Semi-supervised Relation Extraction using EM Algorithm

Sachin Pawar[1,2], **Pushpak Bhattacharyya**[1]
[1]Computer Science and Engineering
Indian Institute of Technology, Bombay
{sachinpawar,pb}@cse.iitb.ac.in

Girish Keshav Palshikar[2]
[2]Systems Research Lab
Tata Consultancy Services Ltd., Pune
gk.palshikar@tcs.com

Abstract

Relation Extraction is the task of identifying relation between entities in a natural language sentence. We propose a semi-supervised approach for relation extraction based on EM algorithm, which uses few relation labeled seed examples and a large number of unlabeled examples (but labeled with entities). We present analysis of how unlabeled data helps in improving the overall accuracy compared to the baseline system using only labeled data. This work therefore shows the efficacy of a sound theoretical framework exploiting an easily obtainable resource named "unlabeled data" for the problem of relation extraction.

1 Introduction

Relation Extraction is an important task in Information Extraction, which is an extension to the task of Named Entity Extraction. The task of named entity extraction deals with extracting named entities of interest from a set of documents, whereas relation extraction goes a step further and tries to extract entities along with the relations between them. For example, in the sentence "`David Cowan, Internet Startup founder and acting chief executive , is a general partner of Bessemer.`", along with named entities `David Cowan` (PERSON) and `Internet Startup` (ORGANIZATION), relation extraction system is expected to identify that these two entities are related and recognize the relation type as "*Role*". Table 1 shows various types of relations that we consider in this paper.

Type	Description
Role	Indicates the role a person plays at an organization. e.g. member, founder, citizen-of etc.
At	Represents location relationships like based-in, residence or located-at.
Part	Indicates part-whole relationships like part-of, subsidiary etc.
Near	Identifies relative locations.
Social	Represents various social or professional relationships between two persons like mother, sister, spouse, secretary etc.
Affects	This is specific to the agriculture domain and represents the relationship between DISEASE and CROP named entities
Null	We consider this additional relation which indicates that the two entities are not related.

Table 1: Various types of relations considered in this paper. Top 5 relations were defined as part of the ACE program (Doddington et al., 2005)

A lot of supervised approaches addressing the Relation Extraction problem have been studied. These approaches can be roughly classified into two types : features-based (Zhou, 2005; Jiang, 2007) and kernel-based (Zelenko, 2003; Mooney, 2005). These approaches analyze lexical, syntactic and semantic features (like in table 2) with respect to the labeled relation. Their performance is dependent on the size of the labeled data and the extent to which labeled data covers evidence about various features considered. It is time consuming and effort-intensive to create such labeled data for relation extraction. Data with no relation labeling but named entity labeling is relatively easy to get with the help of advanced named entity extraction and recognition systems available today. This motivated us to focus on addressing relation extraction problem using only entity labeled data and a few seed instances marked with relation information.

The major contribution of this paper is the principled formulation of the relation extraction problem in the framework of the Expectation-Maximization algorithm. The rest of the paper is organized as follows - section 2 covers some related work, then section 3 provides exact problem definition and describes our approach. Experiments and results are discussed in section 4 and conclusions and future work are discussed in section 5.

2 Related Work

There have been many approaches (Brin, 1999; Agichtein and Gravano, 2000) which extract relations from text using a bootstrapping method. These approaches just need a few seed examples from the user and the initial extraction patterns are learned based on the seeds. Then they proceed iteratively by making new extractions and using these new extractions, patterns are updated. For these approaches, one entity pair can have one and only one relation label across all mentions of that pair in the data. In general, the same entity pair can have different relation labels depending on how they are mentioned in sentences. For example, consider the entities `John` and `India` in the following sentences,

1. `John, who is a citizen of India, arrived in the United States.` - Here entities `John` and `India` have relation label *Role*.
2. `John lives in India.` - Here entities `John` and `India` have relation label *At*.
3. `John likes India's culture.` - Here entities `John` and `India` have no relation at all.

Our problem definition of relation extraction (which is described in detail in the next section) allows the same entity pair to have different relation labels depending on how they are mentioned in the sentences. Moreover, some approaches like Brin's (1999) DIPRE deal with documents on the web, i.e. they also use HTML structure/tags in their patterns, whereas we only consider "natural language" sentences.

