DIS-423 Project 3 · InfoExplorers

Minh Dinh Trong, Anton Balykov, Eric Saikali

14.12.2023

---

1 Introduction

Named Entity Disambiguation (NED) is the task of mapping entities, such as persons, locations, or companies, from a given text document to corresponding unique entities in a target Knowledge Base (KB). In this specific project, we want to associate a named entity with a relevant Wikipedia article. While the concept of Wikipedia autobiography is familiar, indexing a correct Wikipedia article that describes the entity precisely is not a trivial task. To tackle the ambiguity of named entities, in this project, we integrated textual aliases and redirects mapping with graphical data of Wikipedia articles. By leveraging this technique, we can resolve ambiguous entities not provided in the training set by leveraging the connectivity to other entities within the same document. The result is an efficient framework that maximizes the use of training data while autonomously discovering the article network in an unsupervised manner, offering an F1-score of 0.747 on the Kaggle partial testing set and 0.97 on our own cross-validation.

The implementation can be found at: Kaggle Link.

2 Description of Process

In this project, we first study the naive approach in Section 2.1. The insight gained from this experiment led to the final decision that is described in Section 2.2, which offers a robust NED framework by incorporating effective pre-processing, data-efficient usage of existing data, and low-complexity graph usage to capture relationships among Wikipedia articles.

2.1 Simple systems and decision leading to our algorithm

As a straightforward solution, the wiki_items dataset was employed to search for tokens from test dataset in it, replacing spaces with underscores and appending these titles to `https://en.wikipedia.org/wiki/`. After that, we also proceeded by capitalizing the titles from the above-mentioned dataset. These naive approaches resulted in scores close to 0.2, possibly due to a high False Positive rate, which in turn negatively impacted the Precision component of the F1-Score.

In addition, these solutions completely ignore all the other data available to us and any relationships between entities. Taking these limitations into consideration, our direction is towards a model that avoids predictions with low certainty and decides such certainty on textual data and relationships between entities.

2.2 Description of Final Algorithm

After considering at the available data we decided to implement the incremental algorithm, which will first consider the existing data, and then utilize relationships between entities and context in documents. Please find the illustration of the framework in Figure 1 of Appendix A.

2.2.1 Pre-processing

While the datasets provided were well-structured, the textual representation of entities as well as their relatively large size poses some challenges. To tackle these challenges, we implemented two key pre-processing strategies. Firstly, with word-represented entities, it is necessary to transform the textual names to their lowercase version. This facilitated the merging of Wikipedia items and aliases datasets with all possible references of each entity. Secondly, during Phase 1, our matching process relied solely on the training dataset and readily available titles/aliases/redirections. To optimize efficiency, we transformed these multi-dimensional datasets into simpler and more lightweight dictionaries, which accelerated data access drastically, from more than 8 minutes to just 2 seconds for the first part.

2.2.2 The existing data usage

The first step of our approach starts with the test set with all entities unresolved. We attempt to assign a URL for each of those entities by exact matching in the following order of priority: (1) full mentions in the training set, (2) Wikipedia article's title in wiki_items, and (3) English alias in item_aliases dataset.

If steps 2 or 3 give us any candidates, we only consider them if there is a single candidate. Then we search for the redirect in enwiki_redirects dictionary. The desired link is obtained with the redirect if it exists or just with the candidate itself. Otherwise, if the aforementioned steps returned either no match or more than 1 candidate for the token, we left those tokens unresolved, indicated with the original '?'.

Phase 2 of the framework will then carry on the disambiguation process to fill these unresolved gaps based on resolved entities with the Knowledge Graph approach.

2.2.3 Graph usage

After exhaustively processing textual datasets, we identified 735 entities that lacked article assignments. These entities include instances where only parts of full names are present (e.g., 'Vetterli' instead of 'Martin Vetterli') or cases involving multiple entities with similar names. It is evident that these names likely represent substrings of the correct article titles and can be easily retrieved.

To address this, the relationships between entities can be discovered and leveraged by constructing an Undirected Graph using the networkx library. In this graph, item_ids from the dataset serve as nodes, connected by uniform, undirected edges. As this construction process does not involve any training, we pickled this graph for faster access.

With the obtained graph, the relevance of potential articles can be decided based on retrieved entities from Phase 1. In essence, when encountering an unresolved entity, we identify all Wikipedia articles whose titles contain the entity's name as a substring. For each potential article, we calculate its average distance to all retrieved entities within the same document. The candidate with the lowest distance is then selected as the correct match.

To improve efficiency, we store candidate nodes to avoid costly string operations. We filtered out unresolved mentions with over 25 matches to prevent excessive computation of shortest paths between nodes. In such case, the unresolved mention is likely short and pertains to a longer mention in the same document and we can easily identify such reference without using the graph. To minimize variability from retrieved entities, we select a subset of 10 entities. This results in a framework that acknowledges the closeness of entities within the same document and complements the preceding phase with a mere worst-case complexity of $\mathcal{O}(N * 25 * 10 * m)$, where N is the number of unique unresolved mentions that have more than 1 word, and m is the computation cost to find shortest path length.

3 Results & Conclusion

3.1 Final Algorithm Results

The first step of our algorithm results in a score of 0.721 alone, while taking less than a minute when using previously saved dictionaries. We went from 9166 unassigned tokens to just 735.

