Report

Rocco Caruso

November 27, 2015

Chapter 1

Problem definition

Dato uno stream di tweet (ordinato temporalmente) l'obiettivo è quello di individuare dei "topic" o "event". In letteratura esistono due principali metodologie:

- document-pivot
- feature-pivot

In questo lavoro è stata adottata la prima metodologia, l'obiettivo è quindi suddividere lo stream in cluster, tali che ciascun cluster corrisponda a tutti i tweet relativi ad un "evento ". E' altresì necessario, una volta suddiviso lo stream in vari cluster, distinguere quali sono realmente eventi (flashmob) da quelli che non lo sono, per mezzo di un classificatore.
La definizione di evento utilizzata è quella utilizzata per il Topic Detection and Tracking (TDT) [All02] :

Definizione 1. *un evento è qualcosa che accade in un luogo specifico in un tempo specifico*

L'intero processo può essere suddiviso in 5 attività come descritto in figura 1.1

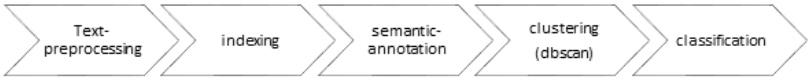


Figure 1.1: workflow

1.1 Preprocessing

Il preprocessing è uno step essenziale per qualunque task di text-mining, quando si ha a che fare con testo derivante dai social media come twitter, tale step diviene di vitale importanza a causa delle caratteristiche dei tweet. È dunque necessaria una fase accurata di preprocessing per il testo dei tweet prima di poter eseguire qualsiasi modellazione su di essi. Il primo passo in questa fase è la rimozione di quelle feature testuali legate ai tweet come:

- *url:* dal testo sono eliminati tutti i riferimenti a url o media
- *@-mentions:* vengono eliminate tutte le mentions ad altri utenti
- *#hashtag:* Per quanto riguarda gli hashtag viene solo eliminato il caratattere # poichè se fossero eliminati si potrebbe perdere della semantica dal testo. Inoltre spesso nei tweet gli hashtag sono composizioni di più parole dove ciascuna parola inizia con una lettera maiuscola (camel Case), come ad esempio #StopBombingGaza. Gli hashtag che si presentano nella forma su descritta verranno scomposti in nelle parole che di cui sono composti (#StopBombingGaza → Stop Bombing Gaza).
- *RT:* per i retweet viene considerato il testo del tweet originale.

Per il primo passo non è necessaria alcuna fase di parsing del testo poichè tutte queste informazioni sono fornite dalle api di twitter sotto forma di dati strutturati[1]. Molto spesso i tweet sono composti da keywords appartenenti a lingue diverse o contengono caratteri speciali, per tale motivo saranno eliminati tutti i non latin characters. Tramite apposita espressione regolare vengono identificati ed eliminate le emoticons presenti nel testo. Una volta effettuate queste pulizie, si passa all'identificazione della lingua del tweet per mezzo di una libreria java[2], e saranno considerati solo i tweet in lingua inglese.

[1] nell attributo entities del tweet

[2] https://code.google.com/p/language-detection/

1.2 Indexing

Il testo del tweet verrà rappresentato con un boosted tf-idf vector utilizzando Apache-lucene[3]. A partire dal testo dei tweet ripulito, sono state effettuati ulteriori step di nlp come :

- stop word removal
- pos tagging (tramite lo stanford pos tagger addestrato su un modello creato a partire da tweet
- stemming (o lemmtization)

Poichè si analizza uno stream di tweet in maniera incrementale è necessario che anche lo schema di pesatura tf-idf sia incrementale ovvero le document-frequencies per una word w variano nel tempo.

Il peso di una word w di un tweet avente tempo t è dato da:

$$tf - idf_w = tf(w)\ log\frac{N_t}{df_t(w)}boost(w) \tag{1.1}$$

Dove N_t è il numero di tweet al tempo t. Risulta fondamentale assegnare un boost a determinate word poiché, a causa della natura dei tweet (max 140 caratteri), il tf di ciascuna keyword è solitamente pari a 1.

$$boost(w) := \begin{cases} 2.0 & \text{se } w \text{ è un hashtag,} \\ 1.5 & \text{se } w \text{ è un Proper-Noun,} \\ 1 & \text{altrimenti.} \end{cases}$$

Solitamente uh hashtag ha un alto potere informativo all'interno di un tweet, nel lavoro di Phuvipadawat [PM10] il boost pari a 1.5 per i poper-noun ha prodotto risultati migliori.

