**Knowledge graph**

**Knowledge graph e integrazione con NLP**

Sono risorse che hanno sempre più a che fare con il trattamento del linguaggio naturale. Che cosa è l’integrazione dei knowledge graph nell’NLP e vantaggi e svantaggi di questi.

Concetto generico: nei knowledge graph la cosa più importante è data dalle relazioni trai nodi del grafo. Questi dei knowledge graph sono concetti che derivano dai database a grafo piuttosto che i database relazionali. Nei database relazionali la cosa più importante è ricordare singole istanze di singoli oggetti definiti attraverso degli attributi, nei knowledge graph l’importanza è quella relazionale tra nodi. Uno schema diverso rispetto a quello dei record. Esistono diversi DB a grafo ma non andiamo nel dettaglio, ad ogni modo sono importanti dal punto di vista della scalabilità con enormi dati. In un DB relazionale le join sono costosissime con molti dati.

Knowledge graph: quando servono e perchè:

* Importanza delle relazioni, usare legami al posto di istanze
* Differenza dallo schema a record
* La trasformazione dei dati. Caratteristica importante perchè se non ho dei dati che hanno la naturale forma di network, come le connessioni tra utenti in un Social network, potrei usare uno schema a grafo.

Si può pensare ad un testo a tutto tranne che a un grafo ma attraverso la trasformazione dei dati posso tradurre un testo in un grafo. Abbiamo gia accennato degli esempi, nel momento in cui parlavamo di matrici co-occorrenza. Le parole sono legate ad altre parole, con una matrice quandrata con stesse features su X e Y è un grafo bipartito: connessioni tra nodi e gli stessi nodi.

**Perchè trasformare i dati?**

Dipende dagli obiettivi, dalle analisi. Se ho dei dati che non hanno una struttura a grafo ma l’aspetto relazionale è importante per una certa analisi, posso trasferire questa conoscenza in una struttura a grafi e applicare le metriche e le conoscenze della teoria dei grafi per analizzare i dati.

**Modello del knowledge graph:**

* **Nodi e relazioni**
* Ogni nodo e ogni relazione possono avere dei metadati associati, ad esempio in graphDB oppure in Neo4j ci sono proprietà associate chiave-valore associate ai nodi, ma dipendono dal framework.

Uniamo il contesto dei DB a grafo con il NLP. Si parla sempre di rappresentazione della conoscenza, sono che l’NLP è un qualcosa di fluido e complesso (ambiguità, polisemia...)........

Integrazione con NLP:

* **Con KB o KG** si intende knowledge base o knowledge graph:
* L’informazione codificata nelle KB e nei KG è una informazione gia strutturata che si spiega da sola. I nodi sono dei concetti che possono essere documenti o parole
* **Vantaggi:** Le relazioni sono tipate: hanno una etichetta e posso dire cosa è connesso con cosa leggendo il grafo con le etichette delle relazioni. La conoscenza è auto esplicante.
* **Svantaggio1**: Non tutta la conoscenza ha la forma di grafo, spesso si deve ricorrere ad una trasformazione dei dati che può portare a rumore e approssimazione.
* **Svantaggio2**: A volte servono quantità enormi di dati in input e facilmente interpretabili nei modelli di KG. Corpus enormi.

Dagli approcci neurali recenti, si riescono a catturare molte informazioni semantiche di tipo relazionale e con buone performance, ad esempio BERT.

La cosa principale è che molto spesso non serve usare reti neurali complesse per estrarre informazioni già codificate. Ad esempio sfruttando la conoscenza gia codificata in KG si riesce a capire se due concetti sono molto simili o meno. E’ questo il punto di aggancio di NLP e KG, ovvero che a volte, sempre più spesso, non è così utile andare a fare NLP avanzato quando della conoscenza è già proiettata in KG. Oggi una cosa importante recente è come coniugare i due ambiti di ricerca, perchè le KG non contengono tutto e NLP non è in grado di estrarre tutto e spesso trova cose che sono gia state estratte ed etichettate e codificate nei KG. E’ una area di ricerca interessante con sfumature che vanno nel deep learning.

**Utilizzi in NLP dei KG:**

* Calcolare la centralità ovvero la rilevanza, l’importanza di nodi rispetto ad altri relativamente alla loro struttura e al loro intorno strutturale. Quindi possiamo usare meccanismi della teoria dei grafi su dati che inizialmente non hanno una struttura a grafo
* Similarità tra nodi e relazioni
* Ricerca di percorsi migliori: più brevi tra 2 concetti
* Si possono calcolare degli embeddings, ad esempio il Node2Vec
* Si fa predizione: link prediction, vedere quali nodi che non sono direttamente collegati sono potenzialmente collegabili usando la loro struttura.
* Analisi delle comunità, quindi fare clustering su uno spazio a grafo.

Esistono 2 principali approcci metodologici ai KG:

* **Guidati dai dati:** analizzo i dati e vedo cosa tiro fuori e cerco di legare queste conoscenze con degli obiettivi specifici
* **Guidati dagli obiettivi:** parto da specifici goal, minimizzare il tempo, minimizzare risorse.

Considerano 2 punti di vista differenti ed entrambi utili.

Task di NLP in cui si usano KG:

* WSD: reti di parole che vengono costruite attorno alla parola da disambuiguare dato il contesto
* Question answering
* Ricerca semantica
* Reccomander systems

Grafo di Google usato nel motore di ricerca con dimensioni immense: +600

milioni di nodi +187 miliardi di fatti e relazioni.

Ha tanti scopi:

* trovare le giuste info (disambiguazione, Information Retrival)
* summarization
* anticipazione di nuove domande

Le risorse lessicali possono essre viste come un grafo, ad esempio il grafo di wordNet, nel mezzo c’è la root “entity” e poi ci sono altri nodi e relazioni iperonimiche e meronimiche, relazioni trasversali.