INDIRECT SUPERVISION FOR RELATION EXTRACTION USING QUESTION-ANSWER PAIRS

BY

ZEQIU WU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Professor Jiawei Han

ABSTRACT

Automatic relation extraction (RE) for types of interest is of great importance for interpreting massive text corpora in an efficient manner. For example, we want to identify the relationship "president_of" between entities "*Donald Trump*" and "*United States*" in a sentence expressing such a relation. Traditional RE models have heavily relied on human-annotated corpus for training, which can be costly in generating labeled data and become obstacles when dealing with more relation types. Thus, more RE extraction systems have shifted to be built upon training data automatically acquired by linking to knowledge bases (distant supervision). However, due to the incompleteness of knowledge bases and the context-agnostic labeling, the training data collected via distant supervision (DS) can be very noisy. In recent years, as increasing attention has been brought to tackling question-answering (QA) tasks, user feedback or datasets of such tasks become more accessible. In this paper, we propose a novel framework, ReQuest, to leverage question-answer pairs as an indirect source of supervision for relation extraction, and study how to use such supervision to reduce noise induced from DS. Our model jointly embeds relation mentions, types, QA entity mention pairs and text features in two low-dimensional spaces (RE and QA), where objects with same relation types or semantically similar question-answer pairs have similar representations. Shared features connect these two spaces, carrying clearer semantic knowledge from both sources. ReQuest, then use these learned embeddings to estimate the types of test relation mentions. We formulate a global objective function and adopt a novel margin-based QA loss to reduce noise in DS by exploiting semantic evidence from the QA dataset. Our experimental results achieve an average of 11% improvement in F1 score on two public RE datasets combined with TREC QA dataset. Codes and datasets can be downloaded at https://github.com/ellenmellon/ReQuest.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would firstly like to thank my advisor Professor Jiawei Han of the Department of Computer Science at University of Illinois at Urbana-Champaign. He has provided me extensive support and guidance not only on this thesis project, but also throughout my whole master research journey. He guided me to start my exploration in academic research two years ago and mentored me to achieve various research milestones. I really appreciate that Professor Han helped spark my interest in research, so that I become determined to continue my research career by pursuing a Ph.D. degree. It was my great fortune to join the Data Mining Research Group and get the chance to work with such a brilliant group of peer researchers. Each of them has provided me valuable ideas and help, both in life and study. I especially want to thank Xiang Ren, for being a peer mentor throughout the past two years. I would not be able to conduct research projects that smoothly without his guidance.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 DEFINITIONS AND PROBLEM 5

CHAPTER 3 APPROACH 7

3.1 Heterogeneous Network Construction 7

3.2 Joint RE and QA Embedding 8

3.3 Type Inference 13

CHAPTER 4 EXPERIMENTS 14

4.1 Data Preparation and Experiment Setting 14

4.2 Experiments and Performance Study 16

4.3 Case Study 17

CHAPTER 5 RELATED WORK 20

CHAPTER 6 CONCLUSION 22

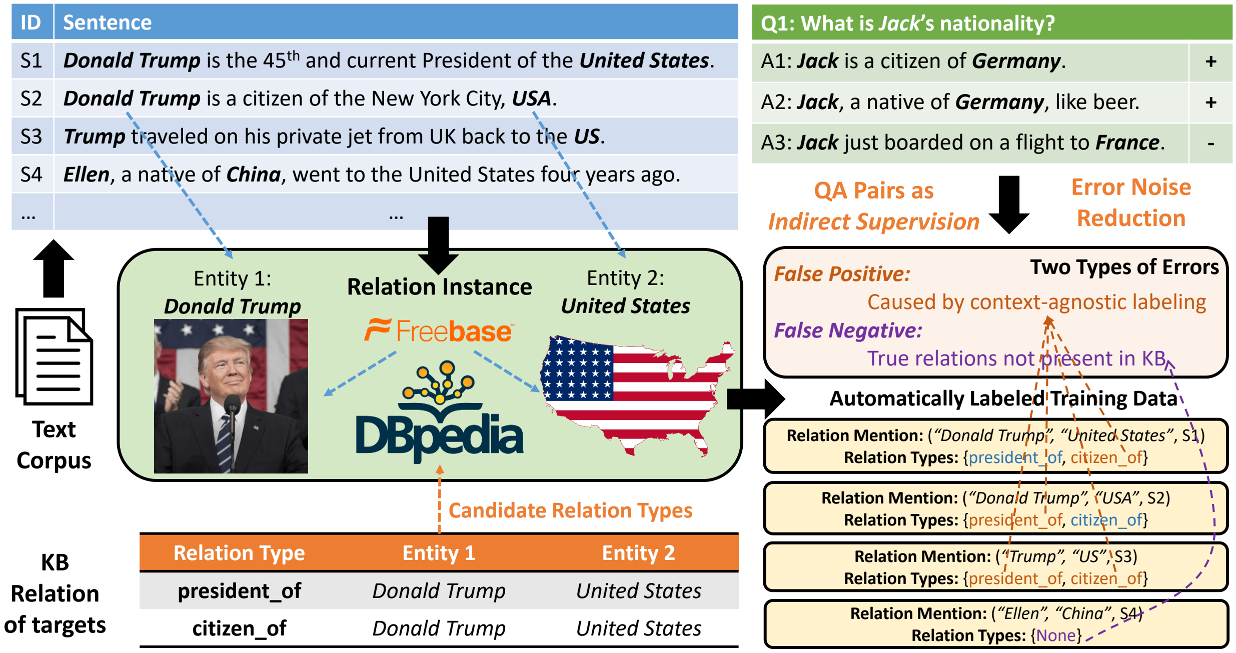
REFERENCES 23

CHAPTER 1: INTRODUCTION

Relation extraction is an important task for understanding massive text corpora by turning unstructured text data into relation triples for further analysis. For example, it detects the relationship "`president_of`" between entities "*Donald Trump*" and "*United States*" in a sentence. Such extracted information can be used for more downstream text analysis tasks (e.g. serving as primitives for information extraction and knowledge base (KB) completion, and assisting question answering systems).

Typically, RE systems rely on training data, primarily acquired via human annotation, to achieve satisfactory performance. However, such manual labeling process can be costly and non-scalable when adapting to other domains (e.g. biomedical domain). In addition, when the number of types of interest becomes large, the generation of handcrafted training data can be error-prone. To alleviate such an exhaustive process, the recent trend has deviated towards the adoption of distant supervision (DS). DS replaces the manual training data generation with a pipeline that automatically links texts to a knowledge base (KB). The pipeline has the following steps: (1) detect entity mentions in text; (2) map detected entity mentions to entities in KB; (3) assign, to the candidate type set of each entity mention pair, all KB relation types between their KB-mapped entities. However, the noise introduced to the automatically generated training data is not negligible. There are two major causes of error: incomplete KB and context-agnostic labeling process. If we treat unlinkable entity pairs as the pool of negative examples, false negatives can be commonly encountered as a result of the insufficiency of facts in KBs, where many true entity or relation mentions fail to be linked to KBs (see example in Figure 1.1). In this way, models counting on extensive negative instances may suffer from such misleading training data. On the other hand, context-agnostic labeling can engender false positive examples, due to the inaccuracy of the DS assumption that if a sentence contains any two entities holding a relation in the KB, the sentence must be expressing such relation between them. For example, entities "*Donald Trump*" and "*United States*" in the sentence "*Donald Trump flew back to United States*" can be labeled as "`president_of`" as well as "`born_in`", although only an out-of-interest relation type "`travel_to`" is expressed explicitly (as shown in Figure 1.1).

