Web and Social Information Extraction

Relazione progetto 2016: US Presidentials

A.A. 2015/2016

Sommario

Introduzione

In questa relazione vengono riportate tutte le fasi e i risultati ottenuti durante lo studio dei dati contenuti nei due dataset assegnati, e delle informazioni ricavate durante l'implementazione dei task richiesti. La relazione è stata suddivisa in vari capitoli, ognuno dei quali tratta in maniera specifica i diversi task assegnati.

I candidati scelti per gli sviluppi dei vari punti sono stati 4, in particolare :
-Hillary Clinton
-Donald Trump
-Mark Rubio
-Bernie Sanders

Prima fase : Identify mentions of candidates

In questa prima fase è stata svolta un'analisi sulla tipologia di dato contenuto nei due dataset, successivamente è stato impostato il lavoro in maniera tale da avere del codice scalabile da poter riutilizzare nei task successivi.

Analisi dei dataset

Partendo dai dati contenuti nei dataset è stato costruito un indice tramite "Lucene" comprensivo di tutte le informazioni utili per lo svolgimento dei task, è stato deciso di creare un *Document* di tipo Lucene per ogni tweet contenuto nei dataset.
Nel dettaglio, ogni documento creato da Lucene possiede al suo interno queste informazioni:

- "idTweet", id del tweet
- "languageTweet", lingua del tweet
- "tweetUser", username dell'utente che ha scritto il tweet
- "tweetUserId", id dell'utente che ha scritto il tweet
- "tweetText", testo del tweet

L'indice creato è stato salvato in locale per questioni di performance, evitando così di dover ripetere la creazione dell'indice(molto onerosa) ogni qualvolta si aveva necessità di eseguire le varie funzioni.

Per l'estrazione delle varie informazioni dai file contenuti nei dataset è stata utilizzata la libreria Jsoup. Dopo aver individuato gli opportuni tag delimitatori, è stato

implementato un metodo per l'estrazione dei dati tramite la funzione "parse" messa a disposizione da Jsoup.
Dopo aver creato l'indice sono state effettuate delle query tramite le varie api di Lucene per trovare i TopDocuments relativi ai candidati presi in considerazione, i documenti relativi ai vari candidati sono stati ricercati attraverso il contenuto del campo "tweetText".
E' stata definita una query contenente i nomi dei vari candidati ed è stata passata in input alla funzione "search" di Lucene.
I top documents trovati sono i soli a contenere menzioni sui candidati presi in esame, tali informazioni sono state riportate in un file di testo contenente i dati relativi a:
-numero di documenti Lucene trovati tramite la query
-numero di utenti che hanno menzionato i vari candidati
-numero di tweet contenenti almeno una menzione
-lista degli utenti, con relativi tweet.

I documenti trovati tramite la query hanno dato luogo ad un sottoinsieme del dataset iniziale ed è stato definito T(M).

Criteri per il restringimento del dataset

Dopo aver condotto la prima parte dello studio sul dataset ci si è resi conto della vastità dei dati a disposizione. Ai fini progettuali e statistici però si è convenuti che il numero totale degli utenti su cui effettuare i vari task doveva essere sensibilmente inferiore.
Per attuare un restringimento del dataset che non fosse semplicemente casuale, si è pensato di basare tale restringimento su una serie di criteri.
Dopo una piccola analisi è stato notato che molti utenti utilizzatori di Twitter possiedono un numero molto basso di followers e un numero ancora più esiguo di tweet pubblicati, per ottenere dei dati abbastanza significati si è deciso di applicare alcuni filtri agli utenti nel dataset. I filtri utilizzati sono stati:

1. Lingua (en)
2. Località (USA & Canada)
3. Numero di followers (> 1500)
4. Numero totale di tweet pubblicati (>1500)

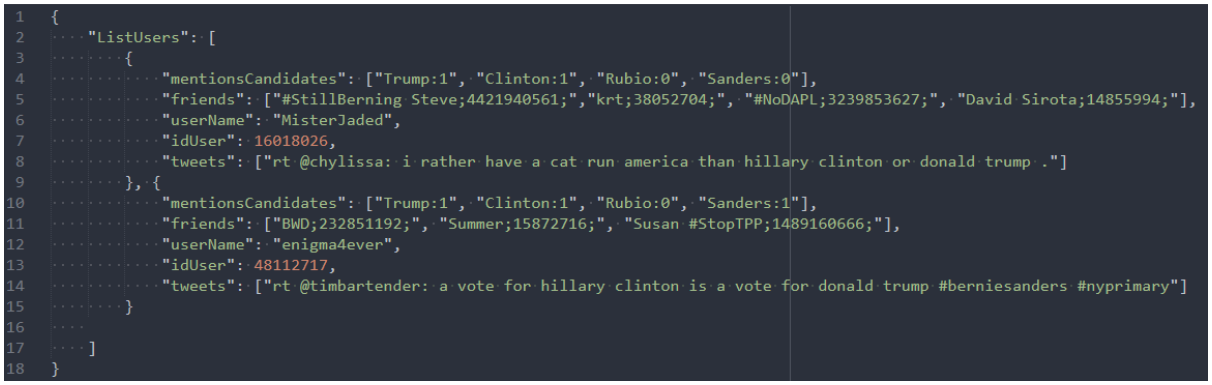
Per quanto riguarda lingua e località si è pensato di dare più risalto ad utenti provenienti dalla zona di svolgimento delle elezioni.
Il filtro sui followers è stato applicato poiché utenti con follower più numerosi equivale ad utenti più seguiti e quindi con più peso specifico, stesso discorso per il numero totale di tweet, un numero totale di tweet molto alto può infatti notare una grande attività sul social network.
Dopo aver applicato i vari filtri si è ottenuto un numero comunque ancora troppo grande di utenti e sono stati selezionati tra gli utenti trovati un sottoinsieme di 2000 utenti.

Fase di crawling

Per immagazzinare tutti i dati necessari allo svolgimento di questo step e per facilitare i successivi, è stato scelto di utilizzare un documento Json contenente tutte le informazioni su cui lavorare. La scelta è ricaduta su questo tipo di documento per i vari vantaggi portati dal suo utilizzo, tra cui le performance, la facilità di aggiornamento dei vari oggetti contenuti in esso e la possibilità di creare oggetti complessi per contenere tutti i dati utili alle varie operazioni da effettuare.(\files\Utenti\friendship\userFriendshipJsonComplete.json)
Il json creato per lo svolgimento del progetto è così composto:

L'array "*ListUsers*" contiene una lista di oggetti, ogni oggetto corrisponde ad un utente, che possiede vari campi, tra cui:

- username,
- id
- un array contenente i tweet scritti dall'utente sui vari candidati
- un array contenente gli amici di quel determinato utente
- un array contenente l'informazione sul numero delle menzioni dell'utente per ogni candidato.



```
{
    "ListUsers": [
        {
            "mentionsCandidates": ["Trump:1", "Clinton:1", "Rubio:0", "Sanders:0"],
            "friends": ["#StillBerning Steve;4421940561;","krt;38052704;", "#NoDAPL;3239853627;", "David Sirota;14855994;"],
            "userName": "MisterJaded",
            "idUser": 16018026,
            "tweets": ["rt @chylissa: i rather have a cat run america than hillary clinton or donald trump ."]
        }, {
            "mentionsCandidates": ["Trump:1", "Clinton:1", "Rubio:0", "Sanders:1"],
            "friends": ["BWD;232851192;", "Summer;15872716;", "Susan #StopTPP;1489160666;"],
            "userName": "enigma4ever",
            "idUser": 48112717,
            "tweets": ["rt @timbartender: a vote for hillary clinton is a vote for donald trump #berniesanders #nyprimary"]
        }

    ]
}
```

Creazione del grafo

Al fine di raggiungere gli obiettivi del progetto e la relativa analisi, abbiamo utilizzato un grafo diretto di tipo *ListenableDirectedGraph<String, DefaultEdge>.*
Analizzando ogni utente abbiamo:

- creato un nodo del tipo "*username* + *userId*"
- creato un arco tra ogni coppia di nodi a-b nel caso in cui a avesse b tra la lista di amici

Identificazione della componente connessa

Per l'identificazione della componente connessa,è stato creato un algoritmo ricorsivo per estrapolare, tramite le api messe a disposizione da twitter, la lista di amici per ogni utente appartenente al sottoinsieme scelto. Per ognuno di questi utenti viene estrapolata una lista contenente gli amici, contestualmente si va a verificare se nella lista di amici estrapolata vi è traccia di utenti presenti nel sottoinsieme dei 2000 scelti in precedenza. La scelta di ricercare e tenere traccia solamente delle relazioni di amicizia tra i 2000 utenti è stata pensata per avere un dataset più completo, e più congruente. Inserire legami di amicizia fittizzi e/o senza un riscontro nel sottoinsieme scelto per effettuare le analisi avrebbe portato a dei dati non completamente precisi.
Dopo aver aggiunto per ogni utente del sottoinsieme le amicizie presenti con gli altri utenti del sottoinsieme stesso si è proceduto alla creazione di un grafo diretto.
Ogni utente rappresenta un nodo del grafo e ogni arco rappresenta una relazione di amicizia tra gli utenti del sottoinsieme.
Per il calcolo delle componenti connesse è stato utilizzato il metodo *connectedSetOf(currentVertex)* messo a disposizione da *jGrapht*, sostanzialmente ciclando su ogni nodo del grafo creato in precedenza si va a richiamare questo metodo, viene calcolata così la componente connessa del nodo corrente. Vengono salvate tutte le componenti connesse in un file di log e alla fine viene calcolata la componente connessa più grande, tenendo traccia del nodo di partenza, del numero di nodi della componente e degli elementi stessi della componente.

Calcolo del pageRank e Centrality

A partire dal grafo creato in precedenza è stato implementato un metodo per trasformare il grafo diretto in un grafo accettato dal metodo per il calcolo del PageRank, è stato utilizzato un metodo messo a disposizione dalla libreria Jung, che preso in input un nodo ritorna un valore double corrispondente allo score di quel nodo. Tutti gli score di ogni nodo sono salvati su un file di log apposito. (\files\Utenti\log4j_logs\pageRankScore.log)
Per quanto riguarda il calcolo della centrality è stato richiesto di dividere gli utenti in M parti, tante quanti sono i candidati presi in considerazione, successivamente di individuare gli utenti che hanno menzionato più volte i vari candidati e sui primi 10 utenti che hanno fatto più menzioni andare a calcolare la loro centralità.
E' stato implementato un metodo per calcolare quante menzioni e per quale candidato sono state fatte da ogni utente dell'insieme preso in considerazione, è stato cosi aggiornato il json con le informazioni relative al numero di menzioni per ogni candidato aggiungendo un jsonArray con questo formato:

```
"mentionsCandidates": ["Trump:1", "Clinton:1", "Rubio:0", "Sanders:0"],
```

grazie alle informazioni salvate con questo formato è stato possibile, tramite un altro metodo, scorrere tutti gli oggetti presenti nel file Json e dividere i vari utenti in M hashMap

ordinate in base al numero di menzioni effettuate.
Per il calcolo della centralità è stata utilizzata un'apposita struttura dati messa a disposizione dalla libreria Jung (ClosenessCentrality<String,DefaultEdge>), su ogni elemento è stato richiamato il metodo che calcola lo score. I valori relativi alla centrality di ogni nodo sono stati ordinati e salvati in un apposito file di log. (\files\Utenti\log4j_logs\centralityUserMentionedCand.log)
Una volta ottenuti i valori della centralità per ogni nodo e i valori relativi alle menzioni di ogni utente per ogni candidato è stato possibile trovare i primi 10 utenti con un valore bilanciato tra numero di menzioni / valore della centrality.

Seconda fase: Supporters and Opponents

Calcolo del sentiment con SentiwordNet

Per l'implementazione di tale funzionalitá é stata utilizzata la nota libreria SentiWordnet. Partendo dai documenti creati in precedenza grazie a Lucene, è stato estratto il testo dei tweet per ogni candidato e successivamente analizzato i singoli termini.

