Self-Adjustable BootStrapping for Web-Scale Named Entity Extraction using N-grams

Sushant Narsale
Johns Hopkins University
sushant@jhu.edu

Abstract

Named Entity Extraction refers to task of identifying and extracting mentions of names like person names, locations, time expressions, monetary values etc from text. There have different approaches to Named Entity extraction and classification based on supervised and semi-supervised learning. This paper describes a bootstrapping approach to extracting Named Entities for 150 categories from Wikipedia. The algorithm uses N-gram data generated from Wikipedia and builds a self-adjustable model, which adapts parameters according to NE category under consideration.

1 Introduction

Information retrieval applications are increasingly benefited from a list of Named Entities (Cucerzan and Yarowsky, 1999) (Pantel et al, 2009). Creating lists of Named Entities on Web is critical for query analysis, document categorization and ad matching. (Pantel et al, 2009). There have been various shared tasks like CoNLL 2002 and CoNLL 2003, with the aim of improving Named-Entity taggers. Use of CRF’s and other supervised methods which treat NER as a sequence labeling task have shown to achieve results which are very close to human performance in terms of accuracy. However the usefulness of supervised methods are restricted by availability of tagged(training) data. Training data is available in abundance for a few languages like English, Spanish, Dutch, however generating training data for new languages and domains is a costly and time consuming affair. This report describes a bootstrapping based approach to Named Entity Extraction, which works with a handful of seed examples for a category and large amount of untagged text to retrieve a bigger set of entities. Since seed examples and large amount of untagged corpus are easily available, this method can be adapted to any domain and language.

2 Background work and Resources

For each Named Entity category a set of 20 to 20,000 seed examples was made available for this project. The seed set was a subset of Extended Named Entity Sets (Sekine, Sudo, Nobata, 2002) developed at NYU and was made available for this project by Prof. Satoshi Sekine. For retrieving context and extracting new entities from wikipedia n-gram data, I used N-gram search engine built during the JHU CLSP summer workshop. The N-gram search engine handles upto 7-grams and indexes over 1.2 billion 7-grams.

3 BootStrapping

BootStrapping is an iterative process which requires minimal supervision in terms of seed entities. In bootstrapping one starts with a set of seed and retrieve contexts with which these seeds occur from unlabeled text and then uses these contexts to extract more entities similar to the seeds. Figure 1 shows

---

1 http://linserv1.cims.nyu.edu:23232/ngram_wikipedia2K/

2 Seeds in this context mean examples of a named entity category, for eg:- Computer Science and Mathematics are good seeds for academic fields category
an example of bootstrapping, here we start with a set of names of Presidents of countries and use these seeds words to retrieve contexts that generally occur with President names. The retrieved contexts are then ranked according to their frequencies and used to extract more President names, this process stops when no new entities are retrieved.

4 Problem Statement and Task definition

BootStrapping has previously been shown to be a robust and effective method for various NLP tasks (Yarowsky, 1995) (Vlachos and Gasperin, 2006). Similar methods have been shown to be successful for NE extraction in biological articles (Vlachos and Gasperin, 2006) and newspaper articles (Radev et al.). However our main challenge arises from the number of domains and categories of named entities. In this project I was working with 150 Named Entity categories, Figure 2 (Sekine, Sudo, Nobata, 2002) shows a hierarchical representation of the categories. As we show in table 3 different categories need different parameter settings, manually adjusting these parameters for each category is intractable and not scalable to a new category or language. The self-adjustable model described in the next section overcomes this problem by adjusting its parameters based on category under consideration.

5 Self-Adjustable Bootstrapping

As mentioned in the previous section our task was to develop a model which would adapt its parameters according to the NE category. For example Academic Fields category has a small number of strong contexts (Department of at) while category for Company Names has a large number of weak contexts (was bankrupted, hires), Award Names have strong suffix feature (Award/Prize), and Nationalities have a specific length (1) while Book Names show a wide variation in lengths. Hence the different parameters the model needs to adjust are:

1. Formula’s to score retrieved contexts and targets$^3$
2. Number of contexts from the total retrieved contexts to be considered for extracting new entities.
3. Weights for prefix/suffix features.
4. Weights for length features.