Towards the goal of diminishing the negative effects by noisy DS training data, distantly supervised RE models that deal with training noise, as well as methods that directly improve the automatic training data generation process have been proposed. These methods mostly involve designing distinct assumptions to remove redundant training information [1, 2, 3, 4]. For example, method applied in [2, 3] assumes that for each relation triple in the KB,



ID	Sentence
S1	***Donald Trump*** is the 45th and current President of the ***United States***.
S2	***Donald Trump*** is a citizen of the New York City, ***USA***.
S3	***Trump*** traveled on his private jet from UK back to the ***US***.
S4	***Ellen***, a native of ***China***, went to the United States four years ago.
…	…

Q1: What is *Jack*'s nationality?	
A1: ***Jack*** is a citizen of ***Germany***.	+
A2: ***Jack***, a native of ***Germany***, like beer.	+
A3: ***Jack*** just boarded on a flight to ***France***.	-

Relation Type	Entity 1	Entity 2
president_of	*Donald Trump*	*United States*
citizen_of	*Donald Trump*	*United States*

Automatically Labeled Training Data

Relation Mention: (*"Donald Trump"*, *"United States"*, S1)
Relation Types: {president_of, citizen_of}

Relation Mention: (*"Donald Trump"*, *"USA"*, S2)
Relation Types: {president_of, citizen_of}

Relation Mention: (*"Trump"*, *"US"*, S3)
Relation Types: {president_of, citizen_of}

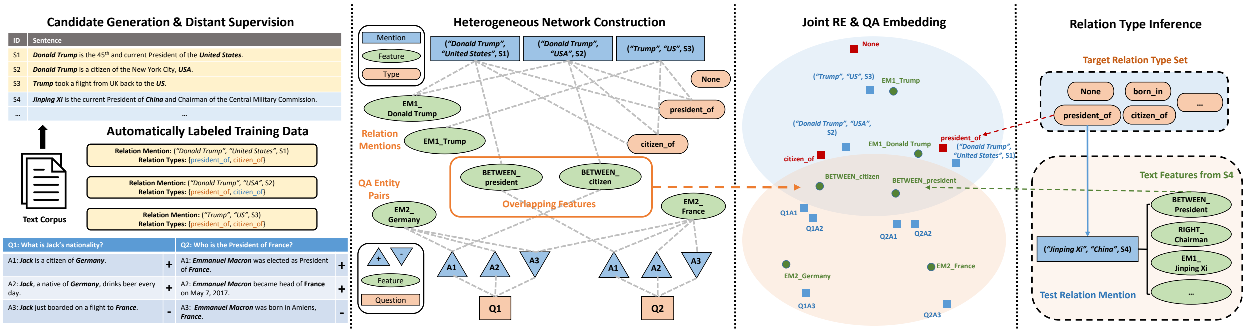
Relation Mention: (*"Ellen"*, *"China"*, S4)
Relation Types: {None}

Figure 1.1: Motivations of REQUEST.

at least one sentence might express the relation instead of all sentences. Moreover, these noise reduction systems usually only address one type of error, either false positives or false negatives. Hence, current methods handling DS noises still have the following challenges:

1. Lack of trustworthy sources: Current de-noising methods mainly focus on recognizing labeling mistakes from the labeled data itself, assisted by pre-defined assumptions or patterns. They do not have external trustworthy sources as guidance to uncover incorrectly labeled data, while not at the expense of excessive human efforts. Without other separate information sources, the reliability of false label identification can be limited.
2. Incomplete noise handling: Although both false negative and false positive errors are observed to be significant, most existing works only address one of them.

In this paper, to overcome the above two issues derived from relation extraction with distant supervision, we study the problem of relation extraction with indirect supervision from external sources. Recently, the rapid emergence of QA systems promotes the availability of user feedback or datasets of various QA tasks. We investigate to leverage QA, a downstream application of relation extraction, to provide additional signals for learning RE models. Specifically, we use datasets for the task of answer sentence selection to facilitate relation typing. Given a domain-specific corpus and a set of target relation types from a KB, we aim to detect relation mentions from text and categorize each in context by target types or Non-Target-Type (`None`) by leveraging an independent dataset of QA pairs in the



Candidate Generation & Distant Supervision

ID	Sentence
S1	***Donald Trump*** is the 45th and current President of the ***United States***.
S2	***Donald Trump*** is a citizen of the New York City, ***USA***.
S3	***Trump*** took a flight from UK back to the ***US***.
S4	***Jinping Xi*** is the current President of ***China*** and Chairman of the Central Military Commission.
...	...

Automatically Labeled Training Data

Relation Mention: (*"Donald Trump"*, *"United States"*, S1)
Relation Types: {president_of, citizen_of}

Relation Mention: (*"Donald Trump"*, *"USA"*, S2)
Relation Types: {president_of, citizen_of}

Relation Mention: (*"Trump"*, *"US"*, S3)
Relation Types: {president_of, citizen_of}

Q1: What is Jack's nationality?		Q2: Who is the President of France?	
A1: ***Jack*** is a citizen of ***Germany***.	+	A1: ***Emmanuel Macron*** was elected as President of ***France***.	+
A2: ***Jack***, a native of ***Germany***, drinks beer every day.	+	A2: ***Emmanuel Macron*** became head of **France** on May 7, 2017.	+
A3: ***Jack*** just boarded on a flight to ***France***.	-	A3: ***Emmanuel Macron*** was born in Amiens, ***France***.	-

Figure 1.2: Overall Framework.

same domain. We address the above two challenges as follows: (1) We integrate indirect supervision from another same-domain data source in the format of QA sentence pairs, that is, each question sentence maps to several positive (where a true answer can be found) and negative (where no answer exists) answer sentences. We adopt the principle that for the same question, positive pairs of (question, answer) should be semantically similar while they should be dissimilar from negative pairs. (2) Instead of differentiating types of labeling errors at the instance level, we concentrate on how to better learn semantic representation of features. Wrongly labeled training examples essentially misguide the understanding of features. It increases the risk of having a non-representative feature learned to be close to a relation type and vice versa. Therefore, if the feature learning process is improved, potentially both types of error can be reduced. (See how QA pairs improve the feature embedding learning process in Figure 2.1).

To integrate all the above elements, a novel framework, ReQuest, is proposed. First, ReQuest constructs a heterogeneous graph to represent three kinds of objects: relation mentions, text features and relation types for RE training data labeled by KB linking. Then, ReQuest constructs a second heterogeneous graph to represent entity mention pairs (include question, answer entity mention pairs) and features for QA dataset. These two graphs are combined into a single graph by overlapped features. We formulate a global objective to jointly embed the graph into a low-dimensional space where, in that space, RE objects whose types are semantically close also have similar representations and QA objects linked by positive (question, answer) entity mention pairs of a same question should have close representations. In particular, we design a novel margin-based loss to model the semantic similarity between QA pairs and transmit such information into feature and relation type representations via shared features. With the learned embeddings, we can efficiently estimate the types for test relation mentions. In summary, this paper makes the following contributions:

1. We propose the novel idea of applying indirect supervision from question answering datasets to help eliminate noise from distant supervision for the task of relation extraction.
2. We design a novel joint optimization framework, ReQuest, to extract typed relations in domain-specific corpora.
3. Experiments with two public RE datasets combined with TREC QA demonstrate that ReQuest improves the performance of state-of-the-art RE systems significantly.