A tale scopo sono state essenziali alcune operazioni:

- eliminazione dei termini non utili ai fine dell'analisi (articoli, congiunzioni, etc…)
- gestione delle negazioni. (isn't **good** president) in tal caso non basta esaminare la parola "good", certamente di carattere positivo.

di seguito il metodo utilizzato in fase di decisione (supporter, opponent, neutral) di un tweet per un determinato candidato:

TOTALSentiment(Tweet) = *sentimentW(parola1)* + …… + *sentimentW(parolaN)*

con valore di segno opposto nel caso di negativitá.
Il risultato derivante da tale somma ha portato alla decisione del sentiment dell'intera frase rispetto a determinate soglie.

Per far fronte ad uno spreco di valutazione nei confronti di termini "innocui" o comunque non utili, abbiamo utilizzato una lista di termini noti, il nostro algoritmo ripulisce quindi la stringa di un determinato tweet verificando la presenza o meno di tali termini.

Successivamente, è stata fondamentale la gestione delle negazioni, nello sviluppo dell'algoritmo che gestisse tale punto, ci siamo ispirati ad uno studio effettuato dal prof. Christopher Potts (professore di linguistica a Stanford), di seguito l'idea:

ci siamo serviti di 2 importanti funzioni nell'analisi di una frase:

1. controllo effettivo di una parola con senso di negazione (not, insn't, etc...)
2. controllo di punteggiatura (per la gestione dell'inizio di una nuova frase)

Prendiamo come esempio questa frase:
"I don't think I can: I'm too tired."

sostanzialmente l'algoritmo riconosce la presenza di una negazione, le parole successive ad essa sono contrassegnate in maniera da riconoscerle come "appartenenti ad una negazione", inoltre é importante "resettare" memoria dell'incontro di una negazione nel caso in cui si trovino determinati caratteri di punteggiatura ,il contrassegno utilizzato nel nostro algortimo è "_NEG" vediamo il risultato nella frase di esempio:

i
don't
think_NEG
I_NEG
can_NEG
i'm
too
tired

a questo punto la valutazione restituita da sentimentWordNet per le 3 parole con senso negativo avranno valore negativo.
Dallo studio effettuato puó certamente risultare interessante arricchire ulteriormente la base di dati utilizzata nell'algoritmo di opinion mining.

Terza fase: Co-occurrence analisys

Partendo dal dataset globale contenente tutte le menzioni dei candidati è stato deciso di creare un dataset di appoggio per ogni candidato tramite Lucene.
E' stato implementato un algoritmo che crea un indice di appoggio tramite Lucene per ogni candidato e viene "svuotato" per far spazio all'indice del candidato successivo durante l'esecuzione dei vari metodi relativi al task dell'analisi delle occorrenze.

Per calcolare l'indice di Jaccard sulle varie coppie di parole sono stati eseguiti i seguenti passi:

1. Creazione dell'indice di Lucene per il candidato
2. Calcolo della doc frequency dei termini presenti nei tweet per il candidato analizzato
3. Ricerca tramite Lucene per ricercare le co-occorrenze delle coppie di termini trovati nel punto precedente e calcolo della loro doc frequency
4. Calcolo dell'indice di Jaccard

Calcolo delle frequenze dei termini

Per ogni candidato è stato ricavato un indice Lucene contenente tutti i tweet in cui compaiono menzioni per il candidato analizzato, su questo specifico dataset sono stati estrapolati i termini contenuti nei tweet e creati dei vettori di "terms".
Tramite un algoritmo è stata ricavata una Map contenente come Key il termine analizzato e come Value la term-frequency del termine in tutto l'indice Lucene per quel candidato.

