Grafi di conoscenza e testi storici.

Costruzione di un grafo a partire dai bollettini emessi dal Comando Supremo italiano con il resoconto quotidiano delle operazioni militari durante la Grande Guerra.

Seminario di Cultura Digitale.
Lorenzo De Mattei (mat. 469944)

1 Introduzione

Questo lavoro è stato realizzato nell'ambito del Seminario di Cultura Digitale, del corso di laurea in Informatica Umanistica. L'obbiettivo è quello di costruire un grafo di conoscenza[1] i cui nodi sono costituiti dalle entità presenti in una raccolta testuale di bollettini di guerra risalenti alla Prima Guerra Mondiale. Il lavoro si ispira al progetto Memorie di Guerra (http://www.memoriediguerra.it/site) nell'ambito del quale sono stati analizzati linguisticamente e semanticamente i bollettini con l'obbiettivo di creare un indice semantico.

L'obbiettivo del progetto tuttavia non è solo quello di realizzare un grafo relativo ai bollettini, ma anche quello di proporre una metodologia che consenta di costruire grafi di conoscenza basati su testi storici. Tale metodologia dovrà essere scalabile su collezioni di ampie dimensioni e caratterizzate da domini linguistici differenti.

Il progetto si ispira ai principi del web semantico[2], in quanto l'obbiettivo è quello di arricchire collezioni di testi storici con conoscenza di tipo strutturato, proveniente da ontologie semantiche che sia usufruibile da strumenti applicativi come motori di ricerca semantici.

1.1 Il progetto Memorie di Guerra

Nell'ambito del progetto Memorie di Guerra sono stati analizzati linguisticamente e semanticamente i bollettini con l'obbiettivo di creare un indice semantico.

Il risultato applicativo di questo progetto consiste in un sito web che consente di effettuare ricerche nei testi utilizzando informazioni semantiche. Si possono effettuare diversi tipi di ricerche ed è possibile esplorare in vari modi informazioni aggregate di natura semantica estratte dal testo.

Questi risultati sono stati ottenuti grazie a strumenti di analisi linguistica automatica. Il testo è stato infatti tokenizzato, lemmatizzato e sono stati effettuati il POS tagging e il dependency parsing. Inoltre è stato utilizzato uno strumento di Named Entity Recognition al fine di estrarre le entità nominate nel testo (luoghi, persone, unità militari, aerei e navi). Infine i luoghi sono stati geo-localizzati e le altre entità sono state linkate a risorse esterne come Wikipedia. Per una descrizione più completa del progetto rimando a Boschetti e altri (2014).

---

[1] Una rete semantica (o grafo di conoscenza) è una forma di rappresentazione della conoscenza. Essa è un grafo orientato o non orientato formato da vertici, che rappresentano concetti, e archi, che rappresentano relazioni semantiche tra i concetti. (Wikipedia, pagina Rete semantica).

[2] Con il termine web semantico, termine coniato dal suo ideatore, Tim Berners-Lee, si intende la trasformazione del World Wide Web in un ambiente dove i documenti pubblicati (pagine HTML, file, immagini, e così via) sono associati ad informazioni e dati (metadati) che ne specificano il contesto semantico in un formato adatto all'interrogazione e l'interpretazione (es. tramite motori di ricerca) e, più in generale, all'elaborazione automatica. (Wikipedia, pagina Web semantico)

Il progetto ha obbiettivi molto ambiziosi, ma risultano evidenti alcuni limiti in termini di scalabilità e robustezza degli strumenti di analisi linguistica per questa tipologia di testi. I bollettini sulla prima guerra mondiale infatti sono caratterizzati da uno stile linguistico molto particolare. Gli strumenti di analisi linguistica utilizzati, di contro, sono basati su algoritmi di apprendimento automatico, addestrati su grandi corpora per-annotati. Questi corpora sono composti da testi di diversa tipologia e sono molto più recenti; pertanto presentano strutture linguistiche talvolta radicalmente diverse rispetto ai bollettini della Prima Guerra Mondiale.
Ciò ha reso necessarie importanti revisioni manuali degli output degli strumenti linguistici ed importanti sforzi di adattamento degli strumenti al dominio linguistico dei testi in analisi. Questi sforzi sono stati importanti ma la quantità di lavoro è stata sostenibile grazie al fatto che la collezione di testi non fosse eccessivamente ampia (1360 bollettini). Tra gli sviluppi futuri è stato proposto di ampliare il numero di risorse testuali, attingendo a nuove fonti di testi relativi alla Prima e alla Seconda Guerra Mondiale. Tuttavia questi nuovi testi potrebbero avere caratteristiche linguistiche molto eterogenee e forse questa tipologia di approccio potrebbe non essere più sostenibile.

