QA-Driven Relation Extraction

Zeqiu Wu
Computer Science
University of Illinois at Urbana-Champaign
Email: zeqiuwu1@illinois.edu

Anirud Yadav
Computer Science
University of Illinois at Urbana-Champaign
Email: ayadav4@illinois.edu

Wenxuan Zhou
Computer Science
University of Illinois at Urbana-Champaign
Email: wenxuan3@illinois.edu

Abstract—Information Extraction (IE) is concerned with mining factual structures from unstructured text data, including entity and relation extraction. For example, identifying Donald Trump as “person” and Washington D.C. as “location”, and understand the relationship between them (say, Donald Trump spoke at Washington D.C.), from a specific sentence. Typically, IE systems rely on large amount of training data, primarily acquired via human annotation, to achieve the best performance. But since human annotation is costly and non-scalable, the focus has shifted to adoption of a new strategy - Distant Supervision [1]. Distant supervision is a technique that can automatically extract labeled training data from existing knowledge bases without human efforts. However the training data generated by distant supervision is context-agnostic and can be very noisy. Moreover, we also observe the difference between the quality of training examples in terms of to what extent it infers the target entity/relation type. In this project, we focus on removing the noise and identifying the quality difference in the training data generated by distant supervision, by leveraging the feedback signals from one of IE’s downstream applications, QA, to improve the performance of one of the state-of-the-art IE framework, CoType [3].

Keywords—Data Mining, Relation Extraction, Question Answering.

I. INTRODUCTION

A vast majority of data on the internet is unorganized and present in the form of text. Hence it is of vital importance to develop a smart IE system which can extract the entities mentioned in the text along with the relations they share with each other, with minimal human efforts. CoType is one such state-of-the-art tool which uses distant supervision to generate the training data. Though CoType has been shown to have better accuracy than most of its competitors, it still suffers from some key limitations. For the purpose of our project we can focus on two key observations: noisy training data from distant supervision and ignorance of quality difference in training example.

Noisy training data: Even though distant supervision is a scalable technique, it also introduces significant noise in the training data that it generates. The noise in training data is because of the false negatives and false positives. False negatives are encountered because most of the online knowledge bases suffer from incompleteness. That is there are many relation and entity mentions which simply cannot be linked to the knowledge bases. For example, for an input sentence like “Jack plays for Chicago Bears” might be unlinkable via distant supervision as the knowledge base being used by distant supervision might not have the entity Jack present in it. Rather only a small percentage of the extracted entities and relations mention can be linked directly to the knowledge bases. Consequently, since in order to generate negative training data, a model has to heuristically select negative example from the unlinkable pairs, it results into introduction of false negatives.

The other reason contributing to noise are the false positives which are a result of context-agnostic labelling. The assumption that any two entities which are present in a sentence must be related is not always true. For example, as shown in figure 1, for an input sentence like Obama was born in Honolulu, Hawaii, USA as he always said., distant supervision might assign a relation of “president_of” between the entity mentions “Obama” and “USA”, which is certainly not true here. Hence, distant supervision can lead to generation of a lot of false positive because of its context-agnostic behaviour.

Ignorance of quality difference: The second observation we made was that CoType treats all of the relations mentions equally while constructing the heterogeneous information network. This is not true because not all the sentences reflect the relation type between a pair of entity mentions with same degree of confidence. For example, the relation type “president_of” between entity mentions “Donald Trump” and “USA” is expressed better in sentence “Donald Trump is the president of USA” rather than the sentence “Donald Trump sits in the most respected seat in the USA”.

Fig. 1: Noise Training Data Example

Fig. 2: Ignorance of Quality Difference Example
Hence in this paper, we aim to tackle the above mentioned shortcomings by leveraging feedback from a QA system, Jacana. For the purpose of our project we focus on the following target entity types: “Person”, “Location” and “Organization” and the relations between them. Given an input corpus, we first plan to get training data via distant supervision and then eliminate the false positives and false negatives by using Jacana to assign a score to each sentence containing an entity pair based on the confidence of the relation type between them. Finally these scores are fed into the CoType pipeline as weights for each corresponding relation mention. In section II of the paper, we will discuss the background and formulate the problem. In section III, we cover the framework overview. In section IV, we show the results of our experiments. In section V, we go over some related work. Finally in section VI and VII, we draw conclusion based on our experiments and discuss potential future work.