Hasegawa et al. (2004) proposed an approach for unsupervised relation extraction which requires only named entity labeled data. Their approach is based on creating clusters of named entity pairs by calculating cosine similarity between their *context* vectors. These context vectors are mainly based on lexical information. Higher level syntactic and semantic information is not used, which would have been quite useful for relation extraction. Recently, Sun et al. (2011) proposed a semi-supervised approach to address the problem of sparsity of lexical features in supervised relation extraction. They used word clusters as features to deal with this problem and obtained 70-80% F-measure for major relation types in ACE 2004 (Doddington et al., 2005) dataset[1].

In the area of text classification, learning using both labeled and unlabeled data is discussed in detail by Nigam et al. (2000). They use the EM algorithm for this semi-supervised learning. Initially, a Naive Bayes classifier is trained using only labeled data and then unlabeled data is classified using that classifier. Then instances in the unlabeled data are also added to the training data with a label and an instance weight proportional to the classifier confidence for that instance for that particular label. Then a new classifier is trained using labeled as well as unlabeled data and this process is repeated till the weights are converged. Our approach is mainly motivated from this work, but the major difference is that they use *generative* Naive Bayes classifier whereas we use *discriminative* Maximum Entropy (MaxEnt) classifiers. Even with the independence assumptions, Naive Bayes performs well for text classification. But for relation extraction, discriminative classifier like Maximum Entropy classifier works well as it combines various overlapping and correlated features in a better way without requiring any independence assumptions.

3 Relation Extraction

3.1 Problem Definition

The task of binary relation extraction consists of - i) identifying the named entities in the sentence and ii) for each pair of entities, identifying which relation exists between them. This relation can be the *null* relation, indicating that the entities are not related. Here, we are focusing only on the second part, i.e. we assume that the named entities in the sentence are already identified and our task is to identify the relations among these entities. Also, we are focusing only on binary relations, hence an

[1] We could not test our system on this data, as it is not freely available

Feature Type	Description
Lexical	Words within 2 entities
	Words occurring between 2 entities
Syntactic	Head words & dependency relations of 2 entities in the dependency tree
	Path of dependency relations connecting 2 entities
	Length of dependency path
	Common ancestor of 2 entities in dependency tree
	Path of chunk tags connecting 2 entities
Semantic	Whether any of the context words have one of these as an ancestor in the WordNet hypernymy tree - *person*, *organization*, *location* or *relation*.

Table 2: Description of some of the features used

instance to be considered for relation extraction is represented as : $(E_1, E_2, \vec{F})$. Where,

- E_1 : Type of the first named entity in the sentence
- E_2 : Type of the second named entity in the sentence
- $\vec{F}$: Feature vector characterizing the sentence, the entities and their positions in the sentence. Most of our features (table 2) are inspired from the work of Zhou et al. (2005) and Jiang et al. (2007).

Given such an instance $(E_1, E_2, \vec{F})$, our aim is to identify the relation between two named entities.

3.2 Our approach using EM Algorithm

We propose to address this relation extraction problem by using the Expectation-Maximization algorithm. The pair of named entity types $(E_1 E_2)$ and sentence features $(\vec{F})$ are modeled as observed variables, whereas relations are modeled as hidden variables.

3.2.1 Representation

Let the number of instances in the dataset D be N and i^{th} instance be represented as $(\vec{x_i}, \vec{F_i}, \vec{z_i})$, where-

- **Observed variable,** $\boldsymbol{\vec{x_i}} = \langle x_{i1}, x_{i2} \cdots x_{iK} \rangle$, where K is the number of distinct pairs of entity types and $x_{ik} = 1$ if k^{th} pair of the entity types is present in the i^{th} instance and 0 otherwise.
- **Observed variable,** $\boldsymbol{\vec{F_i}} = \langle F_{i1}, F_{i2} \cdots F_{iL} \rangle$,where L is the number of distinct binary features characterizing the sentence and the entities in it and $F_{il} = 1$ if the l^{th} feature is present in the i^{th} instance and 0 otherwise.
- **Hidden variable,** $\boldsymbol{\vec{z_i}} = \langle z_{i1}, z_{i2} \cdots z_{iM} \rangle$, where M is the number of possible relations including a *null* relation and $z_{ij} = 1$ if the j^{th} relation is present in the i^{th} instance and 0 otherwise.