1.2 Dal testo al grafo

Uno degli obbiettivo della presente dissertazione come detto è appunto quello di trovare metodologie alternative che consentano di ottenere risultati applicativi simili che siano tuttavia più scalabili su collezioni di testi più grandi ed eterogenee.

Proprio per raggiunger tali obbiettivi è stato deciso di tentare di costruire un grafo di conoscenza relativo ai testi in analisi. Per farlo ho sviluppato un tool liberamente scaricabile all'indirizzo https://github.com/LoreDema/Text_to_graph. Questo tool sfrutta:

1. il servizio API di TAGME (http://tagme.di.unipi.it/), uno strumento di annotazione semantica automatica,
2. la libreria per Python sparql-client (https://pypi.python.org/pypi/sparql-client/) che consente di connettersi ad un endpoint SPARQL[3] e di richiedere le relazioni tra le entità in analisi,
3. la libreria per Python NetworkX (http://networkx.github.io/) che ci fornisce la struttura dati su cui costruire il grafo e una serie di metodi per la gestione e l'esplorazione del grafo stesso.

Quindi ho utilizzato questo tool per estrarre l'entità dal testo e costruire un grafo di conoscenza. Seguono una breve introduzione al web semantico, una descrizione degli strumenti di annotazione semantica ed in particolare di TAGME, quindi una descrizione completa del tool che ho sviluppato, una descrizione di come questo

[3] Un endpoint SPARQL è un servizio conforme al protocollo SPARQL come definitio nella specifica SPROT. Un endpoint SPARQL consente agli utenti (umani e non) di interrogare una knowledge base attraverso il linguaggio SPARQL. I risultati sono tipicamente restituiti in uno o più formati processabili dalle macchine. (Semanticweb.org, pagina SPARQL endpoint)

strumento sia stato utilizzato per la costruzione di un grafo di conoscenza a partire dai bollettini sulla Prima Guerra Mondiale ed infine saranno esposti i risultati ottenuti e le possibili applicazioni.

2 Web Semantico

Il termine Web Semantico fu coniato da Time Berners Lee (Berners Lee et al 2001) e con esso si identifica un'estensione del Web stesso realizzata attraverso standard definiti dal World Wide Web Consortium (W3C). La promessa del web semantico è quella di passare da un Web dei dati ad un Web dei significati e dei servizi distribuiti. Per fare ciò è necessario che i documenti pubblicati sul web siano associati ad informazioni specificate attraverso l'utilizzo di metadati codificati tramite determinati standard. Queste informazioni devono specificare il contesto semantico di ogni risorsa e i formati devono essere adatti all'interrogazione e all'elaborazione automatiche. Ciò consentirebbe lo sviluppo di applicazioni Web più complesse di quelle attuali che sfruttino appunto la conoscenza semantica codificata delle risorse disponibili sul Web.

Questa idea ha attratto ricercatori provenienti da diverse discipline che si sono posti l'obbiettivo di definire nuovi linguaggi e architetture necessari per rendere questa sfida realizzabile. Nel 2006 lo stesso Tim Berners Lee sosteneva che questa semplice idea fosse rimasta largamente non realizzata (Berners Lee et al 2006), ma uno studio del 2013 ha messo in luce che ben 4 milioni di domini Web contengono markup standardizzato secondo i principi del Web Semantico (Guha, 2013).

2.1 Grafi di conoscenza

Un grafo di conoscenza è un grafo che rappresenta relazioni semantiche tra concetti (o entità). Questi tipi di grafi possono essere direzionati o non direzionati, in ogni caso ogni nodo rappresenta un concetto e ogni arco una relazione semantica consistente tra i due nodi che l'arco stesso collega (Sowa, 1987).
Wikipedia ad esempio può essere considerata come un'ontologia. Essa infatti contiene moltissime informazioni in parte strutturate. Possiamo considerare ogni voce enciclopedica come un entità alla quale sono associati metadati, categorie organizzate secondo strutture gerarchiche e una serie di link verso altre entità.
Ad esempio se analizziamo la voce Luigi Cadorna (https://it.wikipedia.org/wiki/Luigi_Cadorna), osserviamo come a questa siano associati una serie di metadati come il luogo e la data di nascita. Inoltre osserviamo che la voce appartiene a diverse categorie tra le quali Generali Italiani, che a sua volta appartiene alla categoria Generali e via dicendo fino ad arrivare alla categoria Persone e successivamente alla categoria radice Enciclopedia. Inoltre in ogni voce sono presenti una serie di link verso altre voci, ad esempio nella voce di Luigi

Cadorna possiamo trovare un link alla voce Armando Diaz. Tutte queste informazioni ci consentono di utilizzare Wikipedia come un grafo di conoscenza.