II. BACKGROUND AND PROBLEM

The input to our framework is:

- a POS tagged text corpus $D$
- a knowledge base $\psi$
- target entity type set $\mathcal{Y}$
- target relation type set $\mathcal{R}$
- and a QA system $Q$.

The text corpus $D$, knowledge base $\psi$, target entity type set $\mathcal{Y}$ and target relation type set $\mathcal{R}$ serve as an input to CoType and it then performs joint extraction of typed entity and relation instances by embedding them into a lower dimensional space. The problem is formulated as a joint optimization with the aim to embed entity and relation vectors with similar types in a close vicinity. The key terminologies to know in Cotype are: entity and relation mention, knowledge bases and target types and automatically labelled training data.

An entity mention ($m$) is a token span in the text which represents an entity $e$ and a relation mention $r(e_1, e_2)$ is a tuple of two entity mentions which occur in a sentence and denote some type of relation. In our work, we do not consider more complicated relations which involve more than 2 entities participating in a relation. Hence a relation mention is always a tuple containing two entities.

A knowledge base (KB) is set of human curated facts on both relation and entity instances. The facts related to our work are the relation instances $\mathcal{L}_\psi = \{r(e_1, e_2) \in \mathcal{R}_\psi \times \mathcal{E}_\psi \times \mathcal{E}_\psi\}$ and entity types $\mathcal{T}_\psi = \{e, y\} \in \mathcal{E}_\psi \times \mathcal{Y}_\psi$. Each relation mention in CoType can have multiple relation types associated with it.

The automatically labelled training data is generated using distant supervision. Let $M = \{m_i\}_{i=1}^N$ denote the entity mentions extracted from the text corpus $D$. This set $M$ is then linked to knowledge base KB and the set of linkable entities are represented by $\mathcal{M}_L$. Then a set of relation mentions are constructed using all the linkable entity mentions which occur in a sentence. That is for each linkable entity pair $e_1$ and $e_2 \in \mathcal{M}_L$ which occur in a sentence, two relation mentions are constructed: $r(e_1, e_2)$ and $r(e_2, e_1)$. Let $\mathcal{Z}$ be the set of all such relations. The set $\mathcal{Z}_L$, unlinkable relation mentions and false relation mentions. The false relation mentions have no valid relationship between their constituent entity pairs. Let $\mathcal{Z}_U = \mathcal{Z}/\mathcal{Z}_L$ be the set of unlinkable relation mentions.

Consequently for each such generated relation mention $z_i \subset \mathcal{Z}_L$, a candidate relation type set $\mathcal{R}_i = \{r|r(e_1, e_2) \subset \mathcal{R}_\psi\}$ is generated by fetching all the relations between $e_1$ and $e_2$ in KB. Similarly the types for each entity mention $m_1$ and $m_2$ ($y_{i,1}$ and $y_{i,2}$) are generated using KB where $y_{i,x} = \{y|y(e_x, y) \subset \mathcal{Y}_\psi\}$. Let $\mathcal{Z}_L$ be the set of extracted relation mentions that can be linked to KB. Hence the training data for CoType is $D_L = \{z_i, r_i, y_{i,1}, y_{i,2}\}_{i=1}^N$.

Finally we use Jacana which is a answer sentence selection tool in order to reduce the noise in the training data $D_L$. For each question $Q$ and a set of candidate answer set $A$, Jacana assigns a score in the range [0,1]. Higher the score, the more reliable the answer. For our paper, we first generate placeholder questions for each relation type. Then for each linkable relation mention $z \subset \mathcal{Z}_L$, we generate one question per its linked relation type by replacing the placeholder with the appropriate entity mention from $z$. Similarly for each unlinkable relation mention, we generate a set of few related questions.

