withFailureHandler(callback)`.

In caso di fallimento dell'esecuzione viene usato come callback il metodo `showError()`, che mostra un messaggio di errore in cima alla Sidebar in rosso.

2.3 LATO SERVER

2.3.1 TextRange.gs

Contiene un'implementazione di un oggetto che rappresenta una porzione di testo in un documento, con un offset iniziale ed uno finale. Viene inoltre salvato anche il testo corrispondente in modo da agevolarne la manipolazione.

È presente una proprietà dell'oggetto TextRange, la funzione `areOverlapping` che permette di verificare se due intervalli di testo si sovrappongono.

È presente inoltre l'implementazione dell'oggetto che rappresenta un intervallo di testo con un determinato punteggio (score). Questo è usato per rappresentare i token dopo la classificazione da parte dell'algoritmo, il quale assegna ad ognuno di essi un punteggio (positivo per indicare un'estrazione, negativo per indicare una non-estrazione). È possibile creare un nuovo oggetto `TextRangeWithScore` sia a partire da un oggetto `TextRange` preesistente che dai suoi attributi.

2.3.2 Set.gs

Contiene l'implementazione di un oggetto `Set`, ovvero di un insieme. Gli elementi di un insieme non possono comparire più di una volta e l'insieme non ha un ordine intrinseco.

Sono presenti i metodi per calcolare l'intersezione e l'unione di due insiemi.

2.3.3 Store.gs

Questo file contiene l'oggetto `Store`, che fa da tramite per le proprietà dell'utente[5].

La memorizzazione di dati lato server negli Add-on avviene attraverso un database di tipo key-value. Dal momento che si è reso necessario il salvataggio di dati strutturati che non sono semplici stringhe di testo, è stato necessario implementare questo livello di astrazione.

Il salvataggio e il recupero di oggetti strutturati avviene tramite la loro conversione in formato JSON (JavaScript Object Notation). Prima di salvare una variabile nel database il suo valore viene convertito in JSON, e il contrario avviene quando essa viene richiesta.

È stato implementato inoltre un metodo `pushElementToArray` dal momento che a causa della natura del servizio di memorizzazione non è possibile usare la classica sintassi `array.push(element)`.

2.3.4 EntityExtractor.gs

Contiene l'implementazione dell'algoritmo di estrazione delle entità.

L'algoritmo di classificazione implementato è la variante SJ-MC, ovvero *classificatore basato sulla similarità di Jaccard con varianti multitoken e context-aware*.

L'estrattore viene inizializzato fornendo una stringa di base, una lista di desired extractions ed una lista di desired unextractions.

La proprietà `baseString` rappresenta la stringa alla quale si riferiscono le annotazioni. La proprietà `n_context` rappresenta il numero di caratteri da considerare intorno ad una porzione di testo per la variante context-aware. La proprietà `n_validation` è un numero usato in fase di valutazione dell'informatività legata ad una coppia di insiemi (P, N). La proprietà `jaccardMatrix` è un array usato come cache per il calcolo degli indici di similarità di Jaccard. Le proprietà `desiredExtractions` e `desiredUnextractions` rappresentano rispettivamente gli insiemi X^m ed X^u.

Il metodo che effettua l'apprendimento e l'estrazione delle entità è `run()`. Esso esegue le seguenti azioni:

1. Genera il set ottimale di separatori S. Per fare questo:

   a) Viene generato il set base di separatori S_0 composto da ogni carattere precedente e seguente le desired extractions;
   b) Si costruisce una lista comprendente i caratteri di S_0 ordinati per numero di occorrenze in modo decrescente;
   c) Si itera per un numero di volte pari alla lunghezza della lista, prendendo ogni volta i primi i elementi dalla stessa e contando il numero di token che corrisponderebbero a desired extractions effettuando la tokenizzazione con la lista di separatori attuale;
   d) La lista di separatori che corrisponde al numero maggiore di corrispondenze è quella ottimale.

2. Si calcola la costante n_{tokens}, che corrisponde al numero massimo di token in cui un elemento di X^m sarebbe diviso usando l'insieme di separatori ottimale. Questo valore è usato nella variante multitoken dell'algoritmo.

3. Si genera il multiset P che corrisponde a X^m. Si generano poi i multiset P^{before} e P^{after} composti dai caratteri precedenti e seguenti le sottostringhe in P.

4. Si genera il multiset N, che corrisponde alla sottostringhe in X^u tokenizzate ognuna con l'insieme dei separatori ottimale S. Si generano poi i multiset N^{before} e N^{after}.

