Project Progress Report
CS 6120
Christopher Dilger c.dilger1@gmail.com
Taylor Murphy taylor.connell.murphy@gmail.com

Mapping Open Class Entity Relation Extraction to an Ontology

Introduction

Large amounts of factual information is disseminated online in unstructured form, which is both information dense and inherently difficult in a computational context apply data to real world problems. Information Extraction (IE) aims to create a structured representation of facts or information from natural language texts with many applications in Question Answering, Search and PII removal. Creating a general solution for representing entities in unrestricted free form text remains a very difficult problem.

Graph databases are a useful way to store and explore relationships between entities, but often require manually extracting relationship data from sources. For our project, we will attempt to use Named Entity Recognition (NER) and Relation Extraction (RE) in order to populate a database with entities as nodes and their relationships as edges. This database can then be used to easily visualize the relationships between entities in the text.

Overview

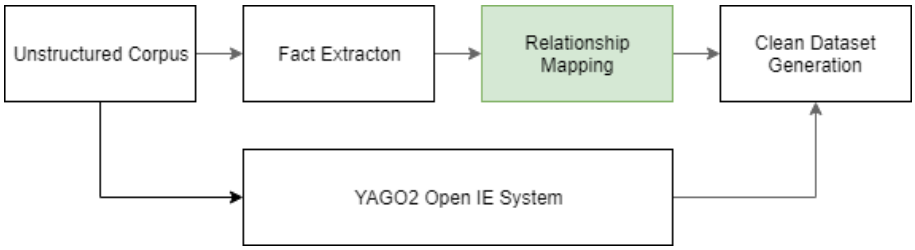

Figure 1: Scope of the contribution of our research to the field in context

Our current high level approach is as follows:

1. Clean the Wikipedia dataset
2. Extract entities and their relations using existing tools
3. Evaluate using the assumed presence of Wikipedia Infobox relations in YAGO to their corresponding Wikipedia pages
4. Using the entity-relationship-entity triples, generate records in graph database.
5. Generate visualizations of specific types of relationships

wiki_entity	wiki_relation	wiki_entity2	classifier	yago_entity	yago_relation	yago_entity2
Mr. Donald Trump	was elected president	the united states		Donald_Trump	isPoliticianOf	United_States

Figure 2: Task input and output, as a hand-crafted example of desired output

Aim

We seek to explore the building upon the existing Named Entity Relation extractor ReVerb by adding NLP post-processing steps to improve the quality of the extracted tuples. Conceptually we are building upon the recommendations of Fader et al.

Specifically classification models are constructed that have the goal of producing the following equivalences:

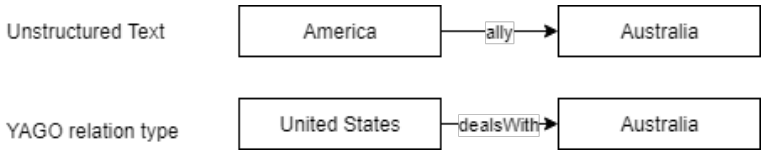

Figure 3: Model of the expected input and output of our specific model, and our contribution to Open IE

Related Work

Each of the subtasks in our pipeline have been explored in depth. Several major NLP tasks share common tasks (mostly NER). Approaches by both Hoffmann et al. and Wang et al. are similar in nature to our proposed approach. Using semi-supervised models some entity relation triples are extracted from Wikipedia in order to recreate the Wikipedia info boxes, which provide highly structured facts to evaluate against.

ReVerb is an Open Information Extraction tool designed to extract facts from a series of sentences. This work by the University of Washington will be incorporated into our pipeline at an early stage to build on the sentence based entity relation triple extraction.

Datasets

Datasets were chosen to balance data sparsity and computational feasibility with our resources.