[3]https://lucene.apache.org/core/

1.3 Semantic Annotation

Tramite Dbpedia-Spootlight [Dai+13] è possibile annotare automaticamente il testo dei tweet con DBpedia URIs. Ciò permette di arricchire la rappresentazione testuale andando a lenire i classici problemi di ambiguità del linguaggio naturale come polisemia e sinonimia. Annotare il testo con DBpedia URis permette di sfruttare la base di conoscenza di DBpedia per poter determinare la correlazione semantica fra due termini. Il progetto DBpedia [Aue+07] oltre a organizzare in maniera strutturata le informazioni di Wikipedia, le collega ad ulteriori open-datasets come: US Census, Geonames, MusicBrainz, the DBLP bibliography, WordNet, Cyc, tramite RDF links come mostrato in figura 1.2

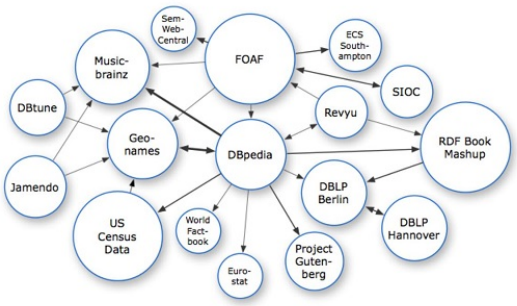


Figure 1.2: Datsets interconnessi a DBpedia.

Avendo due risorse DBpedia è possibile definire una funzione di distanza, che sfrutti la base di conoscenza di DBpedia. È stata definita una funzione di distanza *Dbpedia Semantic Distance DSD*, che impiega sia DBpedia che il dataset geografico Geonames per risorse di tipo geografico, come segue:

$$DSD(a,b) := \begin{cases} GeoDist(a,b) & \text{se } type(a) = location \wedge type(b) = location \ , \\ NDD(a,b) & \text{altrimenti} \end{cases} \tag{1.2}$$

Dove $DSD(a,b) \in [0,1]$, Milne e Witten [MW08] hanno utilizzato gli hyperlink delle pagine Wikipedia per poter definire la correlazione fra due articoli wikipedia (e quindi due risorse dbpedia): date due risorse a, b, possiamo definire una *Normalized DBpedia Distance (NDD)* come segue:

$$NDD(a,b) := \begin{cases} \frac{\log(max\{|A|,|B|\}) - log(|A \cap B|\})}{N - \log(min\{|A|,|B|\})} & \text{se } A \cap B \neq \emptyset \ , \\ 1 & \text{altrimenti} \end{cases} \tag{1.3}$$

A e B sono gli insiemi delle risorse DBpedia che hanno un link rispettivamente verso a e b, mentre N è il numero totale di risorse in DBpedia. Questa distanza varia nell'intervallo [0,1] dove 1 sta ad indicare che non vi è nessuna correlazione fra i due concetti, mentre 0 indica che i due concetti hanno lo stesso significato. L'idea alla base, è che due risorse saranno simili se esiste una terza che ha un link verso entrambe.

Per le risorse DBpedia di tipo "location"[4] è stata definita la funzione di distanza 1.6 che impiega il dataset "GeoNames". Poiché il task, è quello di identificare eventi1 che accadono in un luogo specifico, è utile valutare la correlazione sulla base di informazioni geografiche. Geonames [5] è un database geografico contenente più di 10 milioni di nomi geografici, ogni risorsa è classificata da una "feature class" (administrative divisions, populated places, structures, mountains, water bodies, etc) e ulteriormente sotto classificata da 645 "feature codes". Tra le varie informazione e relazioni che interconnettono le risorse nell'ontologia sono state utilizzate le seguenti per poter valutare la similarità fra le risorse:

- *Country-code:* codice identificativo della nazione cui appartiene la risorsa
- *FeatureClass*
- *FeatureCode*
- *parentADM1:* risorsa di tipo ADM1(regione) che contiene la risorsa
- *parentADM2:* risorsa di tipo ADM2(provincia) che contiene la risorsa
- *Coordinates:* (latitude,longitude)