5. Si generano i valori di soglia $\tau, \tau_{before}, \tau_{after}$ allo scopo di normalizzare il punteggio che successivamente viene assegnato ad ogni token, in modo che i token con punteggio positivo rappresentino un'estrazione e quelli con punteggio negativo rappresentino una non-estrazione.

6. Si calcolano i valori di accuratezza a, a^{before}, a^{after}. Questi valori rappresentano l'informatività legata ad una coppia di multiset (P, N). Se l'accuratezza è minore di $0,5$ essa viene impostata a 0 in modo da non tenere conto della coppia di multiset relativa.

7. Si effettua la classificazione vera e propria.

   La stringa di base viene tokenizzata usando il set di separatori ottimale, e se il valore `n_tokens` è maggiore o uguale a 2, vengono considerate anche le sequenze di n_{tokens} token consecutivi. Per ogni token viene calcolato un valore Δ_m rispetto ad ognuna delle tre coppie di multiset $(P, N), (P^{before}, N^{before}), (P^{after}, N^{after})$ ed il punteggio del token corrisponde alla media pesata dei tre valori, con peso pari all'accuratezza della coppia di multiset relativa.

Eseguito l'algoritmo, per ottenere le system extractions si utilizza il metodo `getSystemExtractions()` il quale restituisce una lista con i token con punteggio positivo.

Per ottenere la query si utilizza invece il metodo `getQuery()` il quale, avendo implementato lo schema di active learning *Uncertainty sampling*, restituisce il token con punteggio assegnato più piccolo in valore assoluto.

2.3.5 Main.gs

È il file principale contenente le funzioni più importanti eseguite dai server di Google.

- I trigger `onInstall()` ed `onOpen()` vengono eseguiti rispettivamente all'installazione del componente aggiuntivo ed alla sua apertura.
- La funzione `localOffsetToGlobal()` serve a trasformare una coppia (paragrafo, offset) in un offset globale relativo al documento. Questo è necessario perchè l'offset relativo ad una selezione fornito da Google si riferisce sempre all'elemento che contiene quella porzione di testo, ovvero al suo paragrafo.
- La funzione `isAnnotationAllowed()` controlla se una nuova annotazione si sovrappone con qualcuna delle esistenti. Due annotazioni non si possono mai sovrapporre.
- La funzione `resetBackground()` reimposta lo sfondo del documento rimuovendo qualsiasi evidenziatura.
- La funzione `reDrawHighlights(extractionResult)` ridisegna l'evidenziatura delle annotazioni, nell'ordine corretto. Se l'evidenziatura di una tipologia di annotazioni è disabilitata non la disegna.
- La funzione `deleteAnnotation(startOffset)` cancella un'annotazione. L'offset di inizio identifica univocamente l'annotazione dal momento che esse non si possono sovrapporre.
- La funzione `getExtractionResult()` esegue l'algoritmo di estrazione e restituisce la lista delle System extractions e la Query. Vengono filtrate le System extractions che si sovrappongono con una Desired extraction o Desired unextraction.
- La funzione `exportExtractions()` esporta le attuali System extractions in un nuovo documento di testo, una per riga. Successivamente apre una finestra modale con un link al nuovo documento creato. Il documento viene creato nella root di Google Drive dell'utente.
- La funzione `selectAnnotation()` seleziona una porzione del testo del documento, ed è usata per individuare le annotazioni.
- La funzione `setHighlightStatus()` permette l'attivazione o disattivazione dell'evidenziatura per una data categoria di annotazioni.

BIBLIOGRAFIA

[1] Google. *Develop Add-ons for Google Sheets, Docs, and Forms*. 2015. URL: https://developers.google.com/apps-script/add-ons/ (visitato il 08/04/2016) (cit. a p. 7).

[2] Google. *Overview of Google Apps Script*. 2016. URL: https://developers.google.com/apps-script/overview (visitato il 08/04/2016) (cit. a p. 7).

[3] Google. *HTML Service: Templated HTML*. 2015. URL: https://developers.google.com/apps-script/guides/html/templates (visitato il 08/04/2016) (cit. a p. 8).

[4] The jQuery Foundation. *jQuery*. 2016. URL: https://jquery.com/ (visitato il 08/04/2016) (cit. a p. 8).

[5] Google. *Properties Service*. 2015. URL: https://developers.google.com/apps-script/guides/properties (visitato il 08/04/2016) (cit. a p. 9).