SDDSqa: un sistema di Question Answering per Dati Strutturati

Lucia Siciliani, Pierpaolo Basile, Pasquale Lops
Dipartimento di Informatica - Università degli Studi di Bari Aldo Moro
{nome.cognome}@uniba.it

Abstract

In questo articolo presentiamo SDDSqa, un sistema di Question Answering per dati strutturati in grado di trasformare domande in linguaggio naturale nel loro equivalente SPARQL. Tra i vari approcci presenti in letteratura, quelli basati su linguaggi controllati hanno ottenuto buoni risultati e, opportunamente combinati con meccanismi di auto-completamento, si sono dimostrati validi anche in contesti reali. Essi tuttavia non sono esenti da limiti: in particolare la restrizione sul vocabolario utilizzato non esime l'utente dalla conoscenza della terminologia utilizzata all'interno della sorgente dati. Con il nostro lavoro abbiamo cercato di superare questo problema utilizzando tecniche basate sulla semantica distribuzionale e introducendo un meccanismo di backtracking. Mostriamo inoltre un caso di studio in cui il nostro sistema è utilizzato come interfaccia in linguaggio naturale per dati resi disponibili dalle Pubbliche Amministrazioni della Regione Puglia inerentemente al settore del turismo.

1 Introduzione

Nell'ultimo ventennio si è assistito ad una notevole crescita del numero e della corposità di Basi di Conoscenza (Knowledge Bases, KBs) come DBpedia [Auer *et al.*, 2007] e Wikidata [Vrandečić e Krötzsch, 2014]. Parte del loro successo è dovuta alla pubblicazione da parte del W3C dei due linguaggi standard per la rappresentazione e l'interrogazione dei dati all'interno delle KBs che sono, rispettivamente, RDF e SPARQL. Nonostante si tratti di linguaggi espressivi e di alto livello, essi sono caratterizzati da un formalismo che li rende generalmente difficili da utilizzare da parte di utenti non esperti.

Lo scopo del Question Answering su KBs è dunque quello di studiare i metodi e le tecniche in grado di trasformare una domanda scritta dall'utente usando il linguaggio naturale, in una query scritta in un particolare linguaggio di interrogazione come ad esempio SPARQL. In letteratura, l'interesse nei confronti di questo tema di ricerca è nato negli anni 60 con la creazione delle prime interfacce in linguaggio naturale per database e con il tempo si è evoluto, portando all'esplorazione di nuovi approcci. Tra questi, gli approcci basati su linguaggi controllati hanno dimostrato di poter ottenere ottimi risultati, anche se essi non sono esenti da limiti, che abbiamo cercato di superare con SDDSqa.

2 Metodologia

L'idea alla base del metodo utilizzato in CANaLI [Mazzeo e Zaniolo, 2016] è che il processo di ricerca della risposta all'interno della KB, attraverso entità, classi e proprietà, possa essere visto come deterministico.

Se vogliamo ad esempio conoscere qual è la capitale d'Italia, dobbiamo necessariamente prendere in considerazione dapprima l'entità `dbo:Italy`, in seguito individuare tra tutte le proprietà quella corretta, cioè `dbo:capital` ed infine considerare l'entità `dbr:Rome`, che rappresenta la risposta alla domanda. Bisogna inoltre notare che l'entità `dbo:Italy` gode della proprietà `dbo:capital` in quanto appartenente alla classe `dbo:Country`: un'istanza della classe `dbo:Person` ad esempio non potrebbe mai avere tale proprietà tra quelle ammissibili. Utilizzando un linguaggio controllato, ottenuto considerando un sottoinsieme del vocabolario e delle regole grammaticali di una lingua, l'analisi della domanda e la navigazione tra le risorse della KB possono essere modellate utilizzando un automa a stati finiti. Accompagnato da un meccanismo di auto-completamento, che permette di guidare in tempo reale l'utente nella formulazione di domande corrette e non ambigue, questo tipo di approccio consente di analizzare la domanda come una sequenza di token disgiunti e di trasformarla in una query SPARQL ben formata.

Sebbene questo metodo abbia ottenuto ottimi risultati in letteratura, esso soffre di un particolare limite rappresentato dalla restrizione imposta dai linguaggi controllati circa la dimensione del vocabolario che è possibile utilizzare nella formulazione delle domande. In questo modo, durante l'analisi della domanda posta dall'utente, viene effettuata un'operazione di string matching tra gli elementi della domanda e quelli che risiedono nell'indice.

Questo metodo tuttavia presenta alcuni svantaggi, in particolare, si va contro quello che è uno degli scopi principali dei sistemi di Question Answering per dati strutturati: costruire un'interfaccia attraverso la quale l'utente può interrogare una sorgente dati senza conoscerne il contenuto o il linguaggio di interrogazione ad essa associato. Ad esempio si consideri la

domanda *Who is the author of the Neuromancer?*, la quale viene correttamente interpretata dall'automa in quanto coerente con gli stati previsti e con le risorse della KB. L'uso di un vocabolario così ristretto fa sì che una lieve modifica, come l'uso della parola *writer* come sinonimo di *author*, renda impossibile ricondurre il termine alla proprietà corretta e pertanto la query finale non può essere generata.