The 3rd and 4th features will be explained in later sections. To tune this parameters we divide our seed set into two parts of 70% 20% and 10% each. The 20% set is used as development set for tuning parameters and the smallest set is used as test set. A brute-force search is performed over all the values of parameters and each combination is evaluated on the development set. The combination of parameters with the best performance is then used. The next few sections explain scoring formulas, features generated and scoring function.

$^3$Targets here mean new entities that are extracted
5.1 Scoring formulas

When seed words are used to retrieve contexts for a given category, the extracted contexts are noisy since data on web is unstructured, therefore only a part of contexts are useful. Similarly not all the new entities (targets) retrieved are good. To weed good contexts/targets from bad ones we score the contexts/targets based on type and token statistics. Our model uses 5 formula’s each to score contexts and targets. These formulas are based on the following 3 statistics retrieved from wikipedia n-grams.

1. $F_i$ = Co-occurrence frequency of targets and the context
2. $F_t$ = Number of target types co-occurred with the context
3. $CF$ = Corpus frequency of the context/target

Table 1 gives a complete list of all the formulae used.

5.2 Features

Scoring formulas described in previous section are well proven techniques in information retrieval and are good indicators of quality of contexts and new entities extracted. However working with 150 categories we observed that there are peculiar features like prefix and suffix for seed entities, length of seed entities which vary across categories. This information is valuable and should be used to decide whether a newly extracted entity belongs to one of these categories, for eg. an entity ending with 'Award' has a higher probability of being an award compared to one which does not end in 'Award'. We incorporate this information about common prefixes/suffixes for a category in our model. The common prefixes/suffixes are obtained from the seed lists. Since we have a large seed list for most of the categories we consider any prefix/suffix that occurs with more than 5% of seeds as an indicative prefix/suffix for that category and use it as a feature while scoring new targets. The weight that each prefix/suffix carries in the score is determined by calculating the percentage of seeds in the original seed list that carry a particular prefix/suffix.

$$W_{prefix/suffix} = \frac{\text{Number of Seeds with a prefix/suffix}}{\text{Total number of seeds}}$$

Prefix and Suffix Features

Seed entities in most of the NE categories have a length of more than one word. Figure 2 shows a sample of seeds for 3 categories viz. Awards, Lake, and Bridge. Most of the entities in the Bridge category end in Bridge, while those in Awards have either prize, award or cup as their suffix. This information is valuable and should be used to decide whether a newly extracted entity belongs to one of these categories, for eg. an entity ending with 'Award' has a higher probability of being an award compared to one which does not end in 'Award'.

Length Features

All the 150 categories have seeds of variable length. However seeds in a particular category tend to have
similar lengths. For eg, 90% of the seeds in nationality category have length 1, while most of the seeds in books have length of more than 4. Like prefix/suffix features this information about length of seed entities is important and a strong indication of a new entity belonging to a category. Graph in figure 3 shows the trend in length of seed entities for 10 categories, similar trend is observed for all 150 categories. We generate statistics about lengths from our seed set and use length as a feature in scoring new targets. The feature is selected so as to bias towards length of maximum seeds in a category.

The final scoring function is a weighted interpolation of scoring formula, prefix/suffix feature score and length feature score.

\[
\text{score}_{\text{target}} = \alpha \cdot \text{formula} + \beta \cdot \text{score}_{\text{prefix/suffix}} + \gamma \cdot \text{score}_{\text{length}}
\]

### 5.3 Mean reciprocal rank and evaluations

To tune parameters of the model we use a development set, which consists of seeds obtained by sampling the original seed set. Each combination of parameters is scored based on the number of seeds from the dev set that are retrieved by the bootstrapping model. The intuition here is that parameters which extract seeds from development set higher in their list after scoring the targets should be given a higher score. Hence for evaluating a retrieved set of examples we use Mean Reciprocal Rank (MRR) (Radev et.al.). MRR was first introduced for evaluation of Question Answering systems with an idea of scoring a system based on position of expected correct answers. For every retrieved entity belonging to the development set, MRR sums the reciprocal of position of the entity in the retrieved list.

\[
MRR = \sum_{i=1}^{Q} \frac{1}{\text{rank}_{i}}
\]

where \( Q \) is entities in development set.