3.2.2 Algorithm

Assuming that the instances in D are independently and identically distributed with the underlying parameters Θ, the data likelihood can be expressed as,

$$L(D;\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[Pr(\vec{x_i}, \vec{F_i}, z_{ij} = 1) \right]^{z_{ij}} \quad (1)$$

$$L(D;\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[Pr(\vec{F_i}) Pr(z_{ij} = 1|\vec{F_i}) \prod_{k=1}^{K} Pr(x_{ik} = 1|z_{ij} = 1, \vec{F_i})^{x_{ik}} \right]^{z_{ij}} \quad (2)$$

$Pr(z_{ij} = 1|\vec{F_i})$=$Pr(j^{th} relation|\vec{F_i})$ can be modeled by a log-linear model using feature functions based on $\vec{F_i}$. Similarly, $Pr(x_{ik} = 1|z_{ij} = 1, \vec{F_i})$=$Pr(k^{th} E_1E_2\_pair|j^{th} relation, \vec{F_i})$ can be modeled by M log-linear models(one for each relation) using the same feature functions. The feature functions (for the i^{th} instance) are defined as *the combinations of the features and the class labels* as follows:

$\boldsymbol{f_{jl}(i,c)} = F_{il}$, if $j = c$ &
$\boldsymbol{f_{jl}(i,c)} = 0$, if $j \neq c$

Using above definition for the feature functions, $Pr(Rel = j|\vec{F_i}) =$

$$= \frac{\exp(\sum_{j'=1}^{M} \sum_{l=1}^{L} \lambda_{j'l} f_{j'l}(i,j))}{\sum_{j''=1}^{M} \exp(\sum_{j'=1}^{M} \sum_{l=1}^{L} \lambda_{j'l} f_{j'l}(i,j''))} \quad (3)$$

and $Pr(x_{ik} = 1|z_{ij} = 1, \vec{F_i}) =$

$$= \frac{\exp(\sum_{k'=1}^{K} \sum_{l=1}^{L} \alpha_{jk'l} f_{k'l}(i,k))}{\sum_{k''=1}^{K} \exp(\sum_{k'=1}^{K} \sum_{l=1}^{L} \alpha_{jk'l} f_{k'l}(i,k''))} \quad (4)$$

Therefore, the final expression for the data log-likelihood is: $LL(D;\Theta) =$

$$= \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}\log(Pr(\vec{F_i})) + \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}\left(\sum_{j'=1}^{M}\right.$$
$$\sum_{l=1}^{L}\lambda_{j'l}f_{j'l}(i,j) - \log\sum_{j''=1}^{M}\exp(\sum_{j'=1}^{M}\sum_{l=1}^{L}$$
$$\left.\lambda_{j'l}f_{j'l}(i,j''))\right) + \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}\left(\sum_{k=1}^{K} x_{ik}\right. \quad (5)$$
$$\left(\sum_{k'=1}^{K}\sum_{l=1}^{L}\alpha_{jk'l}f_{k'l}(i,k)\right.$$
$$\left.\left. - \log\sum_{k''=1}^{K}\exp(\sum_{k'=1}^{K}\sum_{l=1}^{L}\alpha_{jk'l}f_{k'l}(i,k''))\right)\right)$$

Values of the hidden variables ($\vec{z_i}$'s) and the parameters ($\Theta : \lambda's$ and $\alpha's$) can be estimated by the EM algorithm. The parameter values are initialized in some way and the following EM steps are repeated till the log-likelihood is converged.

- **E-step:**

$$E(z_{ij}) = Pr(z_{ij} = 1|\vec{x_i}, \vec{F_i}) \quad (6)$$

$$E(z_{ij}) = \frac{Pr(z_{ij} = 1, \vec{x_i}, \vec{F_i})}{Pr(\vec{x_i}, \vec{F_i})} \quad (7)$$

$$E(z_{ij}) = \frac{Pr(z_{ij} = 1, \vec{x_i}, \vec{F_i})}{\sum_{j'=1}^{M} Pr(z_{ij'} = 1, \vec{x_i}, \vec{F_i})}, \forall_{i,j} \quad (8)$$

  $E(z_{ij})$ can be calculated using the equations 2, 3 and 4 and using the current values of the parameters λ's and α's.

