Natural Language Processing

Information Extraction Project

Filling Templates as Complex n-ary Relations

Edward Hong

exh150330

5/4/2020

Problem Description

The problem presented in this project is to implement an Information Extraction (IE) program using natural language processing and machine learning techniques to fill Information templates, which are predefined n-tuples relating n Named Entities, from a corpus of Wikipedia articles. As an example one such template defined in this project is the WORK template, which is a 4-tuple defined as (Person, Organization, Position, Location).

For any given input text document, the IE program should be able to extract any and all relevant templates defined by the program designer. In this case, three templates are defined:

1. BUY (Buyer, Item, Price, Quantity, Source)
2. WORK (Person, Organization, Position, Location)
3. PART (Location, Location)

Approach

In this report I propose an approach to filling information templates using a method presented by Ryan McDonald et al. (2005) for extracting complex n-ary relations. In brief the method is a three-step approach:

1. Extract binary relations amongst named entities within the text.
2. Construct a weighted entity graph G = (V, E) where the vertices represent the named entities and an edge (u, v) exists iff there exists a binary relation between entities u and v. The weights of the edges are assigned as the scores of the binary relation predicted between u and v in step 1.
3. Having constructed the weighted entity graph G, find maximal cliques in the graph which have satisfactory clique weight, and assign those maximal cliques as templates / complex relations.

We can see that an advantage to this approach is that we're able to insert and substitute any binary relation classifier of our choice for step 1. There is currently a wealth of research and numerous open source projects aimed at solving the binary relation extraction problem, and being able to leverage previous work and progress made in binary relation extraction makes the task of constructing n-ary relations significantly simpler, and was a prime motivation in the approach of this project.

From binary relations between entities, we are able to reconstruct and build complex relations between multiple entities using an intuition that a maximal clique, on average, represents a valid complex relation. In other words, nodes that are strongly related to each other are likely to form a template.

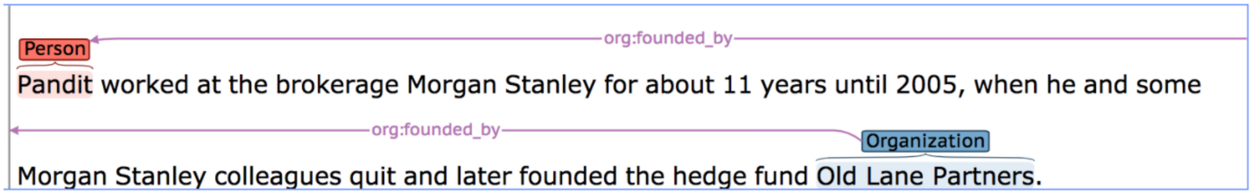
However, there are some caveats. We have yet to define what a “satisfactory clique weight” is in step 3. Ryan McDonald et al. (2005) assign a weight to a clique C using the geometric mean weight of all edges within the clique.

$$w(C) = \left(\prod_{e \in E_C} w(e) \right)^{1/|E_C|}$$

This essentially allows us to define a certain tolerance for how strongly related entities are via their pairwise binary relations. We employ a basic heuristic and define that a clique C is a valid clique and represents a valid template if w(C) ≥ 0.5. However this tolerance can be tuned to fit one’s specific use case or domain.

Lastly, one additional consideration is the decision of which entities should fill which slots within the template. For the purposes of this project, human inference and best judgment was used based on the nature of the binary relations identified.

As an example, consider a relation called ‘founded_by’. The image below depicts the HEAD and TAIL entities within this binary relation, the HEAD being Pandit, a person, and the TAIL being an organization called Old Lane Partners.