**Problem Description**: Our main task is to first denoise the training data using Jacana and then perform joint extraction using CoType on this denoised training dataset to achieve a higher accuracy.

**Definition 1 (Problem Definition)**: Given a POS-tagged corpus $D$, a KB ($\psi$), a target entity type hierarchy $\mathcal{Y} \subset \mathcal{Y}_\psi$ and a target relation type set $\mathcal{R} \subset \mathcal{R}_\psi$, our task aims to (1) detect entity mentions $M$ from $D$; (2) generate training data $D_L$ with KB ($\psi$); (3) denoise $D_L$ using Jacana to get $D_{NL}$ and (4) then estimate a relation type $r \in \mathcal{R} \cup \{None\}$ for each test relation mention $z \in \mathcal{Z}_U$ and an entity type $\mathcal{Y} \cup \{None\}$ for each entity mention in $z$, using $D_{NL}$ and its context $s$.

III. THE FRAMEWORK OVERVIEW

In this section, we will first introduce Jacana-QA [19], the QA model used in our system in section 3.1. In section 3.2, we will briefly go through the pipeline in CoType and the details of the component where Jacana will provide feedback signals. Finally, we will discuss how we will incorporate Jacana-QA into CoType in order to improve the relation extraction process.

A. Jacana-QA

Jacana-QA is a text-based QA system developed for the task that given a text corpus and a factoid question, predict the answer for the question from the text corpus. Similar to a general text-based QA system, Jacana-QA consists of all the following components: (1) Informational retrieval for selecting a subset of articles from the corpus as the main focus; (2) Passage retrieval / answer sentence selection for an even narrower focus on a subset of sentences from the articles; (3) Answer extractor for producing the final answer(s).
In our project, we will only care about the answer sentence selector component of Jacana-QA. This component of Jacana-QA is a tree edit model and has been separately trained and tested on the public dataset [23] for the task of answer sentence selection, which is defined as to identify which sentence(s) contain the answer to a given question. Fig. 3 shows an example of the prediction results that can come from an answer sentence extractor. In the training process, the answer sentence selector in Jacana-QA learns a linear regression classifier based on the feature vector of each candidate sentence as well as the corresponding question. Thus, in the prediction process, it will assign a rank score to each candidate sentence from the learnt classifier besides the positive/negative label.

The answer sentence selector in Jacana-QA uses all the 33 features defined in [24] with additional 15 features summarized in Table I.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance</td>
<td>tree edit distance from answer sentence to question</td>
</tr>
<tr>
<td>renNoun, renVerb, renOther</td>
<td># edits changing POS from or to noun, verb, or other types</td>
</tr>
<tr>
<td>insN, insV, insPunc, insDet, insOtherPos</td>
<td># edits inserting a noun, verb, punctuation mark, delimiter or other POS types</td>
</tr>
<tr>
<td>delIN, delIV, ...</td>
<td>deletion mirror of above</td>
</tr>
<tr>
<td>ins{N,V,P}Mod, insSub, insObj, insOtherRel</td>
<td># edits inserting a modifier for {noun, verb, preposition}, subject, object or other relations</td>
</tr>
<tr>
<td>delINMod, ...</td>
<td>deletion mirror of above</td>
</tr>
<tr>
<td>renNMMod, ...</td>
<td>rename mirror of above</td>
</tr>
<tr>
<td>XEdits</td>
<td># basic edits plus sum of ins/del/ren edits</td>
</tr>
<tr>
<td>alignNodes, alignNum, alignN, alignV, align Proper</td>
<td># aligned nodes, and those that are numbers, nouns, verbs, or proper nouns</td>
</tr>
</tbody>
</table>

**TABLE I: Features for answer sentence selection**

**B. CoType**

CoType [3] is one of the state-of-the-art information extraction system, which jointly extracts typed entity and relation mentions in each sentence of the corpus, with labeled training data heuristically obtained from knowledge bases (distant supervision). This joint extraction addresses the issue of error propagation in traditional incremental extraction pipeline.