The second step of our approach takes around 18 minutes due to Kaggle's limited CPU power and manages to reduce this number even more, up to 111 unassigned tokens.

After combining utilization of the existing data from several datasets available beforehand with the Knowledge Graph approach we achieved a high score of over 0.74. As a cross-validation attempt, we used the last 146 documents of training data as a testing set, applied the framework, and obtained an F1 score of 97% for recall of 100% and precision of 95%.

3.2 Conclusion

In this project, we tackled the task of NED within the domain of Wikipedia articles. To this end, we proposed a 2-phase framework which first utilize textual matching and then resolve uncertainty based on graphical connection among Wikipedia articles. Thanks to the careful design choices of data structures, algorithms, and optimization, this framework is efficient despite the substantial size of data sets. The outcome is a set of high-confidence mapping of Named Entity to relevant Wikipedia articles that has low false negative/ false positive rates. As an opening, please find in Appendix B, discussion about limitations and advantages of our algorithm.

A Entity-Link Graph Resolution

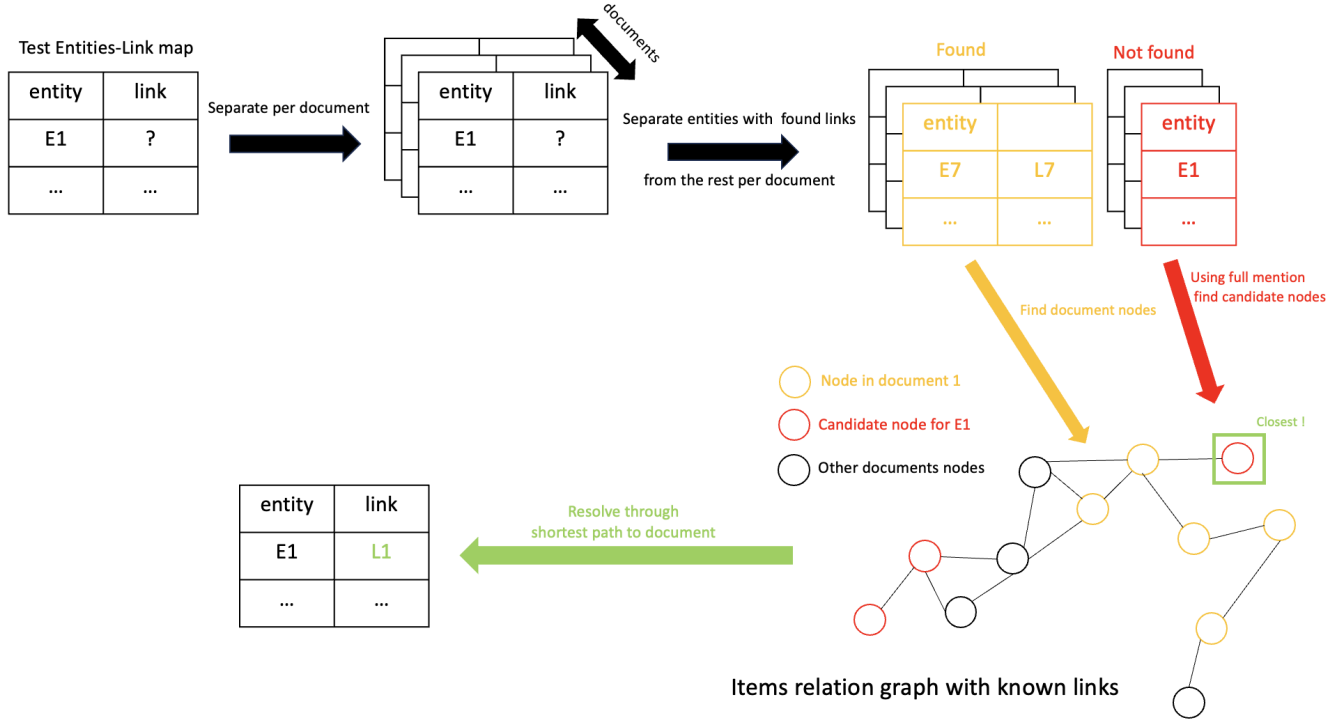

Figure 1: **Diagram of the proposed 2-phase framework.**
First, the model match entities based on existing textual data, returning in a set of highly confident retrieved Wikipedia links.
Second, based on the set of Wikipedia links, the unresolved entities are matched by leveraging relation graph.

B Discussion

B.0.1 Limitation of the "Brute force"

The naive solution of just taking titles from wiki_items dataset and URLs from training dataset is tempting yet too simple for the current setting. "Brute force" cannot take into account different aliases, relationships between entities and lacks context-awareness. What is more, scanning through big datasets will also take too much time for the current setting.

B.0.2 Advantages of utilizing the known data

To counteract the limitations of the naive approach and due to the abundance of the already known data, the good idea is to use that data directly before going deeper. The analysis has shown that nearly half of the unassigned tokens from the test dataset already exists in training dataset.

What is also true is that having the data collected from the Wikipedia itself it would be wise to look for the encounters of the desired entities in there.

B.0.3 Graph approach

For the task of NED the so-called Knowledge Graph approach can be rewarding. The Graph approach empowers NED systems to achieve significantly higher accuracy and performance by harnessing the interconnected nature of information. Its advantages are undeniable thanks to its capability of encapsulating relationships between entities and utilizing contextual information.