CHAPTER 2: DEFINITIONS AND PROBLEM

Our proposed ReQuest framework takes the following input: an automatically labeled training corpus $\mathcal{D}_L$ obtained by linking a text corpus $\mathcal{D}$ to a KB (e.g. Freebase) Ψ, a target relation type set $\mathcal{R}$ and a set of QA sentence pairs $\mathcal{D}_{QAS}$ with extract answers labeled.

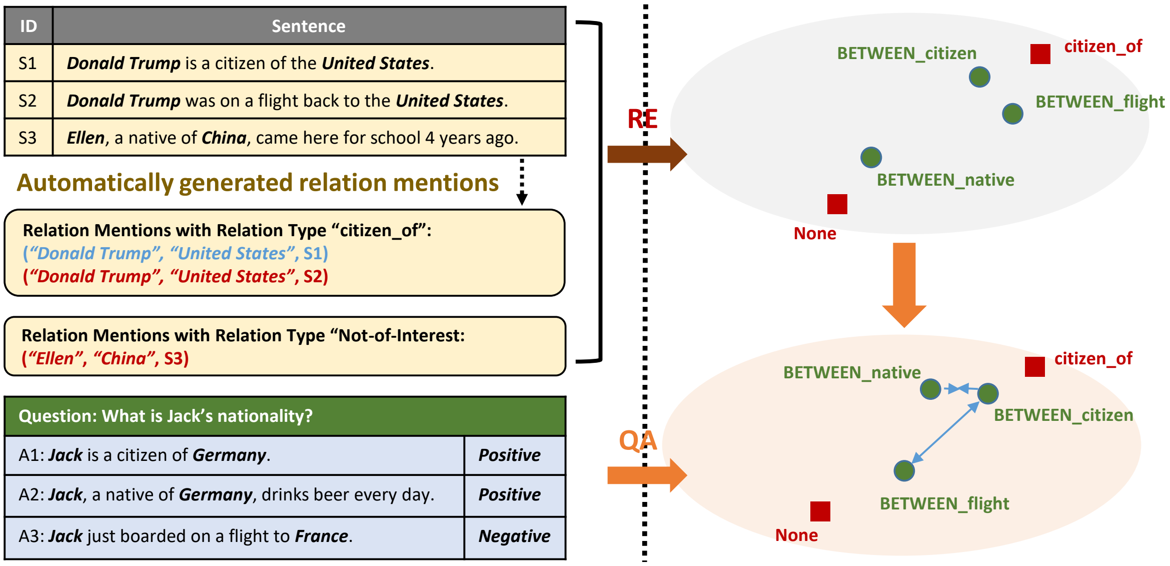
Entity and Relation Mention. An *entity mention* (denoted by m) is a token span in text which represents an entity e. A *relation instance* $r(e_1, e_2, \ldots, e_n)$ denotes some type of relation $r \in \mathcal{R}$ between multiple entities. In this paper, we focus on binary relations, *i.e.*, $r(e_1, e_2)$. We define a *relation mention* (denoted by z) for some relation instance $r(e_1, e_2)$ as a (ordered) pair of entities mentions of e_1 and e_2 in a sentence s, and represent a relation mention with entity mentions m_1 and m_2 in sentence s as $z = (m_1, m_2, s)$.

Knowledge Bases and Target Types. A KB contains a set of entities $\mathcal{E}_\Psi$, entity types $\mathcal{Y}$ and relation types $\mathcal{R}$, as well as human-curated facts on both relation instances $\mathcal{I}_\Psi = \{r(e_1, e_2)\} \subset \mathcal{R}_\Psi \times \mathcal{E}_\Psi \times \mathcal{E}_\Psi$, and entity-type facts $\mathcal{T}_\Psi = \{(e, y)\} \subset \mathcal{E}_\Psi \times \mathcal{Y}_\Psi$. *Target relation type set* $\mathcal{R}$ covers a subset of relation types that the users are interested in from Ψ, *i.e.*, $\mathcal{R} \subset \mathcal{R}_\Psi$.

Automatically Labeled Training Corpora. Distant supervision maps the set of entity mentions extracted from the text corpus to KB entities $\mathcal{E}_\Psi$ with an entity disambiguation system [5, 6]. Between any two linkable entity mentions m_1 and m_2 in a sentence, a relation mention z_i is formed if there exists one or more KB relations between their KB-mapped entities e_1 and e_2. Relations between e_1 and e_2 in KB are then associated to z_i to form its candidate relation type set $\mathcal{R}_i$, *i.e.*, $\mathcal{R}_i = \{r \mid r(e_1, e_2) \in \mathcal{R}_\Psi\}$.

Let $\mathcal{Z} = \{z_i\}_{i=1}^{N_Z}$ denote the set of extracted relation mentions that can be mapped to KB. Formally, we represent the automatically labeled training corpus $\mathcal{D}_L$ for relation extraction, using a set of tuples $\mathcal{D}_L = \{(z_i, \mathcal{R}_i)\}_{i=1}^{N_Z}$. There exists publicly available automatically labeled corpora such as the NYT dataset [2] where relation mentions have already been extracted and mapped to KB.

QA Entity Mention Pairs. The set of QA sentence pairs $\mathcal{D}_{QAS}$ consists of questions $\mathcal{Q}$ in the same domain as the training text corpus. For each question q_i, there will be a number of positive sentences $\mathcal{A}_i^+$, each of which contains a correct answer to the question and another set of negative sentences $\mathcal{A}_i^-$ where no answer can be found. And the tokens spans of the exact answer in each positive is marked as well. For each question, we extract positive QA (ordered) entity mention pairs $\mathcal{P}_i^+$ from $\mathcal{A}_i^+$ and negative entity mention pairs $\mathcal{P}_i^-$ from $\mathcal{A}_i^-$. A positive QA entity mention pair p_k contains an entity mention being asked about (question entity mention m_1) and an entity mention serving as the answer (answer entity mention m_2)



ID	Sentence
S1	***Donald Trump*** is a citizen of the ***United States***.
S2	***Donald Trump*** was on a flight back to the ***United States***.
S3	***Ellen***, a native of ***China***, came here for school 4 years ago.

Automatically generated relation mentions

Relation Mentions with Relation Type "citizen_of":
("Donald Trump", "United States", S1)
("Donald Trump", "United States", S2)

Relation Mentions with Relation Type "Not-of-Interest:
("Ellen", "China", S3)

Question: What is Jack's nationality?	
A1: ***Jack*** is a citizen of ***Germany***.	***Positive***
A2: ***Jack***, a native of ***Germany***, drinks beer every day.	***Positive***
A3: ***Jack*** just boarded on a flight to ***France***.	***Negative***

Figure 2.1: Indirect Supervision From QA Pairs.

to a question. That being said, we can get one positive QA entity mention pair from each positive answer sentence if both entity mentions can be found. In contrast, A negative QA entity mention pair does not follow such pattern for the corresponding question.

Let $\mathcal{Q} = \{q_i\}_{i=1}^{N_q}$ denote the set of questions; $\mathcal{P} = \{p_k\}_{k=1}^{N_p}$ denote all QA entity mention pairs; $\mathcal{P}_i^+ = \{p_{k^+}\}_{k^+=1}^{N_i^+}$ denote the set of positive QA entity mention pairs for q_i; $\mathcal{P}_i^- = \{p_{k^-}\}_{k^-=1}^{N_i^-}$ denote the set of negative QA entity mention pairs for q_i. Formally, the QA entity mention pairs corpus is represented as $\mathcal{D}_{QA} = \{(q_i, \mathcal{P}_i^+, \mathcal{P}_i^-)\}_{i=1}^{N_q}$.