Per quanto riguarda la doc frequency delle coppie di termini è stato implementato un algoritmo che prendendo in input una coppia di termini va ad effettuare una query Lucene sull'indice del candidato che dà evidenza di quanti documenti dell'index contengono la coppia di termini ricercata. Il metodo ritorna in output una mappa contenente come Key la coppia di termini analizzati e come Value il numero di documenti dell'indice in cui compaiono tali termini insieme.
A questo punto avendo una mappa contenente le term frequency di tutti i termini e

un’altra mappa contenente le doc frequency di tutte le coppie di termini è stato possibile implementare un ulteriore algoritmo per il calcolo dell’indice di Jaccard.

Calcolo dell’indice di Jaccard

Per il calcolo di Jaccard è stata usata la seguente formula:

```
Double jaccardIndex = docFreqTerm1Term2/(((countTerm1 + countTerm2) - docFreqTerm1Term2));
```

in notazione insiemistica:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

come si vede dalla porzione di codice la formula utilizza il valore calcolato della doc frequency dei due termini nei vari documenti creati con Lucene e la term frequency dei singoli termini sempre calcolata sullo stesso dataset di documenti Lucene.

Quarta Fase: Scrape news about one candidate

Fase di scraping

Per la fase di scraping è stata utilizzata la libreria **Jsoup**, è stata effettuata una ricerca su google news per il candidato Hillary Clinton, tramite l'editor messo a disposizione dal browser Google Chrome è stata effettuata un'analisi sulla pagina Html, in particolare sui tag utilizzati per contenere titolo e corpo delle varie news. E' stato implementato un algoritmo che tramite jsoup estrae la parte di html necessaria, e su essa effettua un ciclo sui tag delle news, in questo modo è stato possibile estrarre il titolo e il corpo delle varie news per salvarle in una struttura dati apposita, contestualmente queste informazioni sono state anche salvate in un file Json costruito appositamente per accogliere una lista di jsonObject che rappresentano una news, con i campi title, e body.
Dato che le news in una pagina di google news vanno da un minimo di 10 ad un massimo di 100 per pagina(modificabile da impostazioni del browser) si è scelto di estrapolare le prime 300 news riguardante il candidato prescelto.
Una volta completato lo scraping e salvate tutte le informazioni in una struttura dati è stato possibile applicare l'algoritmo per estrapolare le coppie di parole che co-occorrono più spesso nelle varie news.

A questo punto è stato possibile applicare un algoritmo per l'estrazione delle coppie di parole più utilizzate tra quelle estrapolate dal dataset di tweet e quelle estrapolate dalle news. (\files\Scraping\scrapingNews.json)

Quinta fase: Predict the winner

Per quanto riguarda la previsione del candidato vincitore delle presidenziali, riportiamo di seguito l'idea alla base del nostro algoritmo:

1. Calcolo del sentiment sui tweets degli utenti del dataset.
2. Individuazione degli utenti piú influenti (primi 100 con page rank più elevato e primi 100 con centrality più alta per ogni candidato) è successiva aggiunta/decremento di un peso pari a 0.3 sul valore del sentiment.
3. Calcolo del sentiment sulle coppie di parole che co-occorrono maggiormente(Jaccard).

4. Somma dei valori trovati in precedenza per ogni candidato, come possibile vedere nello snapshot.

```
1  double totalTrump = sentimentValueTrump + sentimentJaccardTrump;
2  double totalClinton = sentimentValueClinton + sentimentJaccardClinton;
```

Il candidato con il valore totale maggiore è quello che secondo questo algoritmo avrebbe dovuto vincere le presidenziali.

Risultati ottenuti

In questo paragrafo sono sintetizzati i risultati ottenuti per i vari punti delle specifiche di progetto.