- **M-Step:**
  In this step, the data log-likelihood $LL(D;\Theta)$ is maximized using the current values of the hidden variables $\vec{z_i}$'s. Maximizing $LL(D;\Theta)$ is equivalent to learning a number of *instance-weighted* MaxEnt classifiers .
  One MaxEnt classifier is learnt for estimating $Pr(j^{th}relation|\vec{F_i})$. This is trained using $\vec{F_i}$'s as the features and all the relations as class labels. Each instance is weighted by corresponding $E(z_{ij})$.
  M **MaxEnt classifiers** are learnt for estimating $Pr(k^{th}E_1E_2\_pair|j^{th}relation, \vec{F_i})$, one for each relation. Classifier for the j^{th} relation is trained using $\vec{F_i}$'s as the features and the pair of entity types (E_1E_2) for the i^{th} observation as class label. Each instance is weighted by the corresponding $E(z_{ij})$.

3.2.3 Initial values of the parameters

EM algorithm does not guarantee convergence to a global optimum. Hence, before the EM iterations are started, it is necessary to initialize the parameter values in some intelligent way. When labeled data is available, one of the most effective ways to initialize the parameters is to learn them using only the labeled data available. With the introduction of the labeled data, only change in the above mentioned EM procedure is in the M-Step. Now, the combined log-likelihood of the labeled and unlabeled data i.e. $LL(D_L;\Theta) + LL(D_U;\Theta)$ is maximized. Here, D_U represents unlabeled data and D_L represents small amount of labeled data.

3.2.4 Weighing down unlabeled data

One of the important observations made by Nigam et al. (2000) is - when the unlabeled data is huge compared to the labeled data(which is also true in our case), parameters learned by the EM algorithm in the M-step are almost completely dominated by the unlabeled data. And this leads to degradation in the accuracy as compared to learning only from the labeled data. To overcome this problem, Nigam et al. (2000) proposes to weigh down the contribution of the unlabeled data in the log-likelihood. Therefore, in the M-step, instead of $LL(D_L;\Theta) + LL(D_U;\Theta)$, we maximize $LL(D_L;\Theta) + \beta LL(D_U;\Theta)$, where $0 < \beta < 1$. The value of β is generally correlated with the ratio of number of labeled instances to the number of unlabeled instances. The best value of β is determined experimentally.

4 Experiments and Results

We tested our relation extraction approach on two corpora belonging to completely different domains. In this section, we describe how the datasets were created and analyze the results obtained.

4.1 CoNLL Shared Task 2003 Corpus

This corpus[2] is a collection of various news articles, mostly belonging to the categories - politics and sports. It is tagged with the named entity tags like PER, ORG and LOC, but there are no gold-standard relations labels.

4.1.1 Data set creation

We are interested in binary relations. Hence, if any sentence has n entities, we convert it to $\binom{n}{2}$

[2] http://www.cnts.ua.ac.be/conll2003/ner/

instances. Consider the sentence:
"I think there's probably a whole lot of material that would expand our understanding of the sociology of the war", said [Edward Smith]$_{PER}$, director of American Studies at [American University]$_{ORG}$ in [Washington]$_{LOC}$.
It has 3 named entities : Edward Smith (PER), American University (ORG) and Washington (LOC). Hence, the following 3 instances are created,

1. (Edward Smith, American University) : E_1E_2 pair - PER_ORG, along with the features created as per table 2.
2. (American University, Washington) : E_1E_2 pair - ORG_LOC, along with the features.
3. (Edward Smith, Washington) : E_1E_2 pair - PER_LOC, along with the features.

We consider 10000 such instances in CoNLL 2003 dataset. We are interested in the following high-level relations defined as part of the ACE program (Doddington et al., 2005) : - *Role, At, Part, Near, Social* and *Null*. Table 1 describes these relations. We labeled 130 instances manually with the appropriate relations to create a small labeled set.

4.1.2 Results

We start the EM iterations by learning the initial parameters on the labeled set. After the EM algorithm converges in 10 iterations, we check the values of $E(z_{ij})$ for the unlabeled instances. $E(z_{ij})$ is nothing but the probability of the j^{th} relation label in the i^{th} instance. For each relation, we consider top $\mathcal{K}$ instances according to $E(z_{ij})$ value for that particular relation. The accuracy is computed by manually verifying the relation labels for these $\mathcal{K}$ instances. For example, we check how many of the top 500 instances for the relation *Role* are actually having the relation label *Role*. Table 3 shows these results for all the relations.

4.1.3 Labeled Data Baseline

The MaxEnt classifiers using exactly the same features (table 2) but trained using only the labeled data are used as a baseline. In other words, the values of $E(z_{ij})$(for unlabeled instances) computed in the E-step of our EM algorithm's first iteration, can be considered as a baseline output. Because in the first iteration in our E-step, we use those MaxEnt classifiers whose parameters are learned using only the labeled data. Hence, as the output of baseline system, for each relation, we consider top $\mathcal{K}$ instances according to $E(z_{ij})$ value (after first iteration) for that particular relation. Table 3 compares the baseline system and our technique based on EM using both labeled and unlabeled data.