1) **CoType pipeline:** Firstly, CoType uses a data-driven text segmentation technique, SegPhrase [26], to find token spans as candidate entity mentions. Then it links these candidate entity mentions to Freebase and remain those linkable ones as training typed entity mentions. For each pair of those linkable entities in a sentence, it again checks whether the entity pair can link to a relation tuple in Freebase with the target relation type. Those linkable entity pairs will be the positive relation mentions for training and the negative examples will be randomly selected from those unlinkable entity pairs.

After the step of training data generation from distant supervision, CoType will jointly learn the 2 embedding spaces for entity and relation mentions respectively. In each embedding space, a heterogeneous network is constructed with node type as feature, mention or type. The visualization of the embedding spaces that CoType is trying to learn can be seen in Fig. 4.

**Fig. 4: Embedding spaces in CoType**

With the learnt embeddings for all features, entity/relation types, CoType will represent each test mention by aggregating the mapped feature vectors and infer its type by comparing with each types embedding.

2) **Embedding Space Learning:** In this subsection, we will discuss more details of the embedding space learning process of CoType, in order to show how our methodology can be incorporated.

To learn embedding for all nodes in the 2 mention spaces, CoType formalizes a objective function to minimize, which consists of three components, one from each of the two mention spaces and one from the interactions among the two spaces.

Taking relation mention embedding space as an example, its objective function is formed based on two hypothesis: 1) Two relation mentions tend to share similar types (close to each other in the embedding space) if they share many text features in the corpus, and the converse way also holds; 2) A
relation mention’s embedding vector should be more similar (closer in the low-dimensional space) to its “most relevant” candidate type, than to any other non-candidate type.

They apply second-order proximity[27] to model the first hypothesis as follows. Formally, let vectors $z_i, c_j \in \mathbb{R}^d$ represent relation mention $z_i \in Z_L$ and text feature $f_j \in F_z$ in the $d$-dimensional relation embedding space.

\[
ZF = - \sum_{z_i \in Z_L} \sum_{f_j \in F_z} w_{ij} \cdot \log p(f_j | z_i),
\]

where $p(f_j | z_i) = \exp(z_i^T c_j) / \sum_{f' \in F_z} \exp(z_i^T c_{f'})$ denotes the probability of $f_j$ generated by $z_i$, and $w_{ij}$ is the co-occurrence frequency between $(z_i, f_j)$ in corpus $D$.

For the second hypothesis, they define a partial-label loss $\ell_i$ for each relation mention $z_i \in M_L$ as follows, using vector $r_k \in \mathbb{R}^d$ to represent relation type $r_k \in R$.

\[
\ell_i = \max \left\{ 0, 1 - \left[ \max_{r \in R_i} \phi(z_i, r) - \max_{r' \in R_i} \phi(z_i, r') \right] \right\}. \tag{2}
\]

Adding regularization terms with tuning parameter $\lambda$, the final objective function for relation mention embedding space is:

\[
O_Z = ZF + \sum_{i=1}^{N_L} \ell_i + \frac{\lambda}{2} \sum_{i=1}^{N_L} \|b_i\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K_r} \|r_k\|_2^2. \tag{3}
\]

Similarly, it has a objective function, $O_M$, for the entity mention embedding space.

Finally, it models the interactions between entity and relation mentions based on the third hypothesis: For a relation mention $z = (m_1, m_2, s)$, embedding vector of $m_1$ should be a nearest neighbor of the embedding vector of $m_2$ plus the embedding vector of relation mention $z$. And by defining the error function of a relation mention triple as $\tau(z) = \|m_1 + z - m_2\|_2^2$, the third objective function is formulated as follows.

\[
O_{ZM} = \sum_{z_i \in Z_L} \sum_{v=1}^{V} \max \left\{ 0, 1 + \tau(z_i) - \tau(z_v) \right\}, \tag{4}
\]

where $\{z_v\}_{v=1}^{V}$ are negative samples for $z$, i.e., $z_v$ is randomly sampled from the negative sample set $\{(z', m_1, m_2)\} \cup \{(z, m_1', m_2)\} \cup \{(z, m_1, m_2')\}$ with $z' \in Z_L$ and $m_1', m_2' \in M_L$.