La funzione di similarità deve quindi tener conto dei diversi livelli di granularità delle risorse Geonames e sfruttare le relazioni fra di esse. Verrà definita una distanza *Geonames Distance (GD)* compresa fra [0,1] cosi definita:

$$GD(a,b) := \begin{cases} 0.8 & \text{se } \exists c \mid (c\ parentADM1\ a) \ \wedge \ (c\ parentADM1\ b) \\ 0.8 & \text{se } \ (a\ parentADM2\ b) \ \vee \ (b\ parentADM2\ a) \\ 0.3 & \text{se } \exists c \mid (c\ parentADM2\ a) \ \wedge \ (c\ parentADM1\ b) \\ 0.3 & \text{se } \ (a\ parentADM2\ b) \ \vee \ (b\ parentADM1\ a) \\ 1 & \text{altrimenti.} \end{cases} \tag{1.4}$$

[4]per risorse di tipo location si intendono quelle risorse classficate come http://dbpedia.org/ontology/Location

[5]http://www.geonames.org/

Per alcune risorse di granularità più fine come città, parchi, edifici ha senso invece, considerare le coordinate. Avendo le coordinate è possibile calcolare la distanza espressa in km fra di esse usando l'Harvesine Forula[6].Sia d la distanza espressa in km tra a e b, verrà definita una *Coordinate Distance (CoordDist)*

$$coordDist(a,b) := \begin{cases} 0.0 & \text{se } 0 < d \leq 5, \\ 0.2 & \text{se } 5 < d \leq 10, \\ 0.25 & \text{se } 10 < d \leq 15, \\ 0.3 & \text{se } 15 < d \leq 25, \\ 0.4 & \text{se } 25 < d \leq 40, \\ 0.7 & \text{se } 40 < d \leq 50, \\ 0.8 & \text{se } 70 < d \leq 80, \\ 1 & \text{altrimenti.} \end{cases} \tag{1.5}$$

Si può cosi definire la distanza geografica *GeoDist* come segue:

$$GeoDist(a,b) := \begin{cases} coordDist(a,b) & \text{se sia a e b hanno coordinate} \\ GD(a,b) & \text{altrimenti.} \end{cases} \tag{1.6}$$

[6]Haversine formula

1.4 Clustering

L'algoritmo di clustering utilizzato è Incremental Dbscan poiché può gestire bene il rumore e non necessita di parametri come il numero di cluster a priori. La distanza utilizzata si basa sia sulla rappresentazione tf-idf del tweet sia sui DBpedia URIs estratti :

$$dist(a,b) = 1 - timeSim(a,b)\,\frac{textSim(a,b) + semanticSim(a,b)}{2} \tag{1.7}$$

La similarità testuale è data dalla similarità del coseno fra i vettori tf-idf dei due tweet:

$$textSim(a,b) = \frac{v_a \cdot v_b}{||v_a||\,||v_b||}$$

Anche il tempo di creazione dei tweet verrà preso in considerazione per valutarne la distanza, poiché anche se due tweet avessero un testo molto simile es *"tonight flashmob in central park"*, ma fossero pubblicati ad un mese di distanza, è molto inverosimile che si riferiscano al medesimo evento. Per tale ragione, è stata definita una similarità temporale [7]

$$timeSim(a,b) := \begin{cases} 1 - \frac{|d_a - d_b|}{31} & \text{se } |d_a - d_b| < 31\ , \\ 0 & \text{altrimenti} \end{cases}$$

Ad un tweet, tramite il processo di annotazione semantica, possono essere associate una nessuna o più risorse DBpedia,la distanza semantica sarà data dalla distanza degli insiemi di risorse associati ai due tweet valutata secondo la distanza definita in precedenza per le risorse DBpedia1.2. La similarità fra un elemento x e un insieme Y è data da:

$$sim(x,Y) = sup\{1 - DSD(x,y)\ |y \in Y\}$$

Dati due insiemi di risorse DBpedia D_a, D_b la similarità fra i due insiemi sarà definita come:

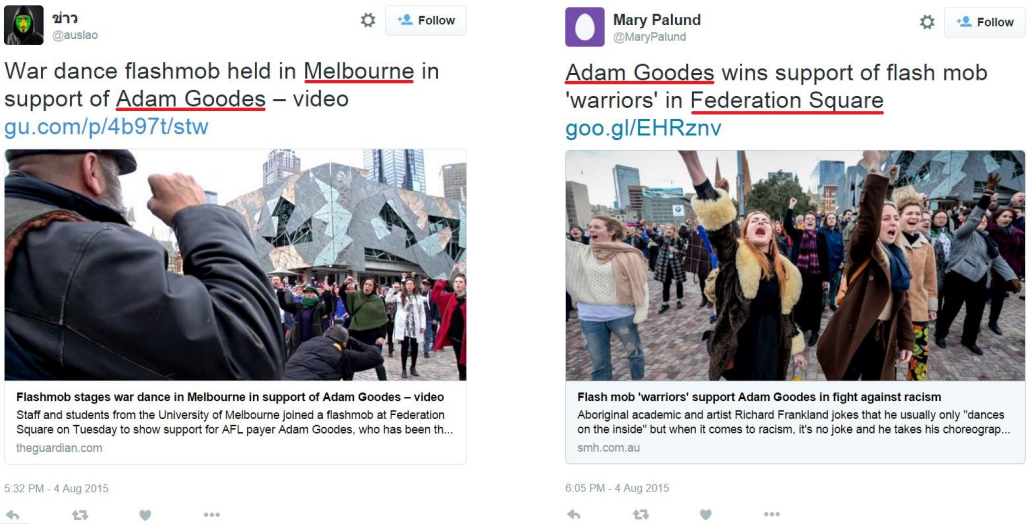
$$sim(D_a,D_b) = \frac{\sum\limits_{x \in D_a} sim(x,D_b)}{|D_a|} \tag{1.8}$$

Questa funzione di similarità non è simmetrica,[8] Per ottenere una funzione simmetrica è sufficiente definirla come:

$$semanticSim(a,b) := \begin{cases} \frac{sim(D_a,D_b)+sim(D_b,D_a)}{2} & \text{se } D_a, D_b \neq \emptyset \\ 1 & \text{altrimenti.} \end{cases}$$

[7] d_a=#days from the epoch of tweet a

[8] se $D_a \subseteq D_b \Rightarrow sim(D_a,D_b) \neq sim(D_b,D_a)$



(a) (b)

Figure 1.3: Due tweet appartenti allo stesso cluster

Se ad uno dei due tweet non è associata nessuna risorsa, la similarità semantica sarà pari ad uno, quindi la loro similarità sarà valutata solo in base alla loro rappresentazione testuale. Utilizzare una similarità semantica serve ad attenuare il problema della "fragmentation" di cui sono affetti i metodi document-pivot, ovvero utilizzando solo la similarità testuale molti eventi possono essere erroneamente suddivisi in più cluster. Inoltre Petkos e Papadopoulus [PPK14] hanno constato che se due tweet condividono uno stesso URL, o un tweet e in reply all'altro, allora si riferiscono allo stesso topic/evento.

Si considerino i tweet a, b in figura 1.3:

- $textSim(a, b) = 0.46$
- $timeSim(a, b) = 1.0$ i tweet sono stati pubblicati entrambi il 4/8/2015
- $semanticSim(a, b) := \frac{sim(D_a, D_b) + sim(D_b, D_a)}{2}$
  $D_a =$ { <Melbourne>, <Adam Goodes>}
  $D_b =$ { <Adam Goodes>, <Federation Square>}

$$
\begin{aligned}
NSD(Melbourne, AdamGoodes) &= NDD(Melbourne, AdamGoodes) \\
&= \frac{\log(max\{|A|, |B|\}) - log(|A \cap B|)}{N - \log(min\{|A|, |B|\})} \\
&= \frac{\log(643) - log(10)}{N - \log(138)} = 0.367
\end{aligned}
$$

$$
\begin{aligned}
&NSD(Melbourne, FederationSquare) = \\
&= GeoDist(Melbourne, FederationSquare) = \\
&= CoordDist(Melbourne, FederationSquare) = 0 \text{ (poichè la distanza in km è 0.8)}
\end{aligned}
$$

