Problem Statement

Kensho aims to develop insights into financial markets, economic predictions, and trends within human society quickly and accurately. Natural language processing is an effective method for automatic parsing and understanding of online text. The project's focus is to utilize named entity recognition (NER), named entity disambiguation (NED) / named entity linking (NEL) to aid in Kensho's mission. The development of a knowledge graph using Wikidata will improve the accuracy of NED. After successful training on Wikidata/Wikimedia data, we plan to test the generalization of our model using Bloomberg or Yahoo! Finance articles.

Data

The following datasets are related to Wikipedia:

- **enwiki.k_link**: Dataset of hyperlinks on Wikipedia articles that link to other Wikipedia articles.
    - Hyperlink is on source page, while hyperlink links to target page.
- **enwiki.k_plaintext**: Dataset of the text of sections of Wikipedia articles.
- **enwiki.k_raw_anchors**: Dataset of the anchor texts of hyperlinks that link to other Wikipedia articles.
    - An anchor text is the actual clickable text (i.e. the blue text) of the hyperlink.
    - Useful for indicating synonyms/other names of an entity that is not the title of the target Wikipedia page if anchor text != target page title.
- **enwiki.page**: Dataset of Wikipedia pages.
- **enwiki.redirect**: Dataset of redirects.
    - Example: https://en.wikipedia.org/wiki/PRC redirects to https://en.wikipedia.org/wiki/China

The following datasets are related to Wikidata:

- **wikidata.item:** Dataset of Wikidata items.
    - Nodes of the knowledge graph.
- **wikidata.property:** Dataset of Wikidata properties.
    - Edges of the knowledge graph.
- **wikidata.qpq_item_statements**: Dataset of Wikidata triplets.
    - (source, edge, target)

We construct a knowledge graph from **wikidata.qpq_item_statements**. Almost every Wikipedia page has a corresponding Wikidata item. As such, these datasets are essential in producing a training and test set used for NED/NEL, as together they produce data of sentences and their corresponding entities.

Literature Review

One of the main research issues in bridging Knowledge Graph with textual data is entity linking (Ji et al. 2010). The entity linking system is a function $f : N -> E$ that maps name mentions N in a document to a set of entities E in a knowledge graph. The main challenges for entity linking are the name ambiguity problem and the name variation problem (Deng et al. 2018).

In a traditional entity framework (non deep learning approach), there are 3 main steps:

1. Name Mention Identification which can be done through Named Entity Recognition or dictionary-based matching. Dictionary-based matching constructs a name dictionary for all entities in the Knowledge Graph and matches all the names mentioned in a document. For this task, Stanford NER tagger is seen as the standard in NER (Dishmon).
2. Candidate Entity Selection which selects candidate entities for name mentions identified in a document. This is also where the name variation problem comes in.
3. Local Compatibility Computation to calculate the likelihood of each candidate entity for a given name mention. This is also the main component of the name ambiguity problem.

Guo et al. (2011) proposes a graph-based entity linking method with a simple local compatibility computation - for each name mention, we find the direct connections of candidate entities to all other entities mentioned in a sentence. The candidate entity with the highest degree is selected.

One problem with the traditional approach is that this system relies heavily on Local Compatibility Computation, which is limited by the weakness of feature engineering in incorporating the contextual evidence (Deng et al. 2018). Neural Network approaches have been developed by encoding contextual evidence in a vector space for the task of entity linking. In particular, knowledge graph can incorporated by representing a graph as (Cetoli et al. 2018):

1. a list of triplets ($x_i, e_{i,j}, x_j$), where x_i and x_j are the two node vectors of the name mentions, represented as GloVe word vectors, and $e_{i,j}$ is the edge computed from averaging the word vector in edge's label. The list of triplets can be processed by a Bi-LSTM.

2. embeddings by encoding the topology of the graph using Graph Convolutional Network (GCN) which stacks together N convolutional layers to propagate the features of nodes N hopes away.

The attention mechanism can also be added to reweight the output vector of graph representations. These additional context evidence is concatenated with the embedding for the input text to generate a binary vector indicating whether the input graph is consistent with the entity in the text.

Finally, we note that a joint model for entity tagging, coreference resolution and relation extraction can yield much better performance than individually performing each on a pipeline (Singh et al. 2013).

Our Approach

The problem can be separated into a 2-step process - named entity recognition (NER) and named entity linking/disambiguation (NED). We utilize the Stanford 7-class NER implementation available on *nltk* to identify all the entities (Location, Person, Organization, Money, Percent, Date, Time) within a subset of text extracted from *Wikipedia*. While the exact classification is not necessary, it will be useful in filtering our knowledge graph subsequently, saving on computational resources.

The focus of the project is on NED, where we leverage on *Wikidata* triplets to construct a knowledge graph of all entities mentioned within the text~, across a specified local word span. The contextual information of the surrounding words is thus captured within the knowledge graph. In this knowledge graph, there may be several repeated entities. For example, for the text, "**Mount Bromo** is a mountain located in **Java**", the entity **Java** might have 2 nodes, one referring to the location in Indonesia, and the other referring to the programming language. Using a distance/similarity metric, we can then utilize the knowledge graph to identify which is the **Java** referred to in the text.

We start with the simple approach of computing the number of direct connections to the other entities within the knowledge graph and selecting the entity with the highest degree as the correct entity. This would be a simple baseline for the project. Subsequently, we explore more complicated graph based ranking algorithms and, if time permits, some of the deep learning methods mentioned within the literature review such as Graph Convolutional Network.

We note that by leveraging knowledge graphs in NED, we essentially have identified the relation between entities mentioned within the text. As such, in a typical entity information extraction pipeline, we have already completed both entity tagging, and relation extraction. A natural extension would then be to identify all the coreferences within the text. Due to constraints, we may not be able to do so. However, if time permits, we would explore coreference resolution to complete a full information extraction pipeline that could be used by Kensho on any textual data.

Having built our model on *Wikimedia* data, we can then explore the generalizability of the model on other textual data. Specifically, we would explore the generalizability of the model on financial data, such as *Bloomberg* articles.

Reference

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In In Third Text Analysis Conference (TAC).

Li Deng and Yang Liu. 2018. Deep Learning in Natural Language Processing (1st ed.). Springer Publishing Company, Incorporated.

Guo, Y., Che, W., Liu, T. and Li, S., 2011, November. A graph-based method for entity linking. In Proceedings of 5th International Joint Conference on Natural Language Processing (pp. 1010-1018).

Cetoli, A., Akbari, M., Bragaglia, S., O'Harney, A.D. and Sloan, M., 2018. Named Entity Disambiguation using Deep Learning on Graphs. arXiv preprint arXiv:1810.09164.
Singh, S., Riedel, S., Martin, B., Zheng, J. and McCallum, A., 2013, October. Joint inference of entities, relations, and coreference. In Proceedings of the 2013 workshop on Automated knowledge base construction (pp. 1-6). ACM.