- YAGO Ontology - using the 'Sample facts for popular entities' which is a 4MB subset of the entire 168GB dataset (as we lack the required disk space), will contain our target relational triples and will be used for evaluation.
- Our training data is from Wikipedia, and we will specifically be using the 420 pages used in our sample dataset from YAGO.
- We manually annotated $n = 278$ ¡relation1-relation2¿ pairs that were generated as part of our work to train our classifiers, which is available in the repo

The YAGO ontology is a set of facts built on top of Wikipedia using wordnet embeddings. YAGO is a high quality dataset of facts which have been extrinsically evaluated with a 95%

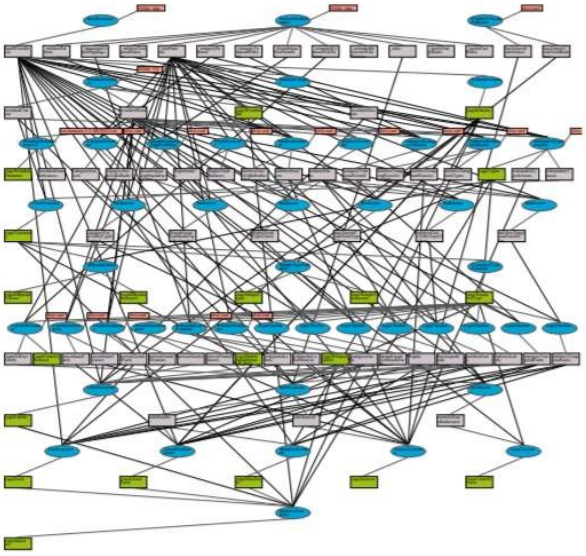

Figure 4: Dependency graph of YAGO[15]

accuracy. YAGO is a very complex system which uses the Wikipedia dataset to source data. The dependency graph for the full YAGO dataset is outlined in Fig. 3.

Due to limited access to disk space and compute, we chose to run our experiments on a subset of YAGO. The subset, "Sample Facts for Popular Entities" includes facts from 420 Wikipedia articles.

Data Preprocessing

Our data preprocessing steps are summarized sequentially below:

- Download sample set of popular wiki pages from YAGO
- Extract list of article titles from the yagoWikidataInstances.tsv file.
- Using Wikipedia's export feature, get a single xml file with all of the article bodies.
- Use python script from the Wikiextractor return a plain text version of all the pages.
- Run ReVerb on the output to generate $entity1, rel, entity2$ triples
- Lemmatize, find the stem words, remove stop words and format words to be compatible with YAGO

Since the data processing steps are rather intensive, we've included a zip file on our repo that contains the plaintext for each article in a file titled with the article's name.

1 Experiments

We tried several models, ranging from simple to complex using these average word embeddings of the hand annotated relationship equivalence table. 10 different models were tested. Our models during selection were evaluated using both MSE and a generalized Pearson's Correlation coefficient in order to choose the final model for our evaluation.

1.1 K-Neighbors Classifier

After tuning, k=3 returned an $MSE = 7.53$ which was the best performing non neural network model and a suprising result.

1.2 Random Forest Classifier

For random forest we chose to restrict the splitting as much as possible in an attempt to get the model to generalise well, as it is obvious we have a very sparse training set. The following parameters achieved an $MSE = 10.87$

Tree Depth	5
Estimators	10
Maximum Features	1

1.3 AdaBoost Classfier

Using 50 Decision Tree classifiers with depth of 1 and a learning rate of 1.0, an $MSE = 13.17$ was achieved.

1.4 Naive Bayes

A Gaussian Naive Bayes was used, and despite no expectation that the Word2Vec embeddings would be normally distributed per class the classifier compared well to the others tested with $MSE = 9.16$

1.5 SVM Classifier

With both Linear and RBF kernels, SVM models achieved $MSE = 14.61$ and $MSE = 11.91$ respectively with $C = 0.025$ and $C = 1, \gamma = 2$ for each respective model.

1.6 Neural Network

We setup a Neural Network with 3 layers, with 300 nodes, 50 nodes and a softmax activation function to predict 1 of 12 classes. In order to favour a generalized model, a high dropout rate of 80% was selected. All predictions with below a 50% probability were classified out of class. An accuracy of 40% was achieved across all classes with an $MSE = 0.0221$ which vastly outperforms all other models tested.

Evaluation

Inherent difficulties in the evaluation of open class problems became readily apparent upon commencement of this project. Results presented here are an attempt to rectify difficulties in evaluation of a new ontology by comparing with the existing YAGO ontology.

We found that taking direct $entity1, rel, entity2$ relations from our baseline IE Extraction methods returned 0 matches which is hardly a meaningful result. We present the intrinsic and extrinsic evaluations of our method.