Definition 2.1 (Problem Definition) ***Given*** *an automatically generated training corpus $\mathcal{D}_L$, a target relation type set $\mathcal{R} \subset \mathcal{R}_\Psi$ and a set of QA sentence pairs $\mathcal{D}_{QAS}$ in the same domain, the relation extraction task* ***aims to*** *(1) extract QA entity mention pairs to generate $\mathcal{D}_{QA}$; (2) estimate a relation type $r^* \in \mathcal{R} \cup \{\texttt{None}\}$ for each test relation mention, using both the training corpus and the extracted QA pairs with their contexts.*

CHAPTER 3: APPROACH

Framework Overview. We propose an *embedding-based* framework with indirect supervision (illustrated in Figure 1.2) as follows:

1. Generate text features for each relation mention or QA entity mention pair, and construct a heterogeneous graph using four kinds of objects in combined corpus, namely relation mentions from RE corpus, entity mention pairs from QA corpus, target relation types and text features to encode aforementioned signals in a unified form (Section 3.1).
2. Jointly embed relation mentions, QA pairs, text features, and type labels into two low-dimensional spaces connected by shared features, where close objects tend to share the same types or questions (Section 3.2).
3. Estimate type labels r^* for each test relation mention z from learned embeddings, by searching the target type set $\mathcal{R}$ (Section 3.3).

3.1 HETEROGENEOUS NETWORK CONSTRUCTION

Relation Mentions and Types Generation. We get the relation mentions along with their heuristically obtained relation types from the automatically labeled training corpus $\mathcal{D}_L$. And we randomly sample a set of unlinkable entity mention pairs as the negative relation mentions (*i.e.*, relation mentions assigned with type "`None`").

QA Entity Mention Pairs Generation. We apply Stanford NER [7] to extract entity mentions in each question or answer sentence. First, we detect the target entity being asked about in each question sentence. For example, in the question "*Who is the president of United States*", the question entity is "*United States*". In most cases, a question only contains one entity mention and for those containing multiple entity mentions, we notice the question entity is mostly mentioned at the very last. Thus, we follow this heuristic rule to assign the lastly occurred entity mention to be the question entity mention m_0 in each question sentence q_i. Then, in each positive answer sentence of q_i, we extract the entity mention with matched head token and smallest edit string distance to be the question entity mention m_1, and the entity mention matching the exact answer string to be the answer entity mention m_2. Then we form a positive QA entity mention pair with its context s, $p_k = (m_1, m_2, s) \in \mathcal{P}_i^+$ for q_i. If either m_1 or m_2 can not be found, this positive answer sentence is dropped. We randomly select pairs of entity mentions in each negative answer sentence to be negative QA entity mention pairs for q_i (*e.g.*, if a negative sentence includes

Table 3.1 Text Features of Relation Mentions.

Feature	Description	Example
EM head	Syntactic head token of each entity mention	“*HEAD_EM1_Trump*”
EM Token	Tokens in each entity mention	“*TKN_EM1_Donald*”
Tokens	Each token between two EMs	“*is*”, “*the*”, “*current*”, “*President*”, “*of*”, “*the*”
POS tag	POS tags of tokens between two EMs	“*VBZ*”, “*DT*”, “*JJ*”, “*NN*”, “*IN*”, “*DT*”
Collocations	Bigrams in 3-word window of each EM	“*NYC native*”, “*native Donald*”, ...
EM order	Whether EM 1 is before EM 2	“*EM1_BEFORE_EM2*”
EM distance	Number of tokens between the two EMs	“*EM_DISTANCE_6*”
EM context	Unigrams before and after each EM	“*native*”, “*is*”, “*the*”, “*.*”
Special pattern	Occurrence of pattern “em1_in_em2”	“*PATTERN_NULL*”
Brown cluster	Brown cluster ID for each token	“*8_1101111*”, “*12_111011111111*”

3 entity mentions, we randomly select negative examples from the $3 \cdot 2 \cdot 1 = 6$ different pairs of entity mentions in total, if we ignore the order), with each negative example marked as $p_k\prime = (m_1\prime, m_2\prime, s\prime) \in \mathcal{P}_i^-$ for q_i.

Text Feature Extraction. We extract lexical features of various types from not only the mention itself (*e.g.*, head token), as well as the context s (*e.g.*, bigram) in a POS-tagged corpus. It is to capture the syntactic and semantic information for any given relation mentions or entity mention pairs. See Table 3.1 for all types of text features used for the example relation mention (“*Donald Trump*”, “*United States*”) from the sentence “*NYC native* ***Donald Trump*** *is the current President of the* ***United States***.”, following those in [1, 8] (excluding the dependency parse-based features and entity type features).

We denote the set of M_z unique features extracted from relation mentions $\mathcal{Z}$ as $\mathcal{F}_z = \{f_j\}_{j=1}^{M_z}$ and the set of M_{QA} unique features extracted of QA entity mention pairs $\mathcal{P}$ as $\mathcal{F}_{QA} = \{f_j\}_{j=1}^{M_{QA}}$. As our embedding learning process will combine these two sets of features and their shared ones will act as the bridge of two embedding spaces, we denote the overall feature set as $\mathcal{F} = \{f_j\}_{j=1}^{M}$.

Heterogeneous Network Construction. After the nodes generation process, we construct a heterogeneous network connected by text features, relation mentions, relation types, questions, QA entity mention pairs, as shown in the second column of Figure 1.2.

3.2 JOINT RE AND QA EMBEDDING

This section first introduces how we model different types of interactions between linkable relation mentions $\mathcal{Z}$, QA entity mention pairs $\mathcal{P}$, relation type labels $\mathcal{R}$ and text features $\mathcal{F}$ into a d-dimensional *relation vector space* and a d-dimensional *QA pair vector space.* In the relation vector space, objects whose types are close to each other should have similar representation and in the QA pair vector space, positive QA mention pairs who share the same question are close to each other. (*e.g.*, see the 3rd col. in Figure 1.2). We then combine multiple objectives and formulate a joint optimization problem.

We propose a novel global objective, which employs a margin-based rank loss [9] to model *noisy mention-type associations* and utilizes the second-order proximity idea [10] to model *mention-feature (QA pair-feature) co-occurrences.* In particular, we adopt a pairwise margin loss, following the intuition of pairwise rank [11] to capture the *interactions between QA pairs*, and the shared features $\mathcal{F}_z \cap \mathcal{F}_{QA}$ between relation mentions $\mathcal{Z}$ and QA pairs $\mathcal{P}$ connect the two vector spaces.

Modeling Types of Relation Mentions. We introduce the concepts of both *mention-feature co-occurrences* and *mention-type associations* in the modeling of relation types for relation mentions in set Z.

The first hypothesis involved in modeling types of relation mentions is as follows.