Relation	$\mathcal{K}$	Labeled Data Baseline	Co-Training Baseline	EM using unlabeled Data
Role	500	422 (84.4%)	296 (59.2%)	442 (88.4%)
At	100	56 (56.0%)	36 (36.0%)	65 (65.0%)
Social	50	28 (56.0%)	27 (54.0%)	31 (62.0%)
Part	20	13 (65.0%)	16 (80.0%)	14 (70.0%)
Near	20	3 (15.0%)	3 (15.0%)	7 (35.0%)

Table 3: Precision within top $\mathcal{K}$ instances for each relation. Each cell shows no. of instances (out of $\mathcal{K}$) for which the correct relation was identified along with the precision shown in brackets. The weight of the unlabeled data used in the EM method is $\beta = 0.15$

4.1.4 Co-Training Baseline

Co-Training (Blum and Mitchell, 1998) is one of the popular methods of combining labeled and unlabeled data. We implemented one more baseline system based on the Co-Training framework. This framework requires the features to be partitioned into two independent sets or views. We partitioned our features as follows:

1. Lexical features as well as features capturing the information about part of speech tags
2. Syntactic features capturing the dependency parsing information as well as semantic features

Two different MaxEnt classifiers (C_1 and C_2) are then trained using only the labeled data, but each one of them uses only one of these feature views. The unlabeled instances are then classified using both C_1 and C_2. Then for each of these classifiers, few instances with highest confidence are chosen for each class label (relation in this case). These instances are then added to the labeled data with

Feature Functions		Labeled data baseline	EM with unlabeled data	
Feature	**Class**		**Iter 2**	**Iter 10**
Word "director" is on dep. path	Role	Absent	0.078	0.112
	Null	Absent	-0.035	-0.042
At least 1 word on dep path has WordNet category "person"	Role	-0.068	0.037	0.318
	Null	0.023	-0.125	-0.253
Dep. relation of 2^{nd} entity = prep_at	Role	Absent	-0.022	0.13
	Null	Absent	0.019	-0.022
Chunk tags path = NP-O-NP-PP-NP-PP-NP	Role	Absent	0.033	0.051
	Null	Absent	0.024	-0.027
Dep. path from 2^{nd} entity to common ancestor=pobj<-prep <-pobj<-prep<-appos	Role	Absent	0.029	0.042
	Null	Absent	0.024	-0.016

Table 4: Weights for the feature functions of MaxEnt classifiers (predicting the probability of relation label given features)

their predicted class labels. Now, both C_1 and C_2 are trained on the new labeled data and the process is iterated. For each class label, the number of instances which are added to the labeled data, varies in accordance with their empirical frequency. For example, in our case the most frequent relation is *Null* followed by *Role*, *At*, *Social*, *Part* and *Near* in that order. Hence, in each iteration, each classifier adds 10 new instances of *Null*, 2 new instances of *Role* and 1 new instance each for the other relations.

The accuracy is computed by manually verifying the relation labels for the top $\mathcal{K}$ instances for each relation. Table 3 shows the results for Co-Training baseline. Our method based on the EM algorithm clearly performs better than the Co-Training method for most of the relations. It was observed that the Co-Training accuracy was quite high in initial few iterations, but it deteriorated fast when some instances with incorrect labels were added to the labeled data.

4.1.5 Analysis of the results

The actual recall is difficult to determine as *true* relations for the unlabeled instances are unknown, hence we compare the precision with *same coverage* for all the 3 cases - i) Baseline using only labeled data, ii) Co-Training baseline using both labeled and unlabeled data and iii) EM using both labeled and unlabeled data. Another important point to note is that some relations like *Part* and *Near* are quite infrequent in the dataset, hence as the EM iterations progress, their corresponding z values diminish fast. Hence, number of extracted instances for such relations is comparatively small.

