CS 410 Project Documentation

Team GYZ: Yuhan Guo(yuhang4), Qi Zeng(qizeng2), Haoduo Yan(haoduoy2)

1. An overview of the function of the code (i.e., what it does and what it can be used for).

We built an application for internet browsing. In the CS department's news of UIUC, we found that each professor's name mentioned in the article links to that professor's department faculty page. This enables readers to know that person's information within our university quickly. We extended this function to create links to Wikipedia for names in the article. This way, readers can view their published papers and life stories with a click. We performed web scraping using Natural Language Toolkit (NLTK) and applied Named Entity Recognition (NER) on the corpus to accomplish this goal. This tool helps identify the named entity on the CS Department's webpage then provides related Wikipedia links as references.

2. Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

Web Scraping:
Retriving articles from *cs.illinois.edu/news* and performing text processing to generate a 2D array with words and sentences separately. This step uses BeautifulSoup and Natural Language Toolkit (NLTK) in Python.

Named Entity Recognition:
Given a sentence, return the identified entity lists. Entity types include location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC).
Example

Input: After the first segment on Yugoslavia which is Episodes 1-4 the next segment is on Brazil (Episodes 5-8) and then Middle East (Episodes 9-12) .

Output: [{'entity': 'B-LOC', 'score': 0.99981433, 'index': 6, 'word': 'Yugoslavia', 'start': 32, 'end': 42}, {'entity': 'B-LOC', 'score': 0.9998037, 'index': 19, 'word': 'Brazil', 'start': 102, 'end': 108}, {'entity': 'B-LOC', 'score': 0.99967813, 'index': 29, 'word': 'Middle', 'start': 144, 'end': 150}, {'entity': 'I-LOC', 'score': 0.99898297, 'index': 30, 'word': 'East', 'start': 152, 'end': 156}]

Implementation:

We integrate bert-base-NER (https://huggingface.co/dslim/bert-base-NER) by building a processing pipeline. To accelerate processing we assign cache files for the pretrained models.

Entity Linking:
Given an entity, return its unique identity in Wikipedia if it existed.
Example:

Input: Cornell
Output: {'url': 'https://en.wikipedia.org/wiki/Cornell'}

Implementation:

This function is mainly based on url analysis. We filter those entities without a reachable wikipedia page by checking whether the url is redirected and whether the returned page is with contents.

Entity Extension:
Given a linked entity with its wikipedia identifier, return related entities. This function serves as the core function for this extension since it can return extra knowledge. We explore co-occurring entities (entities shown on the same wiki page).
Example:

Input: Cornell
Output:
[‘https://en.wikipedia.org/wiki/Ivy_League’,’https://en.wikipedia.org/wiki/Ithaca’]

Implementation:

This function is implemented with web scraping. For each extracted entity, we identify its out-linked entities with BeautifulSoup library and find.

3. Instructions on how to install and run the software.

1) Download the code from github
2) Run in a command terminal by typing: *python app.py*
3) Copy and paste the University of Illinois CS department news article’s URL from the browser to the software
4) The program will generate and output a list of names along with references to Wikipedia

4 . Brief description of the contribution of each team member.

Yuhan Guo(yuhang4): User Interface design; Application build

Qi Zeng(qizeng2) : Text processing; Named Entity Recognition/Linking/Extension

Haoduo Yan(haoduoy2): Web scraping; Data cleaning