Figure 4 gives a diagrammatic representation of TRR.

### 5.4 Phrase Clustering

I used the Phrase Clusters built over google n-grams version 4 data, which were developed at the JHU CLSP Workshop 2009. Various heuristics were used to identify clusters from n-gram data and 1000 clusters were randomly initialized. Identified phrases were then clustered using K-means clustering. The intuition behind using clustering for Named Entity extraction was that phrases containing entities belonging to a particular category would be clustered together. Since the granularity in the hierarchy...
Table 4: Phrase cluster with maximum seed words for a category

<table>
<thead>
<tr>
<th>Category</th>
<th>Cluster ID</th>
<th>Total Matches</th>
<th>Matching %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>287</td>
<td>237</td>
<td>92</td>
</tr>
<tr>
<td>Incidents</td>
<td>548</td>
<td>23</td>
<td>91</td>
</tr>
<tr>
<td>Occasions</td>
<td>474</td>
<td>24</td>
<td>83</td>
</tr>
<tr>
<td>Facilities</td>
<td>769</td>
<td>72</td>
<td>70</td>
</tr>
<tr>
<td>Titles</td>
<td>545</td>
<td>11566</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 5: Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic Fields</td>
<td>71</td>
<td>61</td>
<td>66 (+11)</td>
</tr>
<tr>
<td>Baseline</td>
<td>61</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>Airports</td>
<td>23</td>
<td>61</td>
<td>33 (+0)</td>
</tr>
<tr>
<td>Baseline</td>
<td>23</td>
<td>60</td>
<td>33</td>
</tr>
<tr>
<td>Bridges</td>
<td>16</td>
<td>47</td>
<td>24 (+10)</td>
</tr>
<tr>
<td>Baseline</td>
<td>9</td>
<td>29</td>
<td>14</td>
</tr>
</tbody>
</table>

of Named entity categories and the phrase clusters were no the same we expected some overlap in categories.

Table 4 shows for 10 NE categories the clusters that contain the maximum number of entities from the original seed set for the category. As can be seen from the table a majority of seeds for a category are clustered together. This knowledge about clusters could be used to extract new entities.

6 Results

This section describes the results on 3 categories and compares them with the baseline results. The baseline results were calculated using the parameter settings that worked best on an average for all 150 categories. A test set sampled from the original seeds list was used to calculate recall.

\[
\text{Recall} = \frac{\text{test seeds retrieved}}{\text{total seeds in test set}}
\]

To calculate precision, we sampled 100 examples from the newly retrieved seed set.

\[
\text{Precision}_{100\text{samples}} = \frac{\text{correct entities in retrieved set}}{\text{Total entities retrieved}}
\]

Table 5 shows the precision, recall and f-measure values for 3 categories compared to the baseline performance. As can be seen from this table using adjustable bootstrapping gives a significant improvement over the baseline results for two categories. There is no improvement for Airports and on further investigating we observed that the baseline parameter setting and the best settings used for Airport gave similar MRR scores and hence their is not much improvement of one over the other.

7 Future Work

The work on this project was complemeted in six weeks at the JHU CLSP workshop 2009. The results have been encouraging and this leaves a lot of scope for future work. One path to follow could be represent the new targets (extracted entities) by vectors and find similarity between them. This would give a compact representation and there would be no need for tuning different scoring formulas. A vector representation will also allow us to use unsupervised methods to learn weights for different features. A vector representation will be flexible to allow inclusion of more information relevant information for example using lexical resources like wordnet, POS tags, cluster id’s for phrase cluster’s defined previously.

Acknowledgments

I would like to thank Dekang Lin and Prof. David Yarowsky for giving me the opportunity to work at JHU CLSP workshop 2009. I would also like to thank Prof. Satoshi Sekine for giving me an opportunity to work on this project. It was his guidance and support that made this project a success. Lastly, I would like to thank Kenneth Church for his constant guidance and suggestion during the course of this project.

References