1.7 Intrinsic Evaluation

Matching exactly with the YAGO factbase we achieved an accuracy 0.43% for a subset of 10 classes of the YAGO facts that we expected to be in the Wikipedia pages. This result is both exceptionally low and non-trivial. This result means that character by character matches of $entity1, rel, entity2$ were found between YAGO and our model. Though the matches are literal, there is no guarantee this result does not record false positives. Also, there is no guarantee that all of the YAGO facts tested against were in the corpus.

1.8 Extrinsic Evaluation

55583 triplets were generated in total, of which we took 10 unique random samples of 100 (n=1000) and at the $\alpha = 0.05$ level we assert that the accuracy is between 15.63% and 32.77% with a mean accuracy of 24.2%. This is low when compared to the ReVerb published accuracy, or the YAGO published accuracy. We are suprised that published precision of ReVerb is "More than 30% of REVERBs extractions are at precision 0.8 or higher" Fader et al..

2 Summary

Our model took a novel approach to simplify the YAGO pipeline, with mixed results. The pipeline created improves on existing relationship extraction tools to create higher quality output. Our contribution highlights some areas in which the research is lacking, especially in the disambiguation of named entities, long distance relations and non-spurious relations. This accuracy of 0.43% represents a meaningful improvement over the existing methods we built upon.

2.1 Improvements

Several steps can be taken to extend our project and provide more useful fact and relationship extraction from an unstructured and open corpus.

- Pronouns coreference resolution: Another issue with the output of ReVerb is that it treats pronouns naively (almost all entities have some relationship with the entity "it"). We may try adding a coreference resolution step before running ReVerb to collapse entity references.
- Data Cleaning: There are two major issues here. One is that Reverb seems to have issues with special characters and punctuation. The other is that many of the nodes output from Reverb are similar (ex U.S. vs. United States of America vs. The United States), but our model is unable to identify that these nodes should be merged. To solve this, we could add an entity disambiguation step to merge identical nodes.
- Remove non-verb phrase relations
- Investigate disambiguation of entities

References

[1] Google AI blog: 50,000 lessons on how to read: a relation extraction corpus, . URL https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html.

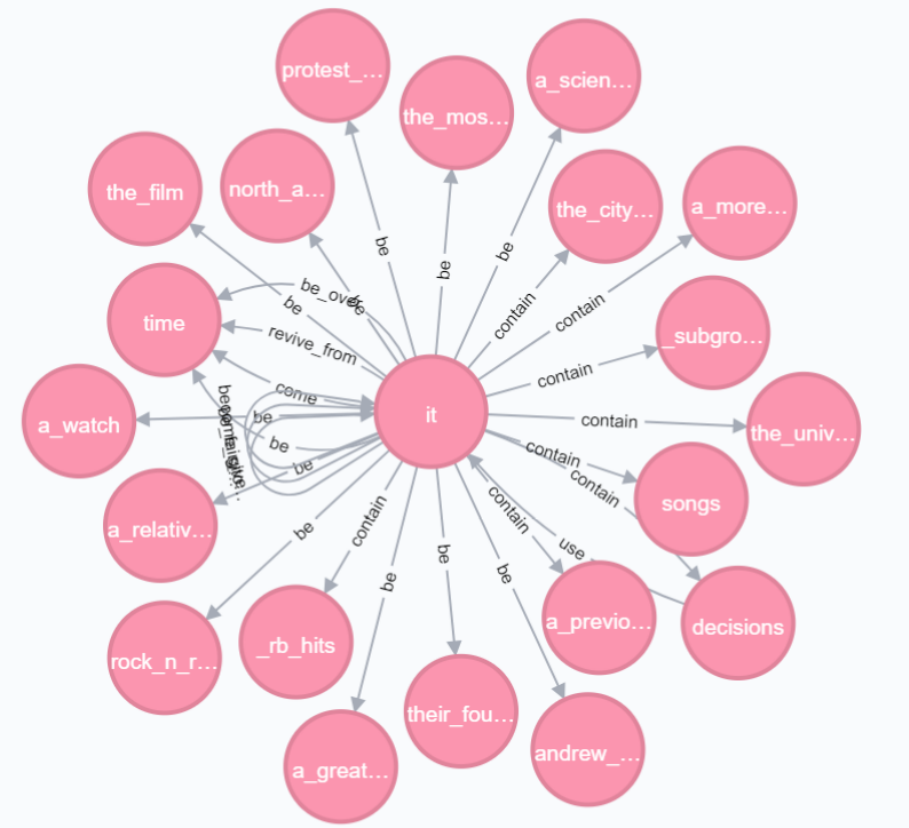

Figure 5: Pronouns contain a lot of information which is difficult to disambiguate

[2] Language-independent named entity recognition (II), . URL https://www.clips.uantwerpen.be/conll2003/ner/.