Tra gli standard del Web Semantico citiamo in questa sede il Resource Definition Framework (RDF), un modello standard, basato su xml, che consente lo scambio di dati attraverso il Web. Tramite RDF è possibile definire grafi di conoscenza secondo uno schema a triple (entità, relazione, entità). Questi grafi possono essere interrogati tramite il linguaggio SPARQL, attraverso l'utilizzo di endpoint SPARQL.

In rete è possibile trovare molte ontologie strutturate secondo lo standard, una delle più famose tra quelle di tipo generico è DBpedia (http://wiki.dbpedia.org/). DBpedia è basata sul grafo di Wikipedia e l'obbiettivo della comunità che gestisce questa ontologia è quello di arricchire le informazioni semantiche già presenti sulla Wikipedia attraverso il modello del crowdsourcing[4]. Le risorse di DBpedia sono usufruibili attraverso endpoint SPARQL. Esiste un progetto specifico basato sulla Wikipedia Italiana, appunto la DBpedia italiana (http://it.dbpedia.org).

3 Strumenti di annotazione semantica

Uno strumento di annotazione semantica ha lo scopo di collegare le entità presenti in un testo con le entità presenti in un'ontologia. Per esempio nella frase seguente un buon strumento di annotazione dovrà rilevare la presenza dell'entità *"Isonzo"* e collegarla alla rispettiva entità nell'ontologia di riferimento utilizzata.

```
Sull' Isonzo la nostra offensiva procede metodica, ordinata,
sicura.
```

Uno dei principali problemi che uno strumento di annotazione semantica deve risolvere è la risoluzione delle polisemie. Ad esempio nella frase "Cadorna è stato un generale italiano.", lo strumento deve riconoscere l'entità Luigi Cadorna, evitando di confondersi con Carlo Cadorna, presidente della Camera dei deputati nel 1857.
Per risolvere le polisemie gli strumenti tipicamente utilizzano un approccio statistico, sfruttando diverse tipologie di features basate sulla struttura del grafo di un'ontologia e su ancore testuali[5] che rimandano alle entità presenti in testi annotati.
Generalmente questi software utilizzano come ontologia Wikipedia, in tal modo infatti è possibile utilizzare il testo delle voci enciclopediche come risorsa testuale preannotata. Tra le principali features utilizzate dai vari sistemi vi sono:

[4] Il crowdsourcing (da *crowd*, "folla", e *outsourcing*, "esternalizzazione di una parte delle proprie attività") è un modello di business nel quale un'azienda o un'istituzione affida la progettazione, la realizzazione o lo sviluppo di un progetto, oggetto o idea ad un insieme indefinito di persone non organizzate precedentemente. Questo processo viene favorito dagli strumenti che mette a disposizione il web. Solitamente il meccanismo delle open call viene reso disponibile attraverso dei portali presenti sulla rete internet.
[5] Con ancora testuale si intende il testo visibile e cliccabile di un collegamento ipertestuale. (Wikipedia inglese, voce Anchor text). Per esempio nel testo delle voci di Wikipedia sono presenti molte ancore che rimandano ad altre voci/entità.

- La *commonness* di una pagina, ovvero il rapporto tra il numero di volte in cui una stringa *a* compare come ancora ad una entità *p* e il numero di volte in cui *a* compare nel testo:

$$Pr\,(p|a) = \frac{\#\,a\,linked\,to\,p}{\#\,a\,as\,anchor},$$

- Il contesto in cui compare un entità,
- La probabilità che una stringa *a* sia un link (*link probability*):

$$lp(a) = \frac{freq\,of\,a\,as\,anchor}{freq\,of\,a\,in\,the\,text},$$

- Features basate sulla struttura del grafo dell'ontologia.