As an example of efficacy of our approach, consider the first instance `(Edward Smith, American University)` from the example sentence in the previous subsection. The labeled data doesn't cover many of the features for this instance, hence probability for the relation *Role* by the baseline system using only labeled data is only $\mathbf{0.23}(E(z_{ij})$ for *Role* after the first iteration) , though *Role* relation exists there. But using the unlabeled data along with the labeled data adds knowledge about such features through clustering using the EM algorithm and rightly increases $E(z_{ij})$ for *Role* to $\mathbf{0.94}$ at convergence. Table 4 lists some of the *informative* features for relation *Role* along with their weights learnt by the MaxEnt classifier in the baseline system as well as by the MaxEnt classifiers in various EM iterations. Note that the weights are shown for a combination of feature and class label(relation in this case), as feature function $(f(x, c))$ for a MaxEnt classifier is in fact of the form - The feature f is present in the instance x AND the class label is c. The weights for the feature functions involving *Role* are increasing whereas the weights for those involving *Null* are decreasing as iterations progress.

Table 4 depicts how our algorithm overcomes the problem of sparsity of features. Unlike Sun et al.'s (2011) approach which proposes a way to handle sparse lexical features, our approach provides a systematic way in which all kinds of sparse features are handled. Like lexical features, other features such as dependency paths, chunk tag paths are also quite sparse and the issue of sparsity has to be addressed for them too. Because these features are also quite informative for relation extraction.

The lexical feature "*Word "director" is on dep. path*" is absent in the labeled data. This feature is a good indicator for *Role* relationship between PERSON and ORGANIZATION. There are many informative features like this one, which are absent in the labeled data. This is because we have only a few seed labeled instances and it is obvious that these labeled instances will cover only a very small set of features. Positive effect of our approach is quite pronounced in this case because even though this feature is absent in the labeled data, over 10 EM iterations it gets a desired positive weight.

Also, consider the semantic feature "*At least 1*

word on dep. path has WordNet category "person"". Although this feature is seen in the labeled data and is a good indicator of relation *Role*, it has got a negative weight when only labeled data is used. This is because of very limited number of training records. But as we can see in the table 4, the weight of this feature increases as EM iterations progress. As an effect of these changes in the feature weights, the instance `(Edward Smith, American University)` gets correctly classified as an instance of relation *Role*.

4.2 Agriculture News Corpus

We wanted to test our relation extraction approach on some other domain and we chose "agriculture" domain as it is completely different from CoNLL corpus (with different entity types and relations) and socially more relevant. Also, unlike the CoNLL corpus, this agriculture domain corpus is not labeled with entity types and we had to use an unsupervised named entity extraction algorithm to get these labels. Therefore, using this corpus enables us to test our approach in the presence of "noisy" entity type labels (we will see an example later), unlike the earlier corpus where we had gold-standard entity labels.

4.2.1 Data set creation

We used the same corpus which was used by Patil et al. (2013). This corpus was obtained by crawling FarmPress group's agriculture news web sites. It contains 30533 news articles containing 999168 sentences. We focused on the following two entity types:

- **CROP** : Names of the crops including crop varieties
- **DISEASE** : Names of crop diseases and disease causing agents such as insects, bacteria, viruses, pests, fungi etc.

We created gazettes for these two entity types by using an unsupervised gazette creation (named entity extraction) algorithm described in Patil et al. (2013). The gazettes were verified manually and incorrect entries were removed. Finally, we obtained a CROP gazette of size 346 and a DISEASE gazette of size 370. Then we labeled the entire corpus with entity labels - CROP and DISEASE, by just looking up in the gazettes created. We obtained 12762 sentences[3] containing at least one phrase of the type DISEASE and at least one phrase of the type CROP. As we are interested in binary relations, we created instances from these sentences in the similar way we did for the CoNLL corpus. But, here we created instances only with entity type pairs CROP_DISEASE and DISEASE_CROP. Other possibles pairs CROP_CROP and DISEASE_DISEASE were not created because we were not interested in extracting any relation among them. Consider the sentence :
$[\texttt{Peanuts}]_{CROP}$ `grown on land where` $[\texttt{soybeans}]_{CROP}$ `have been are especially susceptible to` $[\texttt{white mold}]_{DISEASE}$, $[\texttt{limb rot}]_{DISEASE}$ `and` $[\texttt{CBR}]_{DISEASE}$.
Here, the following 6 instances are created. All of them are having E_1E_2 pair as CROP_DISEASE and features created as per table 2.