1) Identify mentions of candidates

1. Candidati presi in considerazione : Donald Trump, Hillary Clinton, Bernie Sanders, Mark Rubio.

NUMERO UTENTI = 42582 = M
NUMERO TOTALE TWEETS PER I CANDIDATI = 67721 = T(M)

2. La componente connessa più grande è stata identificata partendo dal nodo/utente : MisterJaded (Presente nel file “\files\Utenti\log4j_logs\connectedComponents.log”)

I primi 10 utenti in base al page rank sono :

Pos.	Username	UserId	Score page rank
1	Zeke Miller	21316253	0.032893277407031254
2	BuzzFeed	5695632	0.029127880458282983
3	Good Morning America	22650211	0.02641614030499407
4	Caleb Howe	14964429	0.015271424586850074
5	WTVG 13abc	17787008	0.011359151567170291
6	Jonathan Capehart	97474887	0.008822752974749902
7	Patrick Ruffini	1050111	0.008787775182669225
8	Independent Journal	474464413	0.008336852280178893
9	Beth Reinhard	34022952	0.007798165069277863
10	POLITICO	9300262	0.006289794797201593

3. Per ognuno dei 4 candidati sono indicati i primi 10 utenti che hanno centrality più alta e maggior numero di menzioni per quel candidato:

Primi 10 utenti per Trump

Pos	Username	User id	N° menzioni	Centrality
1	USDeposeUsurperØbama	874859443	8	0.45749530705282915
2	Richard Oberlander	21091983	8	0.44151138716356103
3	RIPManny~CAT~ex-Rep	902596597	8	0.4369877049180328
4	Christie	1618641848	8	0.5562438865340724
5	Jason's Grandpa	23047147	7	0.4602104127326679

6	FloridaTrumpPence	3192848121	7	0.43252595155709345
7	Deplorable Marine	28534205	7	0.44326241134751776
8	DeplorableJanMoss	4821375853	7	0.4313527180783818
9	dr max tparty	34701889	7	0.4469478648153
10	Trump In It To Win	443304640	7	0.484659090909091

Primi 10 utenti per Clinton

Pos	Username	User id	N° menzioni	Centrality
1	Daniel John Sobieski	2188423838	9	0.42918238993710695
2	Marcy McGowan	16146535	9	0.4346496815286624
3	Mrs. Kelly Jacobs	390103810	8	0.44003095176683005
4	DEBI129	16444544	7	0.4384476998200977
5	(((TRUTH_TWEETERS)))	616623398	6	0.43069931835395103
6	Emil	109854452	6	0.44048541182545825
7	Jasper	519728191	5	0.42724768344603054
8	TheSentinel	47680487	5	0.4309744779582367
9	LG4LG	905374200	5	0.44473409801876956
10	CO2HOG™	28885875	5	0.4583557227297152

Primi 10 utenti per Sanders

Pos	Username	User id	N° menzioni	Centrality
1	TheSentinel	47680487	8	0.4309744779582367
2	#NoDAPL	3239853627	7	0.44601307189542483
3	GodandtheBear	3230944180	7	0.4293984394663982
4	Emil	109854452	7	0.44048541182545825
5	Marcy McGowan	16146535	7	0.4346496815286624

6	Joyce Jeffries	73333278	7	0.43498215196328405
7	Wake Up	4907495117	7	0.42575492887446964
8	Sharon Jones	792119786	6	0.43354510800508256
9	LeslieRMyers	3920426061	6	0.43037336024217965
10	DogLovers4Hillary	18220818	6	0.42682011508631473

Primi 10 utenti per Rubio

Pos	Username	User id	N° menzioni	Centrality
1	lawalker	129905825	1	0.44059917355371897
2	SkyCast	3168191809	1	0.45578413037670323
3	Dorothy Schomburg	993358020	1	0.4382224505522733
4	Proud American	2417844248	1	0.45930232558139533
5	Christie	1618641848	1	0.5562438865340724
6	Carol Boothe	67503749	1	0.4362055740219893
7	Trump for President	44382195	1	0.4482396216500263
8	David Jones	626503046	1	0.48083427282976327
9	Anna L. Morris	582927693	1	0.42857142857142855
10	Jesse James Dick Jr	86378821	1	0.44601307189542483

2) Supporters and Opponents

Per ognuno dei quattro candidati è stato calcolato il sentiment per i tweet pubblicati dai 10 utenti del sottoinsieme M', ed è stato definito se tali utenti siano Supporter, Opponent o Neutral.