3.1 TAGME

Tra i vari strumenti di annotazione semantica è stato scelto TAGME, sistema sviluppato da Paolo Ferragina e Ugo Scaiella presso l'università di Pisa basato su Wikipedia. La scelta è ricaduta su TAGME in quanto lo strumento ha accuratezza allo stato dell'arte, è molto prestante su testi brevi ed è molto efficiente in termini di velocità di calcolo. Inoltre è possibile utilizzare lo strumento tramite un comodo servizio API REST. Qui forniremo una sommaria descrizione dello strumento, per approfondire l'argomento è possibile riferirsi a Ferragina e Scaiella (2012).

Per analizzare un testo TAGME esegue innanzitutto la tokenizzazione e cerca ancore di lunghezza massima di 6 token effettuando un look up su un dizionario di coppie ancore testuali - entità. Le ancore estratte potrebbero sovrapporsi, quindi se un ancora a_1 è una sottostringa dell'ancora a_2 e la link probability di a_1 è minore della link probability di a_2 l'ancora a_1 viene eliminata in quanto la a_2 sarà sicuramente più specifica e il compito di disambiguazione sarà più semplice. Nel caso contrario potrebbe verificarsi che nella stringa a_2 sia presente uno o più token con scarso valore semantico, in tal caso quindi il sistema tiene traccia di entrambe le ancore e rimandia l'eliminazione a fasi sucessive.

Una volta estratte tutte le coppie ancora-entità inizia la fase di ***anchor disambiguation.*** In questa fase si cerca scegliere i link tra ancore ed entità in modo che statisticamente le entità scelte siano tra loro il più "vicine" possibile. La "vicinanza" tra due entità viene calcolata in modo statistico utilizzando particolari funzioni (*scoring function*) che attribuiscono un punteggio ad ogni coppia ancora-entità. Ad esempio nella frase "Cadorna fu sostituito da Diaz", l'ancora Cadorna sarà associata molto probabilmente all'entità Luigi Cadorna" e l'ancora Diaz sarà associata molto probabilmente all'entità "Armando Diaz", in quanto queste entità nel

grafo di Wikipedia saranno molto più correlate rispetto ad esempio alla coppia di entità “Carlo Cadorna” e “Cameron Diaz”.

La fase di anchor disambiguation produce un set di possibili annotazioni per un testo. La fase successiva detta ***anchor pruning*** elimina le annotazioni meno probabili e restituisce in output l’annotazione più probabile. Per fare questo si basa essenzialmente su due features: la link probability delle ancore e la coerenza tra le varie entità annotate.

L’approccio sopradescritto è tipico di diversi sitemi di annotazione semantica, la particolarità di TAGME risiede nelle scoring function, che risultano essere molto semplici ma efficaci, garantendo costi computazionali ridotti e precisione allo stato dell’arte. Inoltre queste scoring functions garantisco elevatissima precisione su testi brevi.

TAGME al momento è disponibile solamente per testi in lingua inglese e italiana.

4 Text_to_graph

Il tool Text_to_graph che ho realizzato consente di costruire un grafo di conoscenza a partire da testo semplice. Per fare questo innanzitutto sfrutta il servizio API di TAGME per estrarre le entità dal testo. Quindi utilizzando un endpoint SPARQL vengono estratte le relazioni tra le entità presenti nel testo, tutte le entità e le relazioni vengono quindi immagazzinate in una struttura dati apposita, fornita dalla libreria NetworkX.

Lo strumento è completamente sviluppato in Python, può essere utilizzato come libreria Python ed è molto semplice da utilizzare. Il software si divide in due moduli: il primo, `tagme`, consente di annotare tutti i testi presenti in una cartella, il secondo, `Graph`, consente di costruire il grafo e di effettuare alcune operazioni di base.

Il modulo `tagme` può essere eseguito come script Python oppure può essere richiamato all’interno di un altro script Python. Se vogliamo eseguirlo come script è necessario passare tre parametri tramite standard input: una API KEY da richiedere agli amministratori di TAGME (http://tagme.di.unipi.it/); la cartella dove si trovano i documenti da annotare, la cartella nella quale saranno salvate le annotazioni in formato JSON[6] (per una descrizione del formato rimando a http://tagme.di.unipi.it/tagme_help.html#tagging). Inoltre è possibile aggiungere un parametro che specifica la lingua dei testi da annotare, il valore di default è ‘it’ per l’italiano. Se vogliamo annotare testi in inglese dovremmo passare come parametro la stringa ‘en’. Esempio:

[6] JSON, acronimo di JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client-server. (Wikipedia, voce JSON)

```
~$ python tagme.py YOUR_API_KEY inptut_folder/ output_folder/
it
```

Per richiamare lo strumento di annotazione all'interno di uno script Python è invece necessario innanzitutto importare il modulo `tagme`, quindi sarà possibile richiamare la funzione, oltre ai parametri precedentemente descritti è possibile impostare altri due parametri: abstract, che determina la presenza o meno degli abstract descrittivi delle entità nell'output, e categories che determina la presenza o meno delle categorie descrittive delle entità nell'output. I valori di default sono True per entrambi i parametri. Esempio:

```
from src import tagme
tagme.tag('YOUR_API_KEY', 'input_folder/', 'output_folder/',
          lang='it', abstract=True, categories=True)
```

Una volta ottenute le annotazioni è possibile utilizzare il modulo `Graph` per costruire il grafo. Per farlo è necessario innanzitutto importare il modulo e quindi inizializzare un oggetto `Graph`, passandogli come parametro l'URI[7] generico delle entità della risorsa ontologica utilizzata. Esempio per inizializzare un grafo basato sulla DBpedia italiana:

```
from src import Graph
g = Graph.Graph('http://it.dbpedia.org/resource/')
```

Quindi è possibile costruire un grafo tramite il metodo `build_graph`. I parametri richiesti sono: il path della cartella in cui si trovano gli output annotati di TAGME e l'URL[8] dell'endpoint SPARQL che si vuole utilizzare per estrarre le relazioni tra le entità. Inoltre è possibile impostare altri due parametri: threshold stabilisce il limite minimo di perplessità p al di sotto del quale le entità annotate non vengono prese in considerazione, default $p = 0.1$, e rel_type che definisce quali tipologie di entità e relazioni estrarre, i valori accettabili di questo parametro sono 1 e 2. Se il rel_type è uguale ad per ogni entità vengono estratte tutte le relazioni che la riguardano, inserendo nel grafo anche entità non presenti nel testo. Altrimenti, se rel_type è viene impostato a 2, vengono estratte le relazioni tra ogni coppia di entità all'interno del grafo. Il valore di deafult è 1. Esempio:

---

[7] La locuzione Uniform Resource Identifier (in acronimo URI) in informatica, si riferisce a una stringa che identifica univocamente una risorsa generica che può essere un indirizzo Web, un documento, un'immagine, un file, un servizio, un indirizzo di posta elettronica, ecc. (Wikipedia, voce Uniform Resource Identifier)

[8] La locuzione Uniform Resource Locator (in acronimo URL), nella terminologia delle telecomunicazioni e dell'informatica è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, tipicamente presente su un host server, come ad esempio un documento, un'immagine, un video, rendendola accessibile ad un client che ne faccia richiesta attraverso l'utilizzo di un web browser. (Wikipedia, voce Uniform Resource Locator)

```
g.build_graph('output_folder/',
              'http://it.dbpedia.org/sparql',
              treshold=0.1, rel_type=2)
```

Prossimamente si prevede di aggiungere un terzo metodo che consenta di estrarre per ogni coppia di entità tutti i *path* di una lunghezza massima prefissata che le colleghino.

Una volta costruito il grafo avremmo un oggetto Graph che ha un attributo graph contenente un oggetto MultiDiGraph di NetworkX. Questo oggetto potrà essere gestito utilizzando tutti gli strumenti forniti da NetworkX, la documentazione completa è disponibile al link: http://networkx.github.io/documentation/networkx-1.9.1/.

Oltre a questo nel tool sono compresi una serie di metodi che consentono di svolgere alcune semplici operazioni e analisi sul grafo.

5 Dai bollettini al grafo

Passiamo ora ad analizzare come questo strumento sia stato utilizzato per creare il grafo di conoscenza sui bollettini della prima guerra mondiale.

I bollettini sono stati prima di tutto annotati utilizzando il modulo tagme. Quindi è stato costruito il grafo settando come treshold per ρ 0.1. La scelta per quel che riguarda la treshold è stata fatta analizzando manualmente l'output di tagme, osservando i risultati 0.1 è sembrato un valore sensato. Per quel che riguarda il parametro rel_type sono stati fatti esperimenti utilizzando entrambi i possibili valori.

È stato utilizzato l'endpoint SPARQL della DBpedia italiana (http://it.dbpedia.org/sparql), l'IRI generico delle entità per la DBpedia italiana è "http://it.dbpedia.org/resource/".