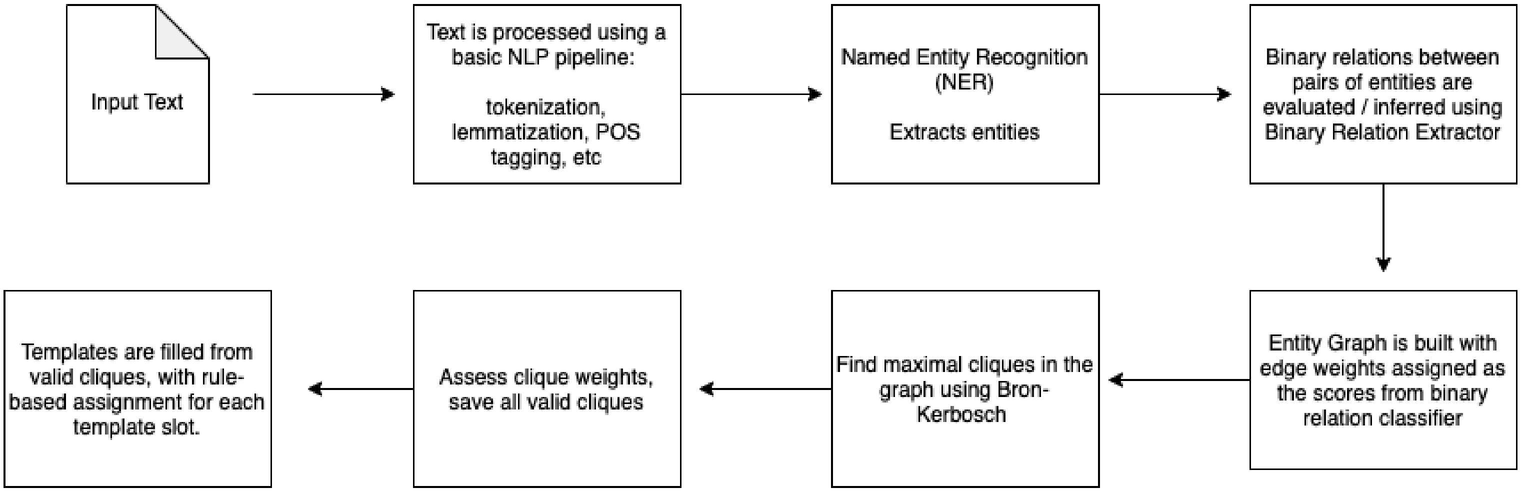
Pandit worked at the brokerage Morgan Stanley for about 11 years until 2005, when he and some Morgan Stanley colleagues quit and later founded the hedge fund Old Lane Partners.

Image courtesy of the Stanford TACRED project: https://nlp.stanford.edu/projects/tacred/#examples

We understand that this relation is between a PERSON entity and an ORGANIZATION entity. Then, for complex relations/templates we define what constitutes a valid relation by considering the entity types present within the binary relation.

As an example within this project, the WORK template has a slot for a person. One of the binary relations defined within the model is the ‘work location’ relation, which contains as the HEAD a person and the TAIL an organization or location. When a maximal clique of satisfactory weight is found and contains a ‘work location’ relation, the PERSON slot within the WORK template is filled using the HEAD of the ‘work location’ relation. The TAIL of the ‘work location’ relation may either be assigned to the ORGANIZATION slot of the WORK template, or the LOCATION slot, depending on its entity type.

Implementation


Input Text
Text is processed using a basic NLP pipeline:
tokenization, lemmatization, POS tagging, etc
Named Entity Recognition (NER)
Extracts entities
Binary relations between pairs of entities are evaluated / inferred using Binary Relation Extractor
Entity Graph is built with edge weights assigned as the scores from binary relation classifier
Find maximal cliques in the graph using Bron-Kerbosch
Assess clique weights, save all valid cliques
Templates are filled from valid cliques, with rule-based assignment for each template slot.

Architecture / Flow Diagram

Tools Used

Binary Relation Classifier

This project makes use of an open source dataset and training framework for training the Binary Relation classifier called OpenNRE from Xu Han, Tianyu Gao et al. (2019)

The project can be found on Github at: https://github.com/thunlp/OpenNRE

Internally the model makes use of a PyTorch implementation of a Softmax Neural Network. In addition for this project, the Bidirectional Encoder Representations from Transformers (BERT) encoder was used as the sentence encoder for the model.

Released as part of the OpenNRE project is their wiki80 dataset, containing approx. ~50,000 annotated examples from Wikipedia articles to classify relations amongst 80 different classes such as ‘work location’, ‘residence’, ‘owned by’, etc.