CANDIDATO - TRUMP

1. USDeposeUsurperØbama is a **Supporter**!
2. Richard Oberlander is **Neutral**!
3. RIPManny~CAT~ex-Rep is **Neutral**!
4. Christie is **Neutral**!
5. Jason's Grandpa is **Neutral**!
6. FloridaTrumpPence is **Neutral**!
7. Deplorable Marine is **Neutral**!
8. DeplorableJanMoss is **Neutral**!
9. dr max tparty is **Neutral**!
10. Trump In It To Win is an **Opponent**!

CANDIDATO - CLINTON

1. Daniel John Sobieski is an **Opponent**!
2. Marcy McGowan is an **Opponent**!
3. Mrs. Kelly Jacobs is a **Supporter**!
4. DEBI129 is **Neutral**!
5. (((TRUTH_TWEETERS))) is **Neutral**!
6. Emil is a **Supporter**!
7. Jasper is **Neutral**!
8. TheSentinel is **Neutral**!
9. LG4LG is an **Opponent**!
10. CO2HOG™ is a **Supporter**!

CANDIDATO - SANDERS

1. TheSentinel is an **Opponent**!
2. #NoDAPL is **Neutral**!
3. GodandtheBear is a **Supporter**!
4. Emil is **Neutral**!
5. Marcy McGowan is a **Supporter**!
6. Joyce Jeffries is **Neutral**!
7. Wake Up is an **Opponent**!
8. Sharon Jones is an **Opponent**!
9. LeslieRMyers is **Neutral**!
10. DogLovers4Hillary is an **Opponent**!

CANDIDATO - RUBIO

1. lawalker is an **Opponent**!
2. SkyCast is an **Opponent**!
3. Dorothy Schomburg is a **Supporter**!
4. Proud American is an **Opponent**!
5. Christie is **Neutral**!
6. Carol Boothe is an **Opponent**!
7. Trump for President is **Neutral**!
8. David Jones is an **Opponent**!
9. Anna L. Morris is an **Opponent**!
10. Jesse James Dick Jr is a **Supporter**!

3)Co-occurrence analisys

Per ogni candidato sono state calcolate le coppie di parole che co-occorrono maggiormente e calcolato l'indice di Jaccard su queste coppie. Il risultato ottenuto per ogni candidato è stato salvato in quattro file json separati. (accessibili in \files\Jaccard\NOMECANDIDATO_jaccard.json)
Di seguito un estratto del file riguardante il candidato Donald Trump :


```
{"docFreqTerm1Term2":233.0,"docFreqTerm1":234.0,"docFreqTerm2":233.0,"jaccard":0.9957264957264957,"term1":"hansoniine","term2":"loest0aakx"}
{"docFreqTerm1Term2":168.0,"docFreqTerm1":173.0,"docFreqTerm2":179.0,"jaccard":0.9130434782608695,"term1":"salmahayek","term2":"dyslexic"}
{"docFreqTerm1Term2":176.0,"docFreqTerm1":195.0,"docFreqTerm2":179.0,"jaccard":0.8888888888888888,"term1":"confuse","term2":"dyslexic"}
{"docFreqTerm1Term2":900.0,"docFreqTerm1":989.0,"docFreqTerm2":925.0,"jaccard":0.8875739644970414,"term1":"coachella","term2":"chanting"}
{"docFreqTerm1Term2":563.0,"docFreqTerm1":630.0,"docFreqTerm2":583.0,"jaccard":0.8661538461538462,"term1":"ck","term2":"40,000"}
{"docFreqTerm1Term2":159.0,"docFreqTerm1":171.0,"docFreqTerm2":172.0,"jaccard":0.8641304347826086,"term1":"oscar","term2":"leo"}
{"docFreqTerm1Term2":1339.0,"docFreqTerm1":1351.0,"docFreqTerm2":1541.0,"jaccard":0.8622021893110109,"term1":"drove","term2":"past"}
{"docFreqTerm1Term2":221.0,"docFreqTerm1":245.0,"docFreqTerm2":235.0,"jaccard":0.8532818532818532,"term1":"instagram","term2":"ruling"}
{"docFreqTerm1Term2":169.0,"docFreqTerm1":195.0,"docFreqTerm2":173.0,"jaccard":0.8492462311557789,"term1":"confuse","term2":"salmahayek"}
{"docFreqTerm1Term2":319.0,"docFreqTerm1":376.0,"docFreqTerm2":320.0,"jaccard":0.8461538461538461,"term1":"games","term2":"hunger"}
{"docFreqTerm1Term2":168.0,"docFreqTerm1":173.0,"docFreqTerm2":198.0,"jaccard":0.8275862068965517,"term1":"salmahayek","term2":"english"}
{"docFreqTerm1Term2":168.0,"docFreqTerm1":173.0,"docFreqTerm2":199.0,"jaccard":0.8235294117647058,"term1":"salmahayek","term2":"however"}
{"docFreqTerm1Term2":1159.0,"docFreqTerm1":1351.0,"docFreqTerm2":1240.0,"jaccard":0.8093575418994413,"term1":"drove","term2":"played"}
{"docFreqTerm1Term2":168.0,"docFreqTerm1":179.0,"docFreqTerm2":198.0,"jaccard":0.8038277511961722,"term1":"dyslexic","term2":"english"}
{"docFreqTerm1Term2":175.0,"docFreqTerm1":203.0,"docFreqTerm2":190.0,"jaccard":0.8027522935779816,"term1":"animals","term2":"endangered"}
```