Combining all these three objective functions, CoType eventually tries to learn all the embeddings by minimizing the following global objective function in the training process.

\[
O = O_M + O_Z + O_{ZM}. \tag{5}
\]

C. QA-Driven Relation Extractor

Although CoType proves to be noise-robust by introducing the joint extraction as well as the partial-label loss technique in their framework and the results shown in their paper outperforms all the other baselines on three different datasets, we still observe limitations existing in the system and we will introduce our proposed solution to these two challenges.

With training data generated from distant supervision, there could be a lot of false positive and negative relation mentions linked from the knowledge base. If we try to minimize the loss of these noisy training examples, it can even counteract the performance. Another challenge we observed not only exist in CoType, but also in other general relation extractors. In many cases, whether there exists a relationship between two entities in a sentence is very subjective due to the ambiguity in the expression of the sentence. Some relation mentions are very straightforward while some are very hard to identify. In other words, different relation mentions have different quality levels in indicating the relation type. Such quality level difference not only exists in positive examples, but also exists in negative examples. However, CoType does not capture such differences between training examples and it basically treats all training examples equally. This can be observed from the embedding space learning process introduced above. In Eq. (1), (3) and (4), it can be seen that in each component of the final objective function, CoType always does summation of some local loss function over each training example, without any differentiation design.

With these two observed issues existing in CoType, we would like to leverage the learning-to-rank mechanism in Jacana-QA to provide feedback signals to enhance the training process of CoType. Our goal is to assign a global rank score to each of the training relation mention from distant supervision, in order to update its label to the most possible one as well as its quality score, which is its weight.

D. Pipeline

In order to get a feedback rank score from Jacana-QA for each of our training example, we would need to map each training relation mention to at least one pair of question and sentence as the input for Jacana-QA due to the nature of an answer sentence selector. Our framework is shown in Fig. 5.

![Fig. 5: The Framework](image-url)
sentence pair(s) for each relation mention as the input for Jacana-QA to calculate rank scores. (5) Run Jacana-QA on the constructed question and sentence pairs to get rank scores. (6) Update the label as well as the weight of each training example based on the rank scores. (7) Train CoType with the updated training dataset.

In our project, the most important parts are step 4 (Question and sentence generation) and 5 (Training data update), which are discussed in the following subsections.

1) Question and Sentence Generation: First of all, we map each target relation type to a question template with a placeholder, which is expected to be replaced by the first entity mention involved in a relation mention. We try to make the question very straight forward in expressing the relation type. For example, for a relation type of “place_of_birth”, we will have the question template as “Where was $PERSON born?” Table 3 shows some example relation type and question template pairs.

<table>
<thead>
<tr>
<th>Relation Type</th>
<th>Question Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>sports_team_location</td>
<td>Which sports team is from $LOCATION?</td>
</tr>
<tr>
<td>child_of_person</td>
<td>Who is the child of $PERSON?</td>
</tr>
<tr>
<td>advisor_of_company</td>
<td>Who advises $ORGANIZATION?</td>
</tr>
<tr>
<td>capital_of_country</td>
<td>What is the capital of $LOCATION?</td>
</tr>
<tr>
<td>place_of_birth</td>
<td>Where was $PERSON born?</td>
</tr>
</tbody>
</table>

TABLE II: Example Question Templates

With this relation type to question template mapping, for each positive relation mention, we will construct a question for it by replacing the placeholder in the corresponding question template with the first entity mention. For instance, say we have a relation triple (“Donald Trump”, “place_of_birth”, “New York”), our model will generate the question “Where was Donald Trump born?” for it. Putting the question together with the sentence where the relation mention is located, it basically generates the input for Jacana-QA to predict the rank score for this relation mention.

For each negative relation mention, things get a little bit trickier. Instead of generating one pair of question and sentence as for a positive example, we need to generate a question per possible relation type for each negative relation mention. The possible relation types for a negative relation mention are defined as having the same entity type signature as the entity types of this relation mention. For instance, say we have a negative relation mention (“Jim Trump”, “None”, “Chicago”), where “Jim Trump” is a person and “Chicago” is a location. Then since relation types like “place_of_birth”, “place_travelled” and so on also have their entity types as (person, location), we will generate a question for each of such relation types. Thus, we will end up getting questions: “Where was Jim Trump born?” “Where did Jim Trump travel?” and so on, for this relation triple. And also, each question will be combined with the sentence of the relation mention to form the input for Jacana-QA.

2) Training Data Update: After we generate all the question and sentence pairs from the previous step and use Jacana-QA to get a rank score for each of the pairs, we would like to update the label and weight of each relation mention based on these assigned scores.

The range of the rank score from Jacana-QA is in the range $[0,1]$ and 0.5 is the score cutoff for dividing positive and negative sentences.

For each positive relation mention, there will only be one rank score $s$ assigned to it. We have a new cutoff score $\theta_1$ such that if $s$ is above $\theta_1$, this relation mention will remain as positive with the original label. Otherwise, it will become a negative example. $\theta_1$ is expected to be smaller than 0.5 because there is a tolerance gap for the relation mentions linkable to the knowledge base. To calculate the weight for each relation mention, we will use a normalized rank score to be the weight, $w$. In each of these two cases (change label or not), we will normalize $s$ such that it is in the range of $[0,5,1.5]$ so that the average $w$ can be controlled to be approximately 1.0:

$$w = \begin{cases} \frac{1 - \theta_1}{1 - \theta_1} + 0.5 & s > \theta_1 \\ \frac{\theta_2 - \theta_1}{\theta_2 - \theta_1} + 0.5 & \text{otherwise} \end{cases}$$

For each negative example, there can be one or more rank scores assigned to it. We have another cutoff score $\theta_2$, such that if the maximum $s$ is above $\theta_2$, this relation mention will be assigned to the corresponding label of the maximum $s$. Otherwise, it will remain as a negative example. $\theta_2$ is expected to be higher than 0.5 because there should be tighter constraint for the unlinkable relation mention to be converted into a positive example. Again, to calculate the weight for each relation mention, we will use a normalized rank score to be the weight, $w$. In each of these two cases (change label or not), we will normalize $s$ such that it is in the range of $[0,5,1.5]$ so that the average $w$ can be controlled to be approximately 1.0:

$$w = \begin{cases} \frac{1 - \theta_2}{1 - \theta_2} + 0.5 & s > \theta_2 \\ \frac{s}{\theta_2} + 0.5 & \text{otherwise} \end{cases}$$

IV. Experiments

A. Datasets

In our project, the answer sentence extractor in Jacana-QA and CoType are trained separately, for different tasks. Thus, they have different datasets and we will discuss both of them.

1) Answer Sentence Extraction: For Jacana-QA, we use the public dataset generated by [23] specifically for the task of answer sentence selection. This dataset is constructed semi-automatically and consists of factoid questions from TREC8-13 QA dataset and candidate sentences from TREC8-13 QA text corpus for each question. TREC8-13 QA text corpus includes news articles from Associated Press Worldstream, Xinhua and New York Times. The following table shows the statistics in this dataset.
Jacana-QA achieves the MRR score of 0.748 on the test data. Although compared to the state-of-the-art deep neural model [25], which achieves 0.845, this performance is not that impressive, it is enough to boost the relation extraction performance in our project.

2) Relation Extraction: For CoType, the information source of training data is still TREC8-13 text corpus, the same as Jacana-QA. We preprocessed randomly selected sentences from TREC8-13 news articles and linked the candidate entity/relation mentions to Freebase to get training examples. Those unlinkable relation mentions are marked as negative examples, with relation type 'None'.

Since we did not find any ground truth dataset from TREC corpus for the task of relation extraction. We used the New York Times (NYT) test dataset in [3]. As both TREC and this NYT corpus are in news domain and actually one third of the articles in TREC corpus are from New York Times, it is reasonable to have this NYT dataset as our test data. We have 3 target entity types (Person, Location, Organization) and 18 target relation types.

The following table gives the statistics of the training and test datasets:

<table>
<thead>
<tr>
<th>set</th>
<th>source</th>
<th>#ques.</th>
<th>#pairs</th>
<th>%pos.</th>
<th>len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAIN-ALL</td>
<td>TREC8-12</td>
<td>1229</td>
<td>53417</td>
<td>12.0</td>
<td>any</td>
</tr>
<tr>
<td>TRAIN</td>
<td>TREC8-12</td>
<td>94</td>
<td>4718</td>
<td>7.4</td>
<td>≤ 40</td>
</tr>
<tr>
<td>DEV</td>
<td>TREC13</td>
<td>82</td>
<td>1148</td>
<td>19.3</td>
<td>≤ 40</td>
</tr>
<tr>
<td>TEST</td>
<td>TREC13</td>
<td>89</td>
<td>1517</td>
<td>18.7</td>
<td>≤ 40</td>
</tr>
</tbody>
</table>

**TABLE III: Jacana-QA Dataset Statistics**

Here is the formula for calculating each score:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (6)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (7)
\]

\[
F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)
\]

where TP = the number of true positives, FP = the number of false positives and FN = the number of false negatives.

D. Results

The following table shows the precision, recall and F1 scores for each of the three models. As we can see, our full version model outperforms the other two models in terms of precision and recall. And our update-label-only model achieves the highest recall score.

<table>
<thead>
<tr>
<th>Model</th>
<th>precision</th>
<th>recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoType</td>
<td>0.264</td>
<td>0.335</td>
<td>0.295</td>
</tr>
<tr>
<td>CoType_QA_Label</td>
<td>0.219</td>
<td>0.501</td>
<td>0.305</td>
</tr>
<tr>
<td>CoType_QA_Full</td>
<td>0.317</td>
<td>0.355</td>
<td>0.335</td>
</tr>
</tbody>
</table>

**TABLE V: Experiment Results**

E. Case Study

Here are two interesting cases showing how the rank scores from Jacana-QA help update the label and the weight of each relation mention in Table VI.

In the first sentence, although “Israel” and “Tel Aviv” are linked as having the relation “contains” from Freebase, however, based on the ranking score from Jacana-QA, the sentence does not express such relation. Thus, it has been converted into a negative example with the weight 0.87. In the second sentence, Jacana-QA still saves the relation mention as the positive example while downgrades its weight to below 1.0.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Original Relation Mention</th>
<th>Updated Relation Mention</th>
<th>Updated Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>In 1985, Jordan regained sovereignty over its territories occupied by Israel and exchanged ambassadors with Tel Aviv.</td>
<td>(Israel, contains, Tel Aviv)</td>
<td>(Israel, None, Tel Aviv)</td>
<td>0.87</td>
</tr>
<tr>
<td>Cameron spent the night at Mercy Hospital near downtown San Diego, and Beltran spent the night at Scripps Clinic in La Jolla.</td>
<td>(Mercy Hospital, located in, San Diego)</td>
<td>(Mercy Hospital, located in, San Diego)</td>
<td>0.72</td>
</tr>
</tbody>
</table>

**TABLE VI: Case Study**
Traditional supervised IE systems require costly human annotation and have domain restriction. The later developed weak supervision still needs a small set of examples and requires very careful selection. Since [1] proposed the distant supervision, which can automatically extract labeled training data from existing knowledge base without human effort, many relation extraction systems [2][3] based on DS have been proposed.

One problem of distant supervision is that it assumes all sentences containing an entity pair to be potential patterns for the relation holding between the entities, so it may introduce a lot of noise to the training data. For example, Trump is the president of US, but not every sentence containing those entities expresses the relation. Different types of methods for noisy reduction were explored and found useful. Some methods [1][4] directly reduced the error by using high-quality text corpus, for example, encyclopedic entries. Some methods loosened the requirement of distant supervision. For example, methods in [2][5] assumes that for each relation triple in the database, at least one sentence might express the relation (instead of all sentences). For each sentence, they will make multiple predictions and use multi-instance learning algorithm to infer the types. [3] reduced the noisy by requiring the entity/relation in the training examples to be close to its closest type label. [6] proposed a method to model the missing data by using side information, for instance, the experience that text tends to mention rare entities in the knowledge base. Some methods leveraged topic modeling to classify the relations and features. The work in [7] used hierarchical topic models to identify those patterns which are expressing the relation and the agnostic patterns. [8] proposed a method to assign larger weights to distinctive features than noisy ones. It makes use of all the training data and then employ unsupervised method by topic model to discover the distribution of features to latent relations. Some methods detect reliable positive and negative methods by checking the semantic consistency between labeled data. [9] proposed a method to reduce the uncertainty of training data by incorporating a wide range of indirect supervision knowledge and jointly inferring types across relation instances. The methods in [10][11] modeled the semantic similarity between the contexts of training data and filtered noisy examples. Some methods combined several models in one pipeline and achieved better performance. The work in [12] combined the at-least-one learner and topic modeling to reduce the noise. The work in [13] combined fine-grained NER, negative example construction and classifier ensembling components. Another type of method is to leverage feedback from downstream applications. [14] proposed that using the feedback from an information retrieval system can help detect false negative examples from distant supervision. The model expands the knowledge base with possibly missing relation instances with information from the highest ranked sentences returned by passage retrieval model trained on the same data. The assumption is that entity pairs appearing in more relevant and more sentences are more likely to express the relation, so the noise can be reduced.

Traditional methods [15][16] for relation extraction partition the process to several steps. They first use entity extraction and type labeling, then extract the relations. Such methods are error-prone and cannot model the interaction between entity and relation types. Recent methods [3][17][18] jointly extract the entities and relations so they can incorporate the type constraints between entities and relations.

Feature engineering is the dominant method for answer selection in text question answering system. Traditional methods model answer selection as a classification problem and use features (number of named entities, semantic type etc.) and classification algorithms (like SVM). Recent research models it as a ranking problem, for each sentence, a score representing the probability of containing desired answer is given. And some new features and learning algorithms have been explored. One type of feature is dependency tree structure. Jacana [19] used the tree edit distance of dependency trees of question and candidate sentence as a new feature and use linear-chained CRF to learn the sequence labels, which is more robust against error propagation in NLP pipeline. [20] proposed an automatic method for building structural new features. Some methods [21][22] use neural network to learn the confidence score and achieve the state-of-art performance.

VI. Conclusion

This project studies error reduction in distant supervision by leveraging the feedback signals from the question answering system to the relation extraction. The framework transforms the relation instances obtained from distant supervision to questions and sentences as input for Jacana, and use the rank scores to update the labels and weights of training examples. Experiment demonstrates the effectiveness of this framework to reduce the noisy in training data and improve the performance of CoType. Specifically, the framework boosts the F1 value on the TREC dataset significantly by 4%.

VII. Future Work

Some possible improvements to the framework havent been fully explored. Now we only use the feedback from QA system to clean the training data. The extracted typed entities and relations of CoType can also serve as features for QA, so that IE and QA may mutually enhance each other.

Secondly, the current method of updating weights of training examples is heuristic. How to formalize the process as an optimization problem with systematic mathematical equations remains as a problem to be solved in the future.

Moreover, for each relation triple (e1, r, e2), we only construct one question asking for e1. For instance, for sentence \textit{Trump is the president of US}, the constructed question is \textit{Who is the president of US}, while the other possible question is \textit{Which country is Trump president of}. Averaging the rank scores of two questions may lead to better performance. In the experiment only 3 entity types and 18 relation types are used, other relation mentions are labeled as “No Relation”. Expanding the current target type set is also a direction to go.

Finally, replacing Jacana-QA with a better performing QA model or ensembling of multiple QA models also has the potential to help us improve the result.
ACKNOWLEDGMENT

We would like to thank Professor Han and Xiang Ren for his supervision over this project. We would also like to thank Hongwei Ng, Xiaoxiao Wang and Xinyu Zhang for their help on setting up Jacana-QA to be incorporated into our system.

REFERENCES