1. `(Peanuts, white mold)`
2. `(Peanuts, limb rot)`
3. `(Peanuts, CBR)`
4. `(soybeans, white mold)`
5. `(soybeans, limb rot)`
6. `(soybeans, CBR)`

19500 such instances are created from 12762 sentences. Here, we are interested in only two types of relations:

- *Affects* : Indicating that in the given instance, DISEASE entity actually affects the CROP entity. For example, the entities in the first three instances in the above example, indicate this relation.
- *Null* : Indicating that the DISEASE entity in the given instance does not affect (or have no relation with) the CROP entity. For example, the entities in the last three instances in the above example have no relation.

Only 24 out of the 19500 instances were randomly selected and manually labeled with the correct relation label - *Affects* or *Null*. These seed examples constitute our labeled data.

[3]This data set can be made available on request

Relation	$\mathcal{K}$	Labeled Data Baseline	Co-Training Baseline	EM using unlabeled Data
Affects	200	160 (80.0%)	148 (74.0%)	176 (88.0%)

Table 5: Precision within top $\mathcal{K}$ instances for each relation. Each cell shows no. of instances (out of $\mathcal{K}$) for which correct relation was identified along with the precision shown in brackets. Weight of unlabeled data used in EM method is $\beta = 0.05$

4.2.2 Noisy labels

As we mentioned earlier, the entity type labels assigned can be "noisy". For example, consider the following labeled sentence:
`Many acres of [tobacco]`$_{CROP}$ `were being hit hard with [tomato]`$_{CROP}$ `[spotted wilt virus]`$_{DISEASE}$`.`
In this sentence, the actual name of the virus is `tomato spotted wilt virus`, but this phrase is missing from our gazette of type DISEASE. Hence, we get incorrect labeling with `tomato` getting label CROP and `spotted wilt virus` getting label DISEASE, whereas the combined phrase should have got the label DISEASE. Because of such noisy labels some of the features (like previous word of the phrase) computed can be misleading.

4.2.3 Results

For extracting relations using our approach, we proceed in exactly the same way as described for the CoNLL corpus. We start the EM iterations by learning the initial parameters on the labeled set. After 15 iterations, we check the values of $E(z_{ij})$ for the unlabeled instances. As $E(z_{ij})$ is nothing but the probability of the j^{th} relation label in the i^{th} instance, we consider the top $\mathcal{K}$ instances according to the $E(z_{ij})$ value for each relation. Accuracy is computed by manually verifying the relation labels for these $\mathcal{K}$ instances. For example, we check how many of the top 200 instances for the relation *Affects* are actually having the relation label *Affects*. Table 5 shows the final results.

4.2.4 Labeled Data Baseline

Here we consider the same "labeled data baseline" that we considered in case of the CoNLL corpus, i.e. we consider the $E(z_{ij})$ values after the first iteration for getting the top $\mathcal{K}$ instances. As we can see in the table 5, considering the additional unlabeled data through the EM algorithm for learning the parameters of the MaxEnt classifiers, improves the accuracy from 80% to 88%.

Feature Functions		Labeled data baseline	EM with unlabeled data	
Feature	**Class**		**Iter 2**	**Iter 15**
Word "mold" is on dep. path	Affects	Absent	-0.004	0.005
	Null	Absent	0.004	-0.005
Common ancestor in dep. tree is "susceptible"	Affects	Absent	-0.017	0.002
	Null	Absent	0.017	-0.002
Dep. path from 2^{nd} entity to common ancestor=conj<-pobj<-prep	Affects	Absent	0.02	0.015
	Null	Absent	-0.02	-0.015

Table 6: Weights for feature functions of MaxEnt classifiers (predicting probability of relation label given features)

4.2.5 Co-Training Baseline

We also consider another baseline based on the Co-Training (Blum and Mitchell, 1998) framework. The basic algorithm and feature partitions are exactly the same as in case of the CoNLL corpus. In our agriculture news corpus, the relations *Affects* and *Null* have almost the same empirical frequency. Hence, in each iteration of Co-Training, 2 new instances are added to the labeled data for both of these relations. Table 5 shows the results for this Co-Training baseline.

4.2.6 Analysis of the results

Consider the instance `(Peanuts, limb rot)` from the example sentence mentioned previously. Probability for the relation *Affects* by baseline system using only labeled data is only $\mathbf{0.47}(E(z_{ij})$ for *Affects* after first iteration) , though *Affects* is the true relation label. But using unlabeled data along with the labeled data adds knowledge about such features through clustering using the EM algorithm and rightly increases $E(z_{ij})$ for *Affects* to $\mathbf{0.7}$ after 15 iterations. Table 6 lists some of the *informative* features for relation *Affects* which were absent in the labeled data, but the knowledge about them is added by EM using the unlabeled data. It also shows how their feature weights are pushed in the right direction, as EM iterations progress.