Come si può notare dal json risultante, per ogni riga è contenuta l'informazione riguardante i 2 termini, la doc frequency di entrambi i termini e la term frequency dei termini singoli ed infine il valore dell'indice di Jaccard calcolato sui due termini.

4)Scrape news about one candidate

Dopo aver effettuato lo scraping delle news da Google News, riguardanti il candidato Hillary Clinton è stata fatta un'analisi sui termini come per le main occurrence words.
Mettendo a confronto i dati ricavati dai tweet del dataset e le parole contenute nel corpo delle news estrapolate da Google News abbiamo notato un leggero inasprimento dei termini utilizzati nelle notizie di Google News, risalenti ad Ottobre.
Alcuni avvenimenti accaduti nei mesi intercorsi tra il periodo di riferimento dei tweet e delle news estrapolate da Google News hanno fatto sì che i termini e quindi anche il sentiment calcolato su quest'ultimo sia peggiorato rispetto a quello dei tweet.

5)Predict the winner

Il risultato finale è stato a favore di HILLARY CLINTON, ma da quanto visto alle presidenziali che si sono svolte nel mese di novembre, contrariamente a tutti i sondaggi e previsioni, il vincitore è stato Donald Trump. Riteniamo quindi la nostra previsione in linea con quella effettuata dai maggiori organi di informazione.

Per quanto riguarda il vincitore finale va fatta una considerazione:
Il dataset preso in esame contiene tweet risalenti ai mesi di Aprile e Maggio 2016, le analisi effettuate su tale dataset hanno evidenziato come nell'arco dei mesi il sentiment verso i vari candidati sia mutato. In particolare considerando il sentiment relativo alla candidata Hillary

Clinton si può notare come il valore del sentiment calcolato sui tweet sia cambiato in negativo rispetto quello calcolato sulle parole contenute nelle news estrapolate da google news, risalenti al mese di ottobre.

Considerazioni su alcuni strumenti utilizzati

Al fine di rendere il progetto facilmente utilizzabile ed avere inoltre una visione "modulare" dei task, abbiamo deciso di utilizzare il famoso logger **log4j** configurando nel file di properties l'insieme di file sui quali scrivere log per ognuna delle nostre esigenze.

Inoltre é stato utilizzato un ulteriore file di configurazione dei path relativi ai file di appoggio utilizzati nel salvataggio dei json creati.
Es: "*PATH_FILE_FRIENDSHIP_JSON = ../files/Utenti/friendship/userFriendshipJson.json*"

Piú in generale abbiamo deciso di utilizzare l'ormai consolidata notazione **JSON** (JavaScript Object Notation) cosí da ottenere una buona manipolazione degli oggetti, in particolar modo:

- gestione del grafo delle friendship
- occurences
- scraping