Per mitigare questo problema e cercare di colmare il *lexical gap*, cioè la distanza tra il vocabolario dell'utente e quello della KB, abbiamo deciso di sfruttare tecniche basate sulla semantica distribuzionale [Siciliani, 2018]. Utilizzando Word2Vec [Mikolov *et al.*, 2013] è possibile infatti creare una rappresentazione vettoriale dei termini contenuti all'interno di un corpus sulla base del contesto in cui esse si trovano, cioè sulla base delle parole che compaiono all'interno di un certo intervallo. I vettori così ottenuti possono essere utilizzati per determinare la similarità semantica tra parole diverse e ciò permette di gestire non solo i sinonimi, ma anche le flessioni di un termine.

Dato un token non riconosciuto attraverso lo string matching, viene calcolata la sua rappresentazione vettoriale e la sua similarità semantica con le etichette della KB. Se il valore di similarità supera una certa soglia, al token viene associata quella particolare risorsa. Sulla base del tipo di token e dello stato corrente, è possibile determinare lo stato successivo in cui transitare e quindi come procedere con l'analisi.

Il rilassamento del vincolo sul vocabolario imposto dal linguaggio controllato conferisce maggiore flessibilità al sistema ma, in alcuni casi può provocare lo stallo dell'automa a stati finiti. Si verificano infatti dei casi particolari in cui un certo token, pur rispettando i vincoli imposti dalle regole dell'automa, causa il transito in uno stato dal quale è impossibile raggiungere quello finale. Un esempio è rappresentato dalla domanda *What is the prize of Alain Connes?* In questa frase il token *prize* viene riconosciuto come una classe (`dbr:Prize`) e non come una proprietà (`dbr:award`) come dovrebbe. L'automa si trova a questo punto in una situazione di stallo in cui l'accettazione del token seguente, *Alain Connes*, che rappresenta un'entità, non è consentita dalle regole di transizione. Per evitare questo tipo di problematiche è necessario fare in modo che l'automa possa riconsiderare le scelte fatte in precedenza mediante una tecnica di backtracking. Ogni volta che viene analizzato un nuovo token, vengono memorizzate le prime dieci risorse candidate ad esso associate ordinate in base al loro grado di similarità semantica. Se l'esecuzione dell'automa si arresta prima di aver raggiunto lo stato finale, si ritorna allo stato precedente e si considera la seconda risorsa candidata. Questo processo può eventualmente ripetersi fino all'esaurimento di tutte le dieci risorse. In questo caso, l'automa retrocede ulteriormente e vengono così valutati i candidati dello stato precedente. Se nonostante siano state prese in considerazione tutte i candidati dei token precedenti l'automa resta in uno stato di stallo, il sistema non può restituire una risposta alla domanda e ciò viene opportunamente segnalato all'utente.

3 Caso di Studio

Secondo le norme attualmente vigenti in Italia, le pubbliche amministrazioni sono obbligate legalmente a rilasciare online i loro dati. Questi ultimi costituiscono un'importante risorsa, che spesso però non viene sfruttata dai cittadini in quanto difficilmente interrogabili. Come detto in precedenza, i sistemi di Question Answering nascono proprio come interfacce in linguaggio naturale, pertanto essi risultano particolarmente adatti in questi contesti per rendere tali informazioni facilmente fruibili dal pubblico.

Abbiamo dunque deciso di sfruttare i dati disponibili sul sito della regione Puglia[1], concentrandoci su quelli relativi al turismo. Dal momento che non vi sono regole stringenti riguardo il formato con cui i dati devono essere pubblicati, i dati raccolti sono stati modificati in modo da renderli conformi al livello 5 della classificazione fatta da Tim Berners Lee per la Linked Open Data (LOD) cloud. Uno degli elementi più importanti da tenere in considerazione nella creazione dei LOD è l'uso di vocabolari condivisi e di collegamenti ad ontologie che permettano di strutturare e descrivere i dati nella maniera più uniforme possibile. Per il nostro caso di studio, le ontologie che sono state scelte ed utilizzate per la modellazione dell'ontologia sul turismo in Puglia sono: DBpedia, Geonames, Geo, Schema, Km4city, Proton e FOAF. L'ontologia finale[2] ottenuta consta di 40 classi, 18 proprietà e 36355 istanze. Per valutare le perfomance del metodo proposto, abbiamo creato manualmente un dataset composto da 55 domande inerenti l'ontologia sul turismo in Puglia e abbiamo utilizzato il sistema descritto in [Mazzeo e Zaniolo, 2016] come baseline. SDDSqa riesce ad ottenere un'accuratezza che si attesta intorno all'84% contro il 62% riportato da CANaLI. Tra i lavori futuri vi è la pianificazione di un test in-vivo al fine di esaminare quali siano i risultati ottenibili usando SDDSqa in un contesto reale.

Riferimenti bibliografici

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, e Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[Mazzeo e Zaniolo, 2016] Giuseppe M Mazzeo e Carlo Zaniolo. Answering controlled natural language questions on rdf knowledge bases. In *EDBT*, pages 608–611, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, e Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Siciliani, 2018] Lucia Siciliani. Question answering over knowledge bases. In *ESWC (Satellite Events)*, volume 11155 of *Lecture Notes in Computer Science*, pages 283–293. Springer, 2018.

[Vrandečić e Krötzsch, 2014] Denny Vrandečić e Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[1] http://www.dataset.puglia.it

[2] https://github.com/swapUniba/tourismInApulia