The features shown in table 6 like "*Word "mold" is on dep. path*" and "*Common ancestor of both entities in dep. tree is "susceptible"*" are absent in the labeled data, as we have only a few seed labeled instances. Positive effect of

our approach is quite pronounced in this case because even though these features are absent in the labeled data, over 15 EM iterations they are getting desired positive weights. As an effect of these changes in the feature weights, the instance `(Peanuts, limb rot)` gets correctly classified as an instance of relation *Affects*.

Error analysis of the difficult cases highlights two important issues to be handled. First and most frequent issue is that of "noisy labels" which we discussed earlier. Enriching gazettes or adding some more robust features might help in this case, but more analysis is required. Another issue is presence of negation. Consider the following sentences:

1. `The fact that [rust]`$_{DISEASE}$ `did not affect Midwest [soybean]`$_{CROP}$ `production in 2005 probably squelched the impulse to hire a consultant for the 2006 crop season.`
2. `But even in Southern areas, where [aflatoxin]`$_{DISEASE}$ `is also no stranger to [cotton]`$_{CROP}$`, rotation may not always work.`

Here, in the first sentence, due to the presence of negation, the *Affects* relation does not hold. But in the second sentence, the *Affects* relation holds even if negation is present between the two entities. The features using deep semantic knowledge might help but more investigation is needed in this case.

5 Conclusion and Future Work

In this paper, we proposed a systematic bootstrapping formulation of relation extraction problem using EM algorithm. We demonstrated the efficacy of this sound theoretical framework exploiting easily obtainable unlabeled data by showing an improvement in the relation extraction accuracy. We presented results of our approach on two different corpora of news articles- one from a general domain and other from the agriculture domain.

In the future work, we plan to test our approach on the ACE 2004 data (Doddington et al., 2005) which has gold-standard relation labels. This will enable us to compute the actual recall of our approach and then we can also compare our performance on that data with others. We also want to consider the relations with more granularity (e.g. considering more specific relations like *FounderOf* or *EmployedBy* rather that the general relation *Role*), which we could not consider because of insufficient labeled data available currently. Another interesting experiment would be to replace the discriminative MaxEnt models with some generative models like Naive Bayes and compare the performance. We also plan to extend our problem formulation to handle the case where even entity mentions are unknown and study how both the tasks of relation extraction and entity extraction help each other using EM algorithm.

References

Kamal Nigam, AK McCallum, S Thrun, T Mitchell 2000. *Text Classification from Labeled and Unlabeled Documents using EM* Machine Learning, 39, 103-134.

Zhou GuoDong, Su Jian, Zhang Jie, Zhang Min 2005. *Exploring Various Knowledge in Relation Extraction* Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (pp. 427-434).

Jing Jiang and ChengXiang Zhai 2007. *A Systematic Exploration of the Feature Space for Relation Extraction* HLT-NAACL (pp. 113-120).

Ang Sun, Ralph Grishman and Satoshi Sekine 2011. *Semi-supervised Relation Extraction with Large-scale Word Clustering* ACL 2011.

Takaaki Hasegawa, Satoshi Sekine and Ralph Grishman 2004. *Discovering Relations among Named Entities from Large Corpora* Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella 2003. *Kernel methods for relation extraction* Journal of Machine Learning Research 2003, 3:1083-1106.

Razvan C. Bunescu and Raymond J. Mooney 2005. *Subsequence kernels for relation extraction* Advances in neural information processing systems (pp. 171-178).

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, Ralph Weischedel 2004 *The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation* LREC 2004.

Patil, S., Pawar, S., Palshikar G. K., Bhat S., Srivastava R. 2013 *Unsupervised Gazette Creation Using Information Distance* Natural Language Processing and Information Systems, Pages 388-391, Springer Berlin Heidelberg.

Blum, Avrim, and Tom Mitchell 1998 *Combining labeled and unlabeled data with co-training* Proceedings of the eleventh annual conference on Computational learning theory. ACM, 1998.

S Brin 1999 *Extracting patterns and relations from the world wide web* The World Wide Web and Databases 1999.

Agichtein and Gravano 2000 *Snowball: extracting relations from large plain-text collections* Proceedings of the fifth ACM conference on Digital libraries, Pages 85-94.