[3] Giuseppe Attardi. A tool for extracting plain text from wikipedia dumps: attardi/wikiextractor. URL https://github.com/attardi/wikiextractor. original-date: 2015-03-22T12:03:01Z.

[4] Nguyen Bach and Sameer Badaskar. A review of relation extraction. page 15.

[5] breckbaldwin. Coding chunkers as taggers: IO, BIO, BMEWO, and BMEWO+. URL https://lingpipe-blog.com/2009/10/14/coding-chunkers-as-taggers-io-bio-bmewo-and-bmewo/.

[6] Oren Etzioni, Anthony Fader, Janara Christensen, and Stephen Soderland. Open information extraction: The second generation. page 8.

[7] Explosion AI. SPACY's ENTITY RECOGNITION MODEL: incremental parsing with bloom embeddings & residual CNNs. URL https://www.youtube.com/watch?v=sqDHBH9IjRU.

[8] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. page 11, .

[9] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. page 11, .

[10] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. 194:130–150. ISSN 00043702. doi: 10.1016/j.artint.2012.04.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0004370212000446.

[11] Maximilian Hofer. Deep learning for named entity recognition #1: Public datasets and annotation methods. URL https://towardsdatascience.com/deep-learning-for-ner-1-public-datasets-and-annotation-methods-8b1ad5e98caf.

[12] Raphael Hoffmann, Congle Zhang, and Daniel S Weld. Learning 5000 relational extractors. page 10.

[13] Mitchell Marcus Eduard Hovy Sameer Pradhan Lance Ramshaw Nianwen Xue Ann Taylor Jeff Kaufman Michelle Franchini Mohammed El-Bachouti Robert Belvin Ann Ralph Weischedel Houston, Martha Palmer. OntoNotes release 5.0 LDC2013t19.

[14] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1030.

[15] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. YAGO3: A knowledge base from multilingual wikipedias. page 11.

[16] Diego Molla, Menno van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58. URL http://www.aclweb.org/anthology/U06-1009.

[17] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. 30(1):3–26. ISSN 0378-4169, 1569-9927. doi: 10.1075/li.30.1.03nad. URL https://www.jbe-platform.com/content/journals/10.1075/li.30.1.03nad.

[18] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge unifying WordNet and wikipedia. page 10.

[19] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. Relation extraction with relation topics. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1132.