5.1 Analisi dei risultati ottenuti

Una delle sfide di questo progetto è quella di utilizzare sistemi di annotazione semantica per analizzare testi storici. In effetti non essendo TAGME basato su analisi linguistiche, lo strumento non dovrebbe essere particolarmente sensibile alle variazioni di dominio linguistico dei testi.

I testi annotati sono stati analizzati manualmente ed effettivamente, filtrando le annotazioni con indice di perplessità inferiore a 0.1, i risultati sembrano molto soddisfacenti. Tuttavia sarebbe necessario, per validare scientificamente l'approccio, fare una valutazione statistica sulla precisione del sistema su testi storici preannotati che costituiscano un gold standard. Purtroppo una risorsa del genere non è attualmente disponibile, non è stato pertanto possibile svolgere un'analisi di questo

tipo. Risorse di questo genere potrebbero essere create in maniera semi-automatica ricorrendo all'utilizzo di strumenti di crowdsourcing.

Inoltre la disponibilità di risorse gold darebbe l'opportunità di migliorare gli strumenti di annotazione semantica su testi di ambito storico (*domain adaptation*). Molto interessante sarebbe inoltre sfruttare algoritmi di *active learning*[9] insieme a strumenti di crowdsourcing per effettuare la domain adaptation in maniera più efficace.

6 Possibili applicazioni

Passiamo ora ad analizzare come questo grafo di conoscenza possa essere sfruttato a livello applicativo. Innanzitutto sfruttando uno strumento del genere è possibile costruire un indice semantico avanzato, ma non solo. Per esempio potremmo sfruttare il grafo per svolgere analisi statistiche di tipo quantitativo e qualitativo utili per fini di ricerca storica.

Analizziamo innanzitutto tutte le possibili funzioni che potrebbe fornire un indice semantico basato su un grafo di conoscenza costruito su testi storici datati come i bollettini della Prima Guerra Mondiale.

Oltre alla semplice ricerca testuale, si potrebbero cercare nei testi entità precise come persone e luoghi. L'applicazione messa a disposizione dal progetto Memorie di Guerra dà la possibilità di ricercare 5 tipologie di entità: persone, luoghi, navi aeroplani e unità militari. Queste sono le tipologie di entità che il Named Entity Recognizer utilizzato nell'ambito del progetto è in grado di estrarre. Un indice basato su un grafo di conoscenza potrebbe sfruttare molte più tipologie di entità, senza limitarsi ad un set predefinito.

Tramite l'applicazione di Memorie di Guerra è inoltre possibile effettuare ricerche di tipo spazio-temporale, in questo caso gioca un ruolo fondamentale la componente visuale che consente di visualizzare i luoghi della guerra su una mappa, dando per esempio la possibilità di visualizzare gli spostamenti del fronte durante le fasi della guerra. Utilizzando un grafo di conoscenza sarebbe possibile ottenere uno strumento equivalente in quanto le entità che fanno riferimento a luoghi sono georeferenziate.

Sfruttando questo tipo di tecnologie è inoltre possibile gestire interrogazioni più complesse. Ricordiamo che TAGME ha ottime prestazioni nell'annotazione di testi brevi. Un'interrogazione espressa in linguaggio naturale può essere considerata come un breve testo, quindi è possibile estrarre le entità presenti nell'interrogazione. Questo ci consente di effettuare ricerche di tipo semantico all'interno della repository documentale. Ad esempio se effettuamo una query come "A Cadorna successe

[9] L'active learning è un particolare caso di apprendimento automatico supervisionato nel quale un algoritmo di apprendimento automatico è in grado di interrogare interrativamente un oracolo, per ottenere l'output desiderato rispetto ad un particolare input. (Wikipedia inglese, voce Active Learning (machine learning))

Diaz", saranno ricercati i documenti in cui compaiono le entità "Luigi Cadorna" e "Armando Diaz", saranno esclusi per tanto documenti che parlano per esempio di "Cameron Diaz" o di "Carlo Cadorna".

Inoltre sfruttare questo tipo di tecnologie ci consente di arricchire i risultati delle ricerche presentando informazioni aggiuntive sulle entità coinvolte. Per esempio se viene effettuata una query in cui è presente l'entità Luigi Cadorna è possibile proporre informazioni aggiuntive, come l'abstract della voce di Wikipedia, data e luogo di nascita e di morte ecc. Nel motore di ricerca di Google è integrato un grafo di conoscenza utile per svolgere funzioni simili a quella appena descritta come si può vedere dallo screenshot estratto dai risultati della ricerca su Google.it della stringa "Cadorna" (img 1). Le informazioni aggiuntive che è possibile mostrare tuttavia non si limitano al fatto di mostrare una serie di metadati, ma è possibile definire delle tipologie di relazioni semantiche da mostrare a seconda della categoria ontologica a cui appartiene un entità. Ad esempio nel nostro grafo è codficato che Diaz è il sucessore di Cadorna. Questa è sicuramente un'informazione utile che potrebbe essere mostrata all'utente. Come esempio ulteriore se avessimo un entità che corrisponde alla categoria "pittori" potremmo mostrare una lista contenente tutte le opere dipinte da un autore. Sarebbe interessante confrontarsi con esperti nel settore storico per definire per le varie categorie di entità quali siano le informazioni più rilevanti da mostrare all'interno di un motore di ricerca per testi storici.

Inoltre, come si può vedere nello screenshot estratto da Google (img 1), è possibile proporre entità correlate a quelle ricercate per suggerire nuove ricerche all'utente. Ad esempio un utente interessato a Cadorna, probabilmente potrebbe essere interessato anche a Diaz, l'entità che nel grafo di conoscenza di Google risulta essere maggiormente correlata a Cadorna.

Oltre a queste ed altre possibili applicazioni il grafo può essere sfruttato per effettuare analisi statistiche sia di tipo quantitativo che di tipo qualitativo utili per fini di ricerca storica.

Per esempio analizzando il grafo estratto a partire dai bollettini della Prima Guerra Mondiale è possibile evidenziare come questo sia particolarmente coeso tramite analisi statistiche. Inanzitutto osserviamo che il grafo è connesso, ovvero non esistono sottografi non collegati tra loro. Inoltre osserviamo che la lunghezza media dei cammini minimi tra ogni coppia di entità è 2.74, un valore molto basso che certifica la coesione del grafo. Potrebbe essere interessante ad esempio provare ad aggiungere bollettini relativi alla Seconda Guerra Mondiale e osservare quanto la coesione del grafo verrebbe ridotta con l'inserimento di entità estranee a quel determinato periodo storico. Inoltre potrebbe essere analizzata la variazione del clustering coefficent, un'indice che esprime quanto un grafo sia organizzato in sottografi strettamente collegati. Potrebbero poi essere ricercate componenti del grafo strettamente correlate cercando di andare ad individuare quali entità sono

centrali per i bollettini relativi alla Prima Guerra Mondiale, quali per i bollettini relativi alla Seconda Guerra Mondiale e quali per entrambi.

Un'altra analisi statistica che è possibile effettuare sul grafo è quella di osservare la distruzione del grado di centralità delle entità presenti nel grafo, magari suddividendole per categoria. Il grado di centralità delle varie entità potrebbe essere calcolato sfruttando diversi indici, quali la betweenness[10], la closeness[11] o il PageRank[12].
Una volta ottenute il grado di centralità per ogni entità potremmo fare delle classifiche per visualizzare quali siano le più importanti. Queste classifiche potrebbero poi essere fatte su specifiche categorie come persone, o più specificatamente Generali, oppure luoghi e comuni ecc.

A titolo esemplificativo ho svolto qualche piccolo esperimento calcolando il PageRank delle entità presenti nel testo. Osserviamo da questi esperimenti che le entità più centrali in assoluto sono “Italia” e “Prima Guerra Mondiale”. I luoghi più centrali sono invece in ordine: “Italia”, “Francia”, “Germania”, “Stati Uniti d’America”.

[10] La btweenness a un indicatore di centralità di un nodo all’interno di un grafo. Il valore è dato dal numero di cammini minimi tra tutte le coppie di nodi che attraverso il nodo stesso (Wikipedia inglese, voce Betweenness centrality)
[11] La closeness è un indicatore di centralità di un nodo all’interno di un grafo. Il valore è dato dalla media della lunghezza dei cammini minimi che connettono tale nodo a tutti gli altri nodi del grafo (Wikipedia inglese, voce Centrality).
[12] Il PageRank è un algoritmo di analisi che assegna un peso numerico ad ogni nodo di un grafo con lo scopo di quantificare la sua importanza relativa. (Wikipedia, voce PageRank)