Hypothesis 3.1 (Mention-Feature Co-occurrence) *If two relation mentions share many text features, they tend to share similar types (close to each other in the embedding space). If two features co-occur with a similar set of relation mentions, they tend to have similar embedding vectors.*

This is based on the intuition that if two relation mentions share many text features, they have high distributional similarity over the set of text features $\mathcal{F}_z$ and likely they have similar relation types. On the other hand, if text features co-occur with many relation mentions in the corpus, such features tend to represent close type semantics. For example, in sentences s_1 and s_4 in the first column of Figure 1.2, the two relation mentions ("*Donald Trump*", "*United States*", s_1) and ("*Jinping Xi*", "*China*", s_4) share many text features including "*BETWEEN_President*" and they indeed have the same relation type "`president_of`"

Formally, let vectors $\mathbf{z}_i,\ \mathbf{c}_j \in \mathbb{R}^d$ represent relation mention $z_i \in \mathcal{Z}$ and text feature $f_j \in \mathcal{F}_z$ in the d-dimensional *relation embedding space.* Similar to the distributional hypothesis [12] in text corpora, we apply second-order proximity [10] to model the idea in Hypothesis 1 as follows.

$$\mathcal{L}_{ZF} = -\sum_{z_i \in \mathcal{Z}} \sum_{f_j \in \mathcal{F}_z} w_{ij} \cdot \log p(f_j | z_i), \tag{3.1}$$

where $p(f_j|z_i) = \exp(\mathbf{z}_i^T \mathbf{c}_j) / \sum_{f' \in \mathcal{F}_z} \exp(\mathbf{z}_i^T \mathbf{c}_{j'})$ 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 $\mathcal{D}$.

For the goal of efficient optimization, we apply negative sampling strategy [12] to sample multiple *false* features for each (z_i, f_j) based on some *noise distribution* $P_n(f) \propto D_f^{3/4}$ [12] (with D_f denotes the number of relation mentions co-occurring with f). Term $\log p(f_j|z_i)$ in

Eq. (3.1) is replaced with the term as follows.

$$\log\,\sigma(\mathbf{z}_i^T\mathbf{c}_j) + \sum_{v=1}^{V} \mathbb{E}_{f_{j'}\sim P_n(f)}\left[\log\,\sigma(-\mathbf{z}_i^T\mathbf{c}_{j'})\right], \tag{3.2}$$

where $\sigma(x) = 1/\big(1+\exp(-x)\big)$ is the sigmoid function. The first term in Eq. (3.2) models the observed co-occurrence, and the second term models the V negative feature samples.

In D_L, each relation mention z_i is associated with a set of candidate types $\mathcal{R}_i$ in a context-agnostic setting, which leads to some false associations between z_i and $r \in \mathcal{R}_i$ (*i.e.*, false positives). For example, in the first column of Figure 1.2, the two relation mentions ("*Donald Trump*", "*United States*", s_1) and ("*Donald Trump*", "*USA*", s_2) are assigned to the same relation types while each mention actually only has one true type. To handle such conflicts, we use the following hypothesis to model the associations between each linkable relation mention z_i (in set $\mathcal{Z}$) and its noisy candidate relation type set $\mathcal{R}_i$.

Hypothesis 3.2 (Partial-Label Association) *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.*

Let vector $\mathbf{r}_k \in \mathbb{R}^d$ denote relation type $r_k \in \mathcal{R}$ in the embedding space, the similarity between (z_i, r_k) is defined as the dot product of their embedding vectors, *i.e.*, $\phi(z_i, r_k) = \mathbf{z}_i^T\mathbf{r}_k$. $\overline{\mathcal{R}}_i = \mathcal{R} \setminus \mathcal{R}_i$ denotes the set of *non-candidate types.* We extend the margin-based loss in [9] to define a partial-label loss ℓ_i for each linkable relation mention $z_i \in \mathcal{M}_L$ as follows.

$$\ell_i = \max\Big\{0, 1 - \Big[\max_{r\in\mathcal{R}_i}\phi(z_i, r) - \max_{r'\in\overline{\mathcal{R}}_i}\phi(z_i, r')\Big]\Big\}. \tag{3.3}$$

To comprehensively model the types of relation mentions, we integrate the modeling of mention-feature co-occurrences and mention-type associations by the following objective, so that feature embeddings also participate in modeling the relation type embeddings.

$$O_Z = \mathcal{L}_{ZF} + \sum_{i=1}^{N_Z}\ell_i + \frac{\lambda}{2}\sum_{i=1}^{N_Z}\|\mathbf{z}_i\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K_r}\|\mathbf{r}_k\|_2^2, \tag{3.4}$$

where tuning parameter $\lambda > 0$ on the regularization terms is used to control the scale of the embedding vectors.

Modeling Associations between QA Entity Mention Pairs. We follow Hypothesis 1 to model the QA pair-feature co-occurrence in a similar way. Formally, let vectors $\mathbf{p}_i, \mathbf{c}'_j \in \mathbb{R}^d$ represent QA entity mention pair $p_i \in \mathcal{P}$ and text features (for entity mentions) $f_j \in \mathcal{F}_{QA}$in a d-dimensional *QA entity pair embedding space*, respectively. We model the corpus-level co-occurrences between QA entity mention pairs and text features by second-order proximity as follows.

$$\mathcal{L}_{PF} = -\sum_{p_i \in \mathcal{P}} \sum_{f_j \in \mathcal{F}_{QA}} w_{ij} \cdot \log\, p(f_j|p_i), \tag{3.5}$$

where the term $\log\, p(f_j|p_i)$ is defined as

$$\log\, p(f_j|p_i) = \log\, \sigma(\mathbf{p}_i^T \mathbf{c}'_j) + \sum_{v=1}^{V} \mathbb{E}_{f_{j'} \sim P_n(f)} \big[\log\, \sigma(-\mathbf{p}_i^T \mathbf{c}'_{j'})\big]. \tag{3.6}$$

For each QA entity mention pair, if we consider it as a relation mention with an unknown type, intuitively, positive pairs sharing a same question are relation mentions with the same relation type or more specifically, are semantically similar relation mentions. In contrast, a positive pair and a negative pair for a question should be semantically far away from each other. For example, in Figure 2.1, the embeddings of the entity mention pair in answer sentence A_1 should be close to the pair in A_2 while far away from the pair in A_3. To impose such idea, we model the interactions between QA entity mention pairs based on the following hypothesis.

Hypothesis 3.3 (QA Pairwise Interaction) *A positive QA entity mention pair's embedding vector should be more similar (closer in the low-dimensional space) to any other positive QA entity mention pair, than to any negative QA entity mention pair of the same question.*

Specifically, we use vector $\mathbf{p}_k \in \mathbb{R}^d$ to represent a positive QA entity mention pair p_k in the embedding space. The similarity between two QA entity mention pairs p_{k1} and p_{k2} is defined as the dot product of their embedding vectors. For a positive QA entity mention pair p_k of a question q_i (e.g. $p_k \in \mathcal{P}_i^+$), we define the pairwise margin-based loss as follows.

$$\ell_{i,k} = \sum_{p_{k_1} \in \mathcal{P}_i^+, p_{k_2} \in \mathcal{P}_i^-, k_1 \neq k} \max\Big\{0, 1 - \Big[\phi(p_k, p_{k_1}) - \phi(p_k, p_{k_2})\Big]\Big\}. \tag{3.7}$$

To integrate both the modeling of QA pair-feature co-occurrence and QA pairs interaction, we formulate the following objective.

$$O_{QA} = \mathcal{L}_{PF} + \sum_{i=1}^{N_Q} \sum_{k=1}^{N_i^+} \ell_{i,k} + \frac{\lambda}{2} \sum_{k=1}^{N_P} \|\mathbf{p}_k\|_2^2. \tag{3.8}$$