Detailed instructions and examples on how to use their training framework to train your own custom models is also provided. In this project I extended the wiki80 dataset with an additional ~4,000 annotated examples and trained a custom model which defines 8 additional relations to fit my use case for identifying BUY templates. The new relations I added are binary relations which I believed would be useful in identifying instances of BUY templates, such as ‘buyer of item’, ‘item bought’, ‘item price’, ‘item quantity’, ‘seller of item’, etc.

Basic NLP Pipeline

For the NLP pipeline I used spaCy, an open source Python framework for numerous fundamental NLP tasks such as tokenization, Part of Speech tagging, and of particular significance in this project, Named Entity Recognition.

The spaCy project and documentation can be found at: https://spacy.io/

spaCy was also used in development of auxiliary programs within this project, namely an annotation tool I'd written which provides a basic interface to help me annotate the given Wikipedia texts for additional examples to train the binary relation classifier.

WordNet

For project tasks that required use of WordNet (namely Task 1), the NLTK Python package was used to extract semantic relations such as Holonymy, Hypernymy, etc.

The NLTK project and documentation can be found at: https://www.nltk.org/

Other projects explored

Stanford TACRED

Large-scale binary relation extraction dataset of 106,264 examples. The dataset was released under the Linguistic Data Consortium and not used in this project. However, as part of the TACRED project, a new type of position-aware recurrent neural network model was developed by Yuhao Zhang et al. (2017), which was studied as research for this project.

The TACRed project can be found at: https://nlp.stanford.edu/projects/tacred/

Problems encountered

Unsurprisingly, the most apparent disadvantage and constraint of this approach is the necessity of large volumes of training data, and *good* training data. In addition, careful consideration and deliberation is needed in defining the binary relations which will compose the complex relations. For this project, no suitable binary relations were found within the 80 relations defined by the wiki80 dataset for the BUY template, so new relations had to be defined and trained for. This of course necessitates more training data.

I extended the dataset with ~4,000 additional annotated examples which were focused and targeted on binary relations which were relevant to the BUY template. To help with the annotation process, a simple Python program was created which processed the training text and allowed the user to indicate whether a relation was present within a given span of text, and which tokens were the HEAD and TAIL of the relation. However, another difficulty in training had become apparent whilst annotating: inter-sentence relations and coreference resolution.

The vast majority of binary relation extraction systems learn and operate at the sentence level, and the model used in this project is no exception. This added an extra complication while annotating, as many of the sentences within the training corpus utilized coreference. It was common to have a named individual, such as Abraham Lincoln, be addressed by name in one sentence, and in all following sentences within the paragraph, be referenced as “He”.

Due to time constraints for this project, I omitted such sentences where the coreference would need to be resolved from the training examples. However it should be noted that more sophisticated coreference tools may prove worthwhile for this application.

Lastly was the issue of Named Entity similarity. While the NER was able to recognize that “Abraham Lincoln” was likely a Person, abbreviating his name to just “Lincoln” proved ambiguous for the classifier. At times the classifier was unable to determine that “Lincoln”, “Abraham”, and “Abraham Lincoln” referred to the same entity. Ludovic Jean-Louis et al. (2011) describe solutions to address this problem, and while they were not implemented in this project, they serve as worthwhile areas for future work.

Further Improvements

Ludovic Jean-Louis et al. (2011) describe additional methods for the task of template filling, of which include ranking and selecting the most relevant entities to best fill the slots of the template. In addition, they also describe an approach for event-based segmentation to select the most relevant portions of text with respect to a given event or template.

These methods look to be promising avenues for further improvements to the Template Filling approach proposed in this report. The major shortcoming of the implementation used within this project is the inherent intra-sentence nature of the binary relation model. Ludovic Jean-Louis et al. (2011) present techniques for information extraction at the discourse level, which serve as a natural next-step for more sophisticated IE applications.

References

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In ACL 2005, pages 491 – 498, Ann Arbor, Michigan, USA.

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, Maosong Sun. 2019. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. In ACL 2019, pages 169 – 174, Hong Kong, China.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In ACL 2017, pages 35 – 45, Copenhagen, Denmark.

Ludovic Jean-Louis, Romaric Besancon, Olivier Ferret. 2011. Text Segmentation and Graph-based Method for Template Filling in Information Extraction. In ACL 2011, pages 723 – 731, Chiang Mai, Thailand.