Graph-based Natural Language Processing

Introduction

Graph analysis is methodology for data analysis where the data is represented in the form of vertices (data entities) and edges (relationships between entities). Such data form enables discovery of valuable non-immediate information about the original dataset (usually in the form of relational tables or free-form text). For instance, PageRank is one of the most popular graph algorithm used by Google Search to rank the webpages.

(Source: Neo4j)

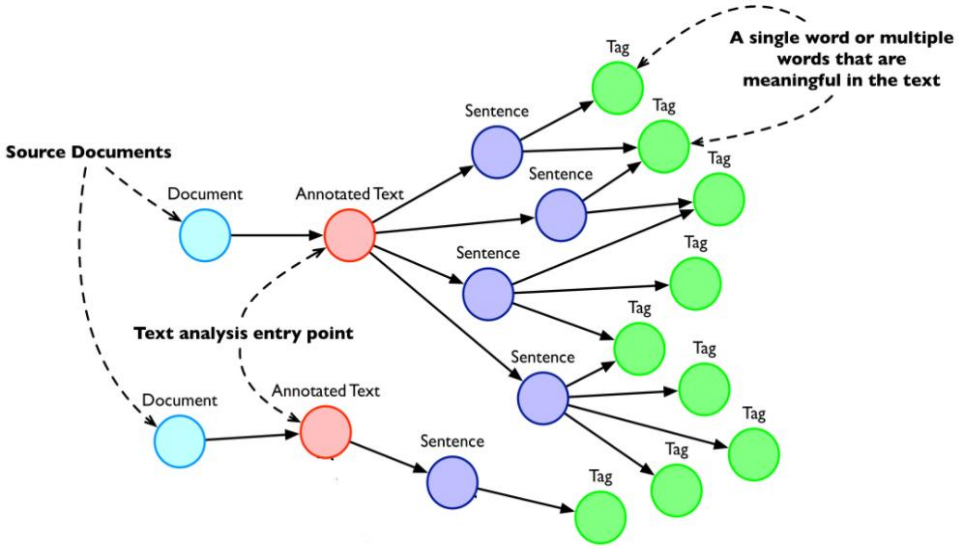
Graph databases are ideally suited for storing information about relationships among entities, for accessing diverse types of information, and for easily incorporating new information. Querying relationships is fast because they are perpetually stored in the database. Relationships can be intuitively visualized using graph databases, making them useful for heavily inter-connected data. These capabilities are a great match for the complex nature of today's information as well as the fast pace of market changes.

Text Mining techniques, including Natural Language Processing (NLP) and Information Retrieval (IR) provide the basis for harnessing large amounts of textual data and converting into useful source of knowledge for further processing.

Text Graph is the graphical representation of a text item. It is typically created as a preprocessing step to support NLP tasks. The graph-based methods focus on how to represent text documents in the shape of a graph to exploit the best features of their characteristics.

- **Nodes** in such a graph can include: tokenized words, sentences or paragraphs. They can also be used to represent semantic concepts.
- **Edges** can be used to indicate: mutual interaction between nodes (Directed Vs Undirected), relationships between words (Labeled Vs Unlabeled) or degree of node relationships (Weighted Vs Unweighted)

Models can be run on top of these text graphs that can be used in various application in the fields of Text Mining, NLP and Information Extraction etc.



(Source: GraphAware)

Graph Representations in NLP

Graph databases and Text Mining work well together because you can apply NLP to extract meaning from free flowing text and then store the results in a graph database to be further used for knowledge discovery and analysis

NLP tasks and use cases can be addressed with different types of graph representations as listed below:

- **Word co-occurrence graphs**: Represents context based word co-occurrences. Information from the graphs can be used to learn *word embedding's* and global vectors for word representation. The Nodes in this graph are labelled but Edges are not labelled.
- **Word-document graphs**: Information is encoded about the occurrence of a word at document level. Models such as *statistical topic models* can be built on top these graphs for retrieving important information. The Nodes in this graph are labelled but Edges are not labelled.
- **Sentence as graphs**: The relationship of syntactic and semantic dependency between words are encoded and represented in this form of graph. This type of graph can be used for wide variety of tasks like: *sentence classification, sentimental analysis sematic role labelling* etc. This graph has both Node and Edges labelled.
- **Knowledge graphs**: This graph represents encoding of different entities relationships. This type of graph is suitable for *Question-Answering* and *Search* tasks. In this case, both Nodes and Edges are labelled.
- **Phrase as Graph**: In this case the graph is represented as encoded with a minimal automata a large set of phrases. This type of graph can be used for: *Text Clustering, Plagiarism detection and Text Summarization.* The Nodes in this graph are labelled but Edges are not labelled.

Tools for Graph NLP:

1. Tool: Neo4j ; Language: Cypher

Neo4j is an ACID compliant graph database management system with native graph storage and processing. Along with traditional graph algorithms and pattern matching analysis, it also comes with proprietary data science library and couple of NLP libraries: APOC NLP and GraphAware NLP. The library supports text extraction, key word extraction, TextRank summarization, word embedding's using Word2Vec, and more

Neo4j graph data can be queried using Cypher which is declarative graph query language and pretty much similar to SQL. Sample Cypher query

Sample Cypher query for sentiment detection:

```
MATCH (t:MyNode)-[]-(a:AnnotatedText)
CALL ga.nlp.sentiment(a) YIELD result
RETURN result;
```

Explore more with Neo4j open source edition: https://neo4j.com/download/

2. Tool: Ontotext; Language: GraphQL/SPARQL

Ontotext platform is about making sense of text and data. It lets big knowledge graphs to improve the accuracy of text analytics and to enable better search, exploration, classification and recommendation across various domains.

Explore more with Ontotext platform: https://www.ontotext.com/products/ontotext-platform/

Conclusion

In this review, we explore how Graphs can be used in the fields of NLP and Information Retrieval (IR). Graphs provide a convenient way of querying and visualizing the results for the same and Graph databases can be considered a viable tool for mining and searching complex textual data. However only in recent years the applicability of graph frameworks to NLP became apparent and increasingly found its way into publications in the field of computational linguistics.

References:

https://neo4j.com/developer/graph-data-science/nlp/
http://platform.ontotext.com/index.html
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9088989
https://pdfs.semanticscholar.org/c326/673a38a7fc9378c7032ca10e61101505dc9d.pdf
https://github.com/graphaware/neo4j-nlp
https://graphaware.com/neo4j/2016/07/07/mining-and-searching-text-with-graph-databases.html