Algorithm 3.1 Model Learning of ReQuest

```
Input: labeled training corpus $\mathcal{D}_L$, text features $\{\mathcal{F}\}$, regularization parameter $\lambda$, learning rate $\alpha$, number of negative samples $V$, dim. $d$
Output: relation mention/QA entity mention pair embeddings $\{\mathbf{z}_i\}/\{\mathbf{p}_k\}$, feature embeddings $\{\mathbf{c}_j\}$, $\{\mathbf{c}'_j\}$, relation type embedding $\{\mathbf{r}_k\}$
Initialize: vectors $\{\mathbf{z}_i\}$,$\{\mathbf{p}_k\}$,$\{\mathbf{c}_j\}$,$\{\mathbf{c}'_j\}$,$\{\mathbf{r}_k\}$ as random vectors
while $\mathcal{O}$ in Eq. (3.9) not converge do
    Sample one component $O_{cur}$ from $\{O_Z,\ O_{QA}\}$
    if $O_{cur}$ is $O_Z$ then
        Sample a mention-feature co-occurrence $w_{ij}$; draw $V$ negative samples; update $\{\mathbf{z},\ \mathbf{c}\}$ based on $\mathcal{L}_{ZF}$
        Sample a relation mention $z_i$; get its candidate types $\mathcal{R}_i$; update $\mathbf{z}$ and $\{\mathbf{r}\}$ based on $\mathcal{O}_Z - \mathcal{L}_{ZF}$
    end
    if $O_{cur}$ is $O_{QA}$ then
        Sample a pair-feature co-occurrence $w_{ij}$; draw $V$ negative samples; update $\{\mathbf{p},\ \mathbf{c}'\}$ based on $\mathcal{L}_{PF}$
        Sample an positive QA entity mention pair $p_k$ of question $q_i$; sample one more positive pair and one negative pair of question $q_i$; update $\mathbf{p}$ based on $\mathcal{O}_{QA} - \mathcal{L}_{PF}$
    end
end
```

By doing so, we can extend the semantic relationships between QA pairs to feature embeddings, such that features of close QA pairs also have similar embeddings. Thus, the learned embeddings of text features from QA corpus carry semantic information inferred from QA pairs. The shared features can propagate such extra semantic knowledge into relation vector space and help better learn the semantic embeddings of both text features and relation types. While feature embeddings of both false positive or false negative examples in the training corpus can deviate towards unrepresentative relation types, the transmitted knowledge from QA space has the potential to adjust such semantic inconsistency. For example, as illustrated in Figure 2.1, the false labeled examples in s_2 and s_3 lead the features "*BETWEEN_flight*" and "*BETWEEN_native*" to be close to "`citizen_of`" and "`None`" type respectively. After injecting the QA pairwise interactions from the example question, these wrongly placed features are brought back towards the relation types they actually indicate. Minimizing the objective O_{QA} yields an QA pair embedding space where, in that space, positive QA mention pairs who share the same question are close to each other.

A Joint Optimization Problem. Our goal is to embed all the available information for relation mentions and relation types, QA entity mention pairs and text features into a single d-dimensional embedding space. An intuitive solution is to collectively minimize the two objectives O_Z and O_{QA} as the embedding vectors of overlapped text features are shared across relation vector space and QA pair vector space. To achieve the goal, we formulate a joint optimization problem as follows.

$$\min_{\{\mathbf{z}_i\},\{\mathbf{c}_j\},\{\mathbf{r}_k\},\{\mathbf{p}_k\},\{\mathbf{c}'_j\}} \mathcal{O} = \mathcal{O}_Z + \mathcal{O}_{QA}. \tag{3.9}$$

When optimizing the global objective O, the learning of RE and QA embeddings can be mutually influenced as errors in each component can be constrained and corrected by the other. This mutual enhancement also helps better learn the semantic relations between features and relation types. We apply edge sampling strategy [10] with a stochastic sub-gradient descent algorithm [13] to efficiently solve Eq. (3.9). In each iteration, we alternatively sample from each of the two objectives $\{O_Z, O_M\}$ a batch of edges (*e.g.*, (z_i, f_j)) and their negative samples, and update each embedding vector based on the derivatives. The detailed learning process of ReQuest can be seen in Algorithm 3.1. To prove convergence of this algorithm (to the local minimum), we can adopt the proof procedure in [13].

3.3 TYPE INFERENCE

To predict the type for each test relation mention z, we search for nearest neighbor in the target relation type set $\mathcal{R}$, with the learned embeddings of features and relation types (*i.e.*, $\{\mathbf{c}_i\}$, $\{\mathbf{c}'_i\}$, $\{\mathbf{r}_k\}$). Specifically, we represent test relation mention z in our learned relation embedding space by $\mathbf{z} = \sum_{f_j \in \mathcal{F}_z(z)} \mathbf{c}_j$ where $\mathcal{F}_z(z)$ is the set of text features extracted from z's local context s. We categorize z to `None` type if the similarity score is below a pre-defined threshold (e.g. $\eta > 0$).

CHAPTER 4: EXPERIMENTS

4.1 DATA PREPARATION AND EXPERIMENT SETTING

Our experiments consists of two different type of datasets, one for relation extraction and another answer sentence selection dataset for indirect supervision. Two public datasets are used for relation extraction: **NYT** [2, 3]and **KBP** [14, 15]. The test data are manually annotated with relation types by their respective authors. Statistics of the datasets are shown in Table 4.1. Automatically generated training data by distant supervision on these two training corpora have been used in [16, 2] and is accessible via public links, as well as the test data[1]. The automatic data generation process is the same as described in Section 2 by utilizing DBpedia Spotlight[2], a state-of-the-art entity disambiguation tool, and Freebase, a large entity knowledge base. As for QA dataset, we use the answer sentence selection dataset extracted from **TREC-QA** dataset [17] used by many researchers [18, 19, 20]. We obtain the compiled version of the dataset from [21, 22], which can be accessed via publicly available link[3]. Then, we parse this QA dataset to generate QA entity mention pairs following the steps described in Section 3.1. During this procedure, we drop the question or answer sentences where no valid QA entity mention pairs can be found. The statistics of this dataset is presented in Table 4.2.

Feature Generation. This step is run on both relation extraction dataset and preprocessed QA entity mention pairs and sentences. Table 3.1 lists the set of text features of both relation mentions and QA entity mention pairs used in our experiments. We use a 6-word window to extract context features for each mention (3 words on the left and the right). We apply the Stanford CoreNLP tool [7] to get POS tags. Brown clusters are derived for each corpus using public implementation[4]. The same kinds of features are used in all the compared methods in our experiments. As the overlapped features in both RE and QA datasets play an important role in the optimization process, we put the statistics of the shared features in Table 4.3.

Evaluation Sets. The provided train/test split are used in NYT and KBP relation extraction datasets. The relation mentions in test data have been manually annotated with relation types in the released dataset (see Table 4.1 for the data statistics). A *validation set* is created through randomly sampling 10% of relation mentions from test data, and the rest are used as *evaluation set.*

[1] `https://github.com/shanzhenren/CoType/tree/master/data/source`

[2] `http://spotlight.dbpedia.org/`

[3] `https://github.com/xuchen/jacana/tree/master/tree-edit-data`

[4] `https://github.com/percyliang/brown-cluster`

Table 4.1 Statistics of Relation Extraction Datasets.

Data sets	NYT	KBP
#Relation types	24	19
#Documents	294,977	780,549
#Sentences	1.18M	1.51M
#Training RMs	353k	148k
#Text features	2.6M	1.3M
#Test Sentences	395	289
#Ground-truth RMs	3,880	2,209

Table 4.2 Statistics of the Answer Sentence Selection Datasets.