A Appendix

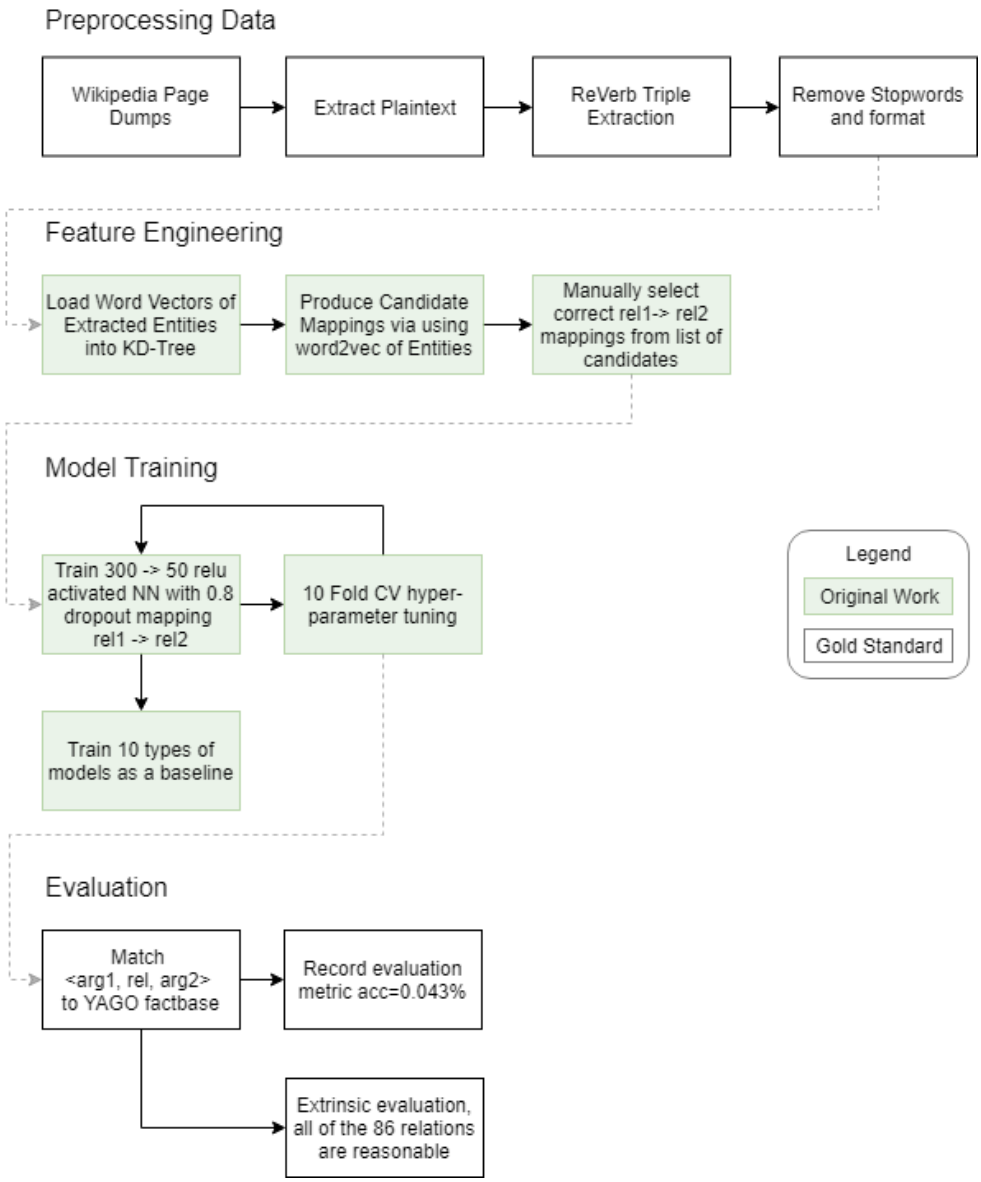

Figure 6: Entire Project Pipeline

B Appendix

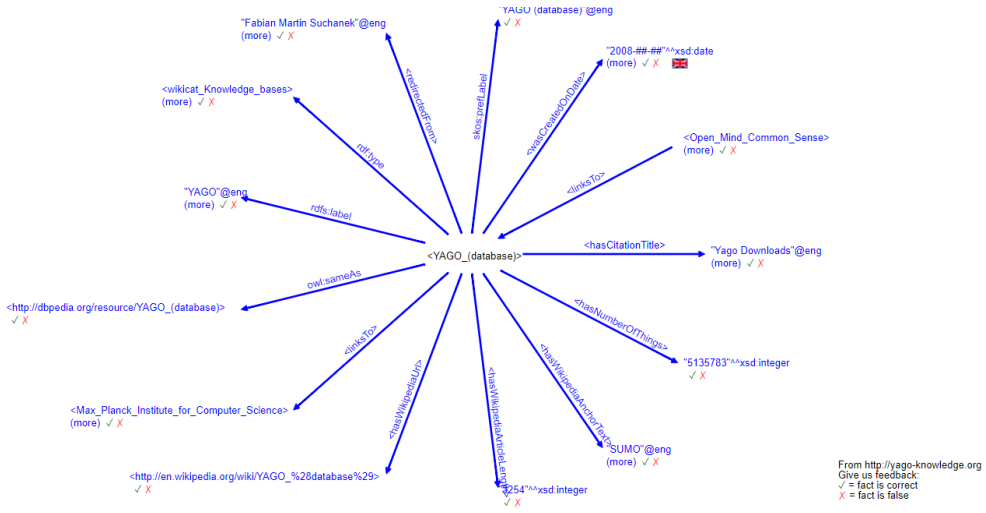

Figure 7: YAGO in fact has an online interface to a similar representation of facts similar to the way we set out to at Graph Browser