Img 1: Screenshot da Google. Query: Cadorna (https://www.google.it/search?client=ubuntu&channel=fs&q=cadorna&ie=utf-8&oe=utf-8&gfe_rd=cr&ei=XvRlVaqgD8iO8Qe7v4DACg)

7 Conclusioni

I grafi di conoscenza sono sicuramente strumenti potenti che garantisco la possibilità di sviluppare applicazioni complesse molto utili. Non è un caso che grandi compagnie, leader mondiali nell'ambito del Web e dei motori di ricerca, stiano investendo ingenti risorse nello sviluppo di questo tipo di tecnologie.
Con questo lavoro si è arrivati a mettere in luce come tali tecnologie possano essere utilizzate nell'ambito di studi umanistici, in particolare quello storico, per costruire indici semantici utili ai ricercatori per garantire un accesso rapido alle fonti. Ma le potenzialità di tali tecnologie non si fermano qui, infatti come abbiamo visto questi strumenti consentono di sviluppare applicazioni complesse di vario tipo che possono essere utilizzate per scopi di ricerca e per scopi didattici. Nell'ultima sezione, quella dedicata alle possibili applicazioni, sono stati dati alcuni spunti, ma il campo è aperto e le potenzialità sono molte.

8 Bibliografia

Berners-Lee, Tim, James Hendler, Ora Lassila (May 17, 2001). "The Semantic Web". *Scientific American Magazine*.

Federico Boschetti, Andrea Cimino, Felice Dell'Orletta, Gianluca E. Lebani, Lucia Passaro, Paolo Picchi, Giulia Venturi, Simonetta Montemagni, Alessandro Lenci (2014), *"Computational Analysis of Historical Documents: An Application to Italian War Bulletins in World War I and II"*, in Proceedings of the LREC 2014 Workshop on "Language resources and technologies for processing and linking historical documents and archives – Deploying Linked Open Data in Cultural Heritage" (LRT4HDA 2014), Reykjavik, 26 May 2014.

François Fages, Sylvain Soliman (2005). *"Principles and Practice of Semantic Web Reasoning".* Springer, Dagstuhò Castle, Germany.

John F. Sowa (1987). "Semantic Networks". In Stuart C Shapiro. *Encyclopedia of Artificial Intelligence*.

Ramanathan V. Guha (2013). "Light at the End of the Tunnel". International Semantic Web Conference *2013 Keynote*.

Nigel Shadbolt, Wendy Hall, Tim Berners-Lee (2006). "The Semantic Web Revisited" . *IEEE Intelligent Systems*.

Paolo Ferragina, Ugo Scaiella: *Fast and Accurate Annotation of Short Texts with Wikipedia Pages*. IEEE Software 29(1): 70-75 (2012)

World Wide Web Consortium (W3C) (November 7, 2011). "W3C Semantic Web Activity".

Wikipedia, voce Rete semantica, <http://it.wikipedia.org/wiki/Rete_semantica>, maggio 2015

Wikipedia, voce JavaScript Object Notation, <http://it.wikipedia.org/wiki/JavaScript_Object_Notation>, maggio 2015

Wikipedia, voce URI, <https://it.wikipedia.org/wiki/URI>, maggio 2015

Wikipedia, voce Uniform Resource Locator, <https://it.wikipedia.org/wiki/Uniform_Resource_Locator>, maggio 2015

Wikipedia, voce PageRank, <http://it.wikipedia.org/wiki/PageRank>, maggio 2015

Wikipedia inglese, voce Anchor text, <http://en.wikipedia.org/wiki/Anchor_text>, maggio 2015

Wikipedia inglese, voce Active learning, <http://en.wikipedia.org/wiki/Active_learning_%28machine_learning%29>, maggio 2015

Wikipedia inglese, voce Betwenness centrality, <http://en.wikipedia.org/wiki/Betweenness_centrality>, maggio 2015

Wikipedia inglese, voce Centrality, <http://en.wikipedia.org/wiki/Centrality>, maggio 2015

Semantic Web Wiki, voce Ontology, <http://semanticweb.org/wiki/Ontology>, maggio 2015

Semantic Web Wiki, voce SPARQL, <http://semanticweb.org/wiki/SPARQL_endpoint>, maggio 2015

Sito del W3C, pagina RDF, <http://www.w3.org/RDF/>, maggio 2015

Wiki Dbpedia, <http://wiki.dbpedia.org/>, maggio 2015

Sito della Dbpedia italiana, <http://it.dbpedia.org>, maggio 2015