Natural Language Processing Final Project Write-up

Nicholas Marton

October 28, 2015

1 Project Overview

1.1 Executive Summary

Named Entity Recognition (NER) is the task of locating and classifying pieces of text into pre-defined categories such as the names of persons, locations, expressions of times, organizations, quantities, and monetary values among others. Furthermore, NER is a principle sub-task of information extraction, one of the most difficult yet pragmatically important tasks in all of Natural Language Processing. In order to make any considerable headway in task of information extraction, it is then necessary to succeed in NER.

One of the most daunting yet important issues in NER is cross domain generalization; most NER systems do not perform well at all when tested in novel domains. Additionally, tuning an NER model to a new domain takes considerable time and effort. I will investigate a new approach to NER that may mitigate this problem. By training a Deep Belief Network (one of the core models employed in the field of Deep Learning) for classification of Named Entities and then extracting the intermediate layers, I hope to obtain a small set of abstract, higher-level features (from the layers closest to the output layer of the DBN) that successfully generalize across domains. Additionally, I will use these features as input to a Conditional Random Field in an attempt to achieve state-of-the-art performance in Named Entity Recognition. Furthermore, I will provide a framework which will allow easy extraction of arbitrary intermediate layers from a trained DBN along with functionality to automatically feed these layers to a CRF model.

1.2 Goal

The goal of this project is to produce a model capable of state-of-the-art performance in Named Entity Recognition by using a Deep Belief Network to generate useful features that will be used by a Conditional Random Field during the classification process. Assuming a vector representation of the words to be classified (or the concatenation of multiple word vectors derived from an N-gram for instance) captures enough information to derive expressive higher-level features, this project may result in a small set of powerful explanatory features that can be reused across different NER applications.

1.3 Background and Motivation

The ability to detect and recognized named entities in large bodies of text is incredibly important throughout Natural Language Processing. The process of doing this so called Named Entity Recognition, also provides a foundation for the task of Information Extraction, arguably one of the most important tasks in all of NLP. Named Entity Recognition as a task has been approached in various ways: state-of-the-art systems are usually hand-crafted by computational linguists that use grammar-based techniques or created by employing statistical models (e.g. traditional Machine Learning methods) trained over domains consisting of enormous datasets. Despite the success both approaches have seen, research indicates that even state-of-the-art NER systems are brittle; typically, an NER system trained for some domain does not perform well on others. Tuning NER systems to new domains also requires considerable effort for both rule-based and machine learning systems.

An NER model capable of generalizing across novel domains is a problem worth addressing. Applications for NER are incredibly numerous and very valuable, especially in the context of Information Extraction. NER can be used, for example, to index and link off named entities, to attribute sentiments to companies or products, for question-answering systems, and of course in Information Extraction tasks. Additionally, NER is used more concretely by many web pages to tag various entities with links to bio or topic pages and by software packages like OpenCalais, Evri, AlchemyAPI, Yahoos Term Extraction, and by companies like Apple, Google, and Microsoft for smart applications related to document content.

The approach that I would like to take could potentially produce a feature set that can be used efficiently and easily across many different domains. Furthermore, it will provide more research into the effectiveness of deep learning paradigms in Natural Language Processing; an area where deep learning should intuitively be very successful in.

1.4 System Architecture and Approach

My system architecture will consist of a preprocessing module, a DBN module, and a CRF module wherein the DBN's intermediate layers will be used as features. My approach will consist of 8 steps:

1. The first step will be receiving, aggregating and scrubbing (if the data is messy or unstructured) any extra data the instructor can provide me with. Upon reception of the data, this will only take a few hours to complete.
2. Then, the next step is converting the words into vector representations either by training a word2vec model myself or by lookup in a massive pre-trained word2vec model. This will take around anywhere from an hour to a day depending on the training size of the data I have access to.
3. Then, I will set up an easily tunable Deep Belief Network. This will take a few hours.
4. After set up, I will begin tuning the performance of the DBN as a classifier for NER. This will take anywhere from a few days to a week.
5. Next, I will "remove" the output layer of the DBN and extract intermediate layers to be used as features. This will only take a few hours.
6. After storing these features to stable storage, I will build and initialize a CRF model capable of easily incorporating the DBN features into its feature set. This will take around a day or so.
7. Then, I will test the CRF with the DBN feature set against other instances of CRFs with different feature sets and compare and analyze performance. I will also record and deliver the CRF with the feature set that performs best. This will take around a week if I am to test with a decent amount of models.
8. Finally, I will perform a comprehensive analysis of the model and results to be included in the project presentation.

1.5 Deliverables

Upon completion of this project I hope to deliver a small, powerful set of features that can be used across multiple domains for Named Entity Recognition. Additionally, I will provide a framework for generating higher-level features related to the classification of named entities from word2vec representations of particular words through the use of a truncated Deep Belief Network.

1.6 Team Organization

Having had experience with machine learning along with deep learning, I feel qualified to take on this project. I have adequate knowledge of statistics and probability and have taken Machine and Computational Learning at RPI. During the course of my research, I have implemented neural networks across many different domains. Additionally, I have experience implementing and tuning Deep Belief Networks though not in the context of Natural Language Processing.

1.7 Required Resources

Ideally, there are some resources I would like to receive from the instructor. Namely, I hope to receive a bigger set of data annotated for Named Entity Recognition to use in training and testing. Additionally, if possible, I would like to receive NER annotated data from various other different domains as I would like to compare performance across the available domains to analyze the generalization of the model.

1.8 Time Table

With the following key assumptions in mind:

1. I will be able to successfully tune the hyperparameters of the DBN for classification, and
2. both the DBN and CRF will be able to train in a tractable amount of time given the data I have access to,

my proposed time table is outlined in the table below.

Table 1: Deliverables Time Table.

Step	Approximate Required Time
1. Scrubbing/Preprocessing data	4 hours
2. Word vector conversion	1 day
3. DBN model set up	4 hours
4. DBN training and testing	1 week
5. DBN feature extraction	4 hours
6. CRF model set up	4 hours
7. CRF training and testing	1 week
8. Formal analysis	1 day