Unsupervised Relation Extraction Using Word Embeddings

Abdi-Hakin Dirie (abdihd@mit.edu)
Emily Kellison-Linn (ekl@mit.edu)
Jason Tong (jktong@mit.edu)

Abstract

There exist many structured databases, such as DBpedia, which have been manually curated to list relationships between known named entities. However, information is available in these databases only after humans have ingested the knowledge and manually regurgitated it into structured tables. We attempt to automate the process of determining relationships between named entities, so that a particular relationship previously unknown by the system, can be discovered between entities mentioned in the same article. We plan to achieve this by learning what we call relation vectors for a variety of relations, and applying those vectors to pairs of named entities found in an input article.

I. Problem Statement

Given an input document with several mentions of named entities (for example, a news article), we wish to identify pairs of entities which have a meaningful relationship between them. We will return such pairs as a list of tuples. For example, let's consider a document D that has the following piece of text: ”The Massachusetts Institute of Technology is a research university based in Cambridge, Massachusetts”. The system should return a list of tuples similar to the following:

1) (Massachusetts Institute of Technology, Cambridge)
2) (Massachusetts Institute of Technology, Massachusetts)
3) (Cambridge, Massachusetts)
4) (Massachusetts, Cambridge)

Note that the name of the relationship is not returned, just the subject and object of some hypothesized relation. However, we would like the tuples extracted by the system to be representative of some semantic relation a human might infer. For example, tuple 3 is representative of the *capitalOf* relation, though the system does not have a name for it.

There are several subproblems that must be solved to succeed at the task as described. First, the system needs to discover what relations are worth looking for. We wish to accomplish this step completely unsupervised (i.e. without using publicly available databases like DBpedia or Freebase). Second, the system needs to find the named entities in the article. Third, the systems needs to generate candidate tuples. Fourth, the system needs to determine if a candidate tuple is an instance of some relation found in the first step. Some of these steps have existing solutions which we will incorporate, while others will require original work.

The general strategies for these four subproblems will be outlined in section III.

II. Related Work

Relation extraction is key to many information retrieval systems. Natural Q&A systems usually convert the question into a structured query that captures the essence of the question. For example, ”Where is MIT?” is represented as *locatedIn*(MIT, **?**). The structured query then simply consists of a look-up in a knowledge base to find out what **?** is. Our project attempts to build that knowledge base in an unsupervised fashion.

There have been such unsupervised systems proposed in the past. Some use strict string matching based on surrounding contexts using regular expressions [2]. Later systems still use contexts, but employ clustering to allow for variations between similar contexts [1].

We aim to do better by using word embeddings (also called word vectors). Various implementations to learn word embeddings have been proposed, and recent developments have shown that word embeddings generated by neural network models have very interesting linguistic properties [3]. We will be using such word embeddings in our project.

III. Technical Approach

We wish to produce an unsupervised model capable of identifying common relationships between named entities in a given input text. This is the process which we will follow:

1) Gather a list of several thousand named entities by running Stanford’s CoreNLP NER (Named Entity Recognizer) tool on a subset of the New York Times Annotated Corpus.
2) Retrieve word vectors for each of these named entities using Google’s word2vec algorithm pre-trained on Freebase. Some of these entities may not have word vectors if they don’t exist in Freebase; this is acceptable, as long as we have a significant number of word vectors to work with.
3) Perform vector subtraction on each pair of named entity word vectors. This subtraction produces a relation vector representing the relationship between the two entities.

4) Run a clustering algorithm (e.g. k-means) on the set of relation vectors. Our expected result is that pairs of entities whose relation vectors are in the same cluster are instances of the same relation. We expect that many of these clusters will correspond to human-understandable relationships, such as the country-capital relationship.
5) Calculate a representative relation vector for each cluster, which could be the average of all vectors in the cluster. The result will be a set of representative relation vectors, one for each notable relation.
6) Once we have the relation vectors, we can apply them on a new set of documents. We will run Stanford's NER tool on a different (non-overlapping) subset of the New York Times Annotated Corpus, calculate the relation vector for each pair of named entities in each article, and compare these to each representative relation vector, and output any pair which is sufficiently close to one of the identified relations.

IV. End Product

Our anticipated end product is a system which takes as input a chunk of plain text, and outputs a list of pairs of entities which the system believes are related.

V. Evaluation Metrics

As a sanity check, we should be able to consider the clusters of relations that arise as semantically significant relations and judge them with human intuition. This will be important in the initial stages of setting parameters and determining, for instance, what the threshold is for a cluster size before it can be considered significant.

Once we have ascertained that the clusters discovered are reasonable to some degree, we can apply more quantitative metrics such as precision and recall. We would find a test set of articles that contain instances of the most promising relations (those with the tightest or most populous clusters), and manually extract the entity pairs we as humans would consider valid examples of the relations of interest. We can then use the representative relation vectors to automatically extract them from the test-set and determine the system's precision, recall, and F1 score.

VI. Conclusion

This project analyzes the merits and challenges of using word vectors and their differences to generate vectors representative of particular relationships between named entities. We hope that this unsupervised process will allow systems to learn new relationships and be able to find them in texts automatically. However, one shortcoming of such a system is the continued dependence on word vectors. If a word vector is unavailable for a given named entity, then it becomes impossible for our system to identify any relationships with that entity. Furthermore, because of our independence from context, we are really only concluding that certain relations are present in a given text, as opposed to the stronger claim that the text actually provides evidence for a particular relation.

To combat this, we propose a possible extension to the system. After determining representative relation vectors, we can begin to use them for training an n-gram model for extracting relations from the text. From a high level, the approach would involve keeping track of the context in which pairs of entities co-occur. For a given relation, we can then cluster the contexts to determine if there are contextual patterns that reliably indicate the relation between two named entities. If this approach is successful, then we have adapted our model to be more robust and capable of learning new instances of relations between entities unknown to our system.

References

[1] Eugene Agichtein and Luis Gravano. “Snowball: Extracting Relations from Large Plain-Text Collections”. *Proceedings of the fifth ACM conference on Digital libraries*, 2000.

[2] Sergey Brin. “Extracting Patterns and Relations from the World Wide Web”. *In Proceedings of the 1998 International Workshop on the Web and Databases (WebDB98)* , March 1998.

[3] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. “Linguistic Regularities in Continuous Space Word Representations”. *In Proceedings of NAACL HLT*, 2013.