$$
\begin{aligned}
&NSD(AdamGoodes, FederationSquare) = \\
&\quad NDD(AdamGoodes, FederationSquare) = 1 \text{ poichè } A \cap B = \emptyset
\end{aligned}
$$

$$
\begin{aligned}
sim(Melbourne, D_b) &= sup\{1 - DSD(Melbourne, y) \,|y \in D_b\} \\
&= sup\{(1 - DSD(Melbourne, AdamGoodes)), \\
&\quad (1 - DSD(Melbourne, FederationSquare))\} \\
&= sup\{(1 - 0.367), (1 - 0)\} = 1 \\
sim(AdamGoodes, D_b) &= sup\{1 - DSD(AdamGoodes, y) \,|y \in D_b\} \\
&= sup\{(1 - DSD(AdamGoodes, AdamGoodes)), \\
&\quad (1 - DSD(AdamGoodes, FederationSquare))\} \\
&= sup\{(1 - 0), (1 - 1)\} = 1 \\
sim(D_a, D_b) &= \frac{1+1}{2} = 1,\ sim(D_b, D_a) := \frac{1+1}{2} = 1 \\
&\implies semanticSim(a, b) = 1
\end{aligned}
$$

$$
\begin{aligned}
dist(a, b) &= 1 - timeSim(a, b)\,\frac{textSim(a, b) + semanticSim(a, b)}{2} \\
&= 1 - \frac{0.46 + 1}{2} = 0.269
\end{aligned}
$$

Se invece, *SemanticSim* fosse definita come la media delle distanze fra tutte le possibili coppie fra i due insiemi avremmo:

$$semanticSim(a,b) = \frac{\sum_{x \in D_a} \sum_{x \in D_b} (1 - NSD(x,y))}{|D_a||D_b|} =$$
$$= \frac{(1-0.36) + (1-0) + (1-0) + (1-1)}{4} = 0.73$$

1.5 Classification

Per poter filtrare gli eventi reali da quelli non reali è stato addestrato un classificatore SVM, sulla base di statistiche derivanti da cluster individuati. Tali statistiche derivano da feature dei tweet che compongono i cluster e dalle annotazioni estratte da tali tweets.

- TWEET FEATURES
  - %retweets: percentuale di retweets presenti nel cluster.
  - %replies: percentuale di tweet nel cluster che sono replies
  - %mentions: percentuale di tweet nel cluster che contengono mentions
  - %hashtags: percentuale di tweet nel cluster che contengono hashtag
  - %urls: percentuale di tweet nel cluster che contengono urls
  - %media: percentuale di tweet nel cluster che contengono media
  - %authors: percentuale di autori distinti dei tweet nel cluster (se i tweet sono stati creati
- TOPIC FEATURES
  - avg token number: numero medio di token dei tweet che compongono il cluster
  - %WhereAnnotations: percentuale di tweet nel cluster che contengono annotazioni semantiche di tipo Location
  - HasSpecificLocation: booleano che indica se fra le annotazioni di tipo location vi sia almeno una con delle coordinate specifiche.

Bibliography

[All02] James Allan, ed. *Topic Detection and Tracking: Event-based Information Organization*. Norwell, MA, USA: Kluwer Academic Publishers, 2002. ISBN: 0-7923-7664-1.

[Aue+07] Sören Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. 2007, pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52. URL: http://dx.doi.org/10.1007/978-3-540-76298-0_52.

[MW08] David Milne and Ian H. Witten. "An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links". In: *In Proceedings of AAAI 2008*. 2008.

[PM10] Swit Phuvipadawat and Tsuyoshi Murata. "Breaking News Detection and Tracking in Twitter". In: *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*. WI-IAT '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 120–123. ISBN: 978-0-7695-4191-4. DOI: 10.1109/WI-IAT.2010.205. URL: http://dx.doi.org/10.1109/WI-IAT.2010.205.

[Dai+13] Joachim Daiber et al. "Improving Efficiency and Accuracy in Multilingual Entity Extraction". In: *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. 2013.

[PPK14] Georgios Petkos, Symeon Papadopoulos, and Yiannis Kompatsiaris. "Two-level Message Clustering for Topic Detection in Twitter". In: *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*. 2014, pp. 49–56. URL: http://ceur-ws.org/Vol-1150/petkos.pdf.