Versions of QA dataset	COMPLETE	FILTERED
#Questions	1.4K	186
#Positive Answer Sentences	6.9K	969
#Negative Answer Sentences	49K	5.5K
#Positive entity mention pairs	-	969
#Negative entity mention pairs	-	28K

Compared Methods. We compare ReQuest with its variants which model parts of the proposed hypotheses. Several state-of-the-art relation extraction methods (*e.g.*, supervised, embedding, neural network) are also implemented (or tested using their published codes): (1) **DS+Perceptron** [14]: adopts multi-label learning on automatically labeled training data $\mathcal{D}_L$. (2) **DS+Kernel** [23]: applies bag-of-feature kernel [23] to train a SVM classifier using $\mathcal{D}_L$; (3) **DS+Logistic** [1]: trains a multi-class logistic classifier[5] on $\mathcal{D}_L$; (4) **DeepWalk** [24]: embeds mention-feature co-occurrences and mention-type associations as a homogeneous network (with binary edges); (5) **LINE** [10]: uses second-order proximity model with edge sampling on a feature-type bipartite graph (where edge weight w_{jk} is the number of relation mentions having feature f_j and type r_k); (6) **MultiR** [3]: is a state-of-the-art distant supervision method, which models noisy label in $\mathcal{D}_L$ by multi-instance multi-label learning; (7) **FCM** [25]: adopts neural language model to perform compositional embedding; (8) **DS+SDP-LSTM** [26, 27]: current state-of-the-art in SemEval 2010 Task 8 relation classification task [28], leverages a multi-channel input along the shortest dependency path between two entities into stacked deep recurrent neural network model. We use $\mathcal{D}_L$ to train the model. (9) **DS+LSTM-ER** [29]: current state-of-the-art model on ACE2005 and ACE2004 relation classification task [30, 31]. It is a multi-layer LSTM-RNN based model that captures both word sequence and dependency tree substructure information. We use $\mathcal{D}_L$ to train the model. (10) **CoType-RM** [16]: A distant supervised model which adopts the partial-label loss to handle label noise and train the relation extractor.

Besides the proposed joint optimization model, **ReQuest-Joint**, we conduct experiments on two other variations to compare the performance (1) **ReQuest-QA_RE**: This variation optimizes objective $\mathcal{O}_{QA}$ first and then uses the learned feature embeddings as the initial

[5] We use liblinear package from `https://github.com/cjlin1/liblinear`

Table 4.3 Statistics of Overlapped Features.

Data sets	**NYT**	**KBP**
% distinct shared features with TREC QA	10.0%	11.6%
% occurrences of shared features with TREC QA	90.1%	85.6%

Table 4.4 Case Study.

Relation Mention	**ReQuest**	**CoType-RM**
.. traveling to *Amman* **,** *Jordan* ..	/location/location/contains	`None`
The photograph showed **Gov.** *Ernie Fletcher* **of** *Kentucky* ..	/people/person/place_lived	`None`
.. **as chairman of** the *Securities and Exchange Commission* , *Christopher Cox* ..	/business/person/company	`None`

state to optimize $\mathcal{O}_Z$; and (2) **ReQuest-RE_QA**: It first optimizes $\mathcal{O}_Z$, then optimizes $\mathcal{O}_{QA}$ to finely tune the learned feature embeddings.

Parameter Settings. In the testing of ReQuest and its variants, we set $\eta = 0.35$ and $\lambda = 10^{-4}$ and $V = 3$ based on validation sets. We stop further optimization if the relative change of $\mathcal{O}$ in Eq. (3.9) is smaller than 10^{-4}. The dimensionality of embeddings d is set to 50 for all embedding methods. For other parameters, we tune them on validation sets and picked the values which lead to the best performance.

Evaluation Metrics. We adopt standard Precision, Recall and F1 score [23, 32] for measuring the performance of relation extraction task. Note that all our evaluations are *sentence-level* or *mention-level* (*i.e.*, context-dependent), as discussed in [3].

4.2 EXPERIMENTS AND PERFORMANCE STUDY

Table 4.5 Performance Comparison on End-to-End Relation Extraction.

	NYT [2, 3]				**KBP** [15, 14]			
Method	**Prec**	**Rec**	**F1**	**Time**	**Prec**	**Rec**	**F1**	**Time**
DS+Perceptron [14]	0.068	**0.641**	0.123	15min	0.233	0.457	0.308	7.7min
DS+Kernel [23]	0.095	0.490	0.158	56hr	0.108	0.239	0.149	9.8hr
DS+Logistic [1]	0.258	0.393	0.311	25min	0.296	0.387	0.335	14min
DeepWalk [24]	0.176	0.224	0.197	1.1hr	0.101	0.296	0.150	27min
LINE [10]	0.335	0.329	0.332	2.3min	0.360	0.257	0.299	1.5min
MultiR [3]	0.338	0.327	0.333	5.8min	0.325	0.278	0.301	4.1min
FCM [25]	**0.553**	0.154	0.240	1.3hr	0.151	**0.500**	0.301	25min
DS+SDP-LSTM [26]	0.307	0.532	0.389	21hr	0.249	0.300	0.272	10hr
DS+LSTM-ER [29]	0.373	0.171	0.234	49hr	0.338	0.106	0.161	30hr
CoType-RM [16]	0.467	0.380	0.419	2.6min	0.342	0.339	0.340	1.5min
ReQuest-QA_RE	0.407	0.437	0.422	10.2min	**0.459**	0.300	0.363	5.3min
ReQuest-RE_QA	0.435	0.419	0.427	8.0min	0.356	0.352	0.354	13.2min
ReQuest-Joint	0.404	0.480	**0.439**	4.0min	0.386	0.410	**0.397**	5.9min

Performance Comparison with Baselines. To test the effectiveness of our proposed framework ReQuest, we compare with other methods on the relation extraction task. The precision, recall, F1 scores as well as the model learning time measured on two datasets are reported in Table 4.5. As shown in the table, ReQuest achieves superior F1 score on both datasets compared with other models. Among all these baselines, MultiR and CoType-RM handle noisy training data while the remaining ones assume the training corpus is perfectly labeled. Due to their nature of being cautious towards the noisy training data, both MultiR and CoType-RM reach relatively high results confronting with other models that blindly exploit all heuristically obtained training examples. However, as external reliable information sources are absent and only the noise from multi-label relation mentions (while none or only one assigned label is correct) is tackled in these models, MultiR and CoType-RM underperform ReQuest. Especially from the comparison with CoType-RM, which is also an embedding learning based relation extraction model with the idea of partial-label loss incorporated, we can conclude that the extra semantic inklings provided by the QA corpus do help boost the performance of relation extraction.

Performance Comparison with Ablations. We experiment with two variations of ReQuest, ReQuest-QA_RE and ReQuest-RE_QA, in order to validate the idea of joint optimization. As presented in Table 4.5, both ReQuest-QA_RE and ReQuest-RE_QA outperform most of the baselines, with the indirect supervision from QA corpus. However, their results still fall behind ReQuest's. Thus, separately training the two components may not capture as much information as jointly optimizing the combined objective. The idea of constraining each component in the joint optimization process proves to be effective in learning embeddings to present semantic meanings of objects (e.g. features, types and mentions).

4.3 CASE STUDY

Example Outputs. We have done some interesting investigations regarding the type of prediction errors that can be corrected by the indirection supervision from QA corpus. We have analyzed the prediction results on NYT dataset from CoType-RM and ReQuest and find out the top three target relation types that can be corrected by ReQuest are "`contains_location`", "`work_for`", "`place_lived`". Both the issues of KB incompleteness and context-agnostic labeling are severe for these relation types. For example, there can be lots of not that well-known suburban areas belonging to a city, a state or a country while not marked in KB. And a person can has lived in tens or even hundreds places for various lengths of period. These are hard to be fully annotated into a KB. Thus, the automatically obtained training corpus may end up containing a large percentage of false negative examples for such relation types. On the other hand, there are abundant entity pairs having both "`contains_location`" and "`capital_of`", or both "`place_lived`" and "`born_in`" relation types in KB. Naturally, training examples of such entity pairs can be greatly polluted by false

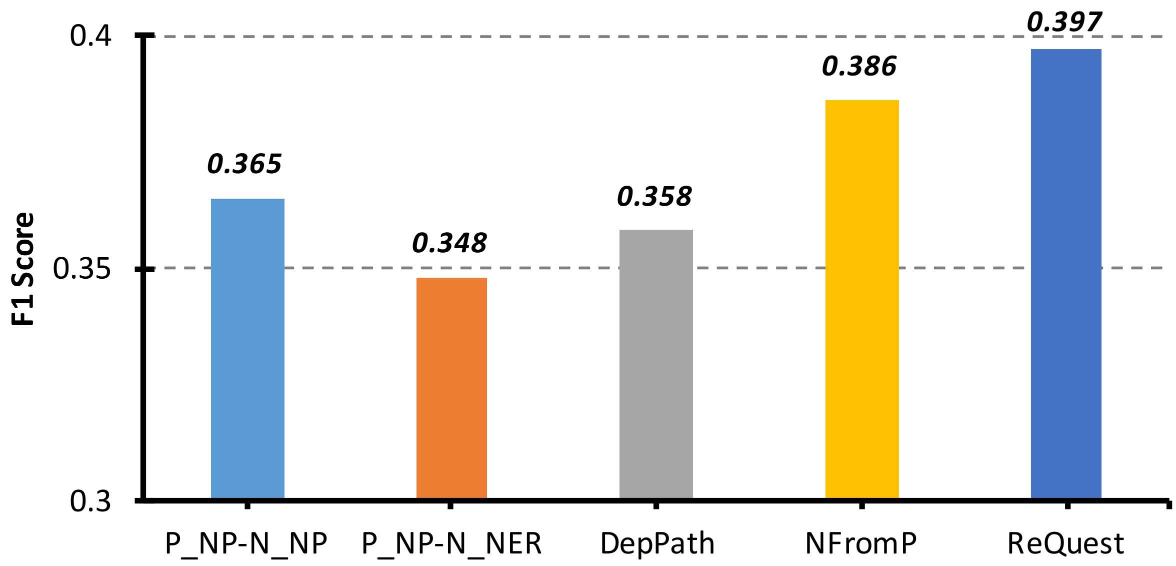


Figure 4.1: Effect of QA Dataset Processing on F1 scores.

positives. In this case, it becomes tough to learn semantic embeddings for relevant features of these relation types. However, we notice there are quite a few answer sentences for relevant questions like “Where is *XXX* located”, “Where did *XXX* live”, “What company is *XXX* with” in the QA corpus, which plays an important role in adjusting vectors for features that are supposed to be the indicators for these relation types. Table 4.4 shows some prediction errors from CoType-RM that are fixed in ReQuest.

Study the effect of QA dataset processing on F1 scores. As stated in Section 3.1, ReQuest uses Stanford NER to extract entity mentions in QA dataset and all QA pairs consist of two entity mentions and if either question or answer entity mention is not found, it drops the sentence. Beyond that, we have conducted experiments with four other ways to construct QA pairs from the raw QA sentences. As shown in Table 4.2, we lose many positive QA pairs if we only remain answer (or question) targets that are detected as named entities. Thus, we have tried to keep more positive pairs by relaxing the restriction from named entities to noun phrases. In addition, we have tried to evaluate the performance by 1) keeping negative pairs as named entity pairs or 2) changing them to noun phrase pairs. Besides that, inspired by [26, 27], the third processing variation we have tried is to parse the QA sentences into dependency paths and to extract features from these paths instead of the full sentences. The last one is that, we sample negative QA pairs not only from negative answer sentences, but also from positive sentence when extracting QA pairs. However, ReQuest achieves highest F1 score compared with these four processing variations (as shown in Figure 4.1) by filtering out all non entity mention answers, keeping full sentences and extracting only positive QA pairs from positive answer sentences.

Although by doing so, ReQuest filters out a large number of question/answer sentences and fewer QA pairs are constructed to provide semantic knowledge for RE, the remaining QA pairs provide cleaner and more consistent information with RE dataset. Thus, it still outperforms the other variations. Another interesting highlight is the comparison between using negative named entity pairs and using negative noun phrase pairs when positive QA pairs are formed by noun phrases. Although enforcing named entities is more consistent with RE datasets, a trade-off exists when the data format of positive and negative QA pairs are inconsistent. As we can see from the bar chart, the performance by using negative noun phrase pairs is better than negative named entity pairs.

CHAPTER 5: RELATED WORK

Classifying relation types between entities in a certain sentence and automatically extracting them from large corpora plays a key role in information extraction and natural language processing applications and thus has been a hot research topic recently. Even though many existing knowledge bases are very large, they are still far from complete. A lot of information is hidden in unstructured data, such as natural language text. Most tasks focus on knowledge base completion (KBP) [33] as a goal of relation extraction from corpora like New York Times (NYT) [2]. Others extract valuable relation information from community question-answer texts, which may be unique to other sources [34].

For supervised relation extraction, feature-based methods [28] and neural network techniques [35, 36] are most common. Most of them jointly leverage both semantic and syntactic features [29], while some use multi-channel input information as well as shortest dependency path to narrow down the attention [26, 27]. Two of he aforementioned papers perform the best on the SemEval-2010 Task 8 and constitutes our neural baseline methods.

However, most of these methods require large amount of annotated data, which is time consuming and labor intensive. To address this issue, most researchers align plain text with knowledge base by *distant supervision* [1] for relation extraction. However, distant supervision inevitably accompanies with the wrong labeling problem. To alleviate the wrong labeling problem, multi-instance and multi-label learning are used [2, 3]. Others [16, 31] propose joint extraction of typed entities and relations as joint optimization problem and posing cross-constraints of entities and relations on each other. Neural models with selective attention [4] are also proposed to automatically reduce labeling noise.

The distant supervision provides one solution to the cost of massive training data. However, traditional DS methods mostly only exploit one specific kind of indirect supervision knowledge - the relations/facts in a given knowledge base, thus often suffer from the problem of lack of supervision. There exist other *indirect supervision* methods for relation extraction, where some utilize globally and cross sentence boundary supervision [37, 38], some leverage the power of passage retrieval model for providing relevance feedback on sentences [39], and others [40, 41, 42]. Recently, with the prevalence of reinforcement learning applications, many information extraction and relation extraction tasks have adopted such techniques to boost existing approaches [43, 44]. Our methodology follows the success of indirect supervision, by adding question-answering pairs as another source of supervision for relation extraction task along with knowledge base auto-labeled distant supervision as well as partial supervision.

Another indirect supervision source we use in the paper, passage retrieval, as described here, is the task of retrieving only the portions of a document that are relevant to a particular information need. It could be useful for limiting the amount of non-relevant material presented to a searcher, or for helping the searcher locate the relevant portions of documents more quickly. Passage retrieval is also often an intermediate step in other information retrieval tasks, like question answering [45, 46, 47, 48] and combining with summarization. Some passage retrieval approaches [49] include calculating query-likelihood and relevance modeling [50], others show that language model approaches used for document retrieval can be applied to answer passage retrieval [51]. Following the success of passage retrieval usage in question-answering pipelines, to the best of our knowledge, we are the first to utilize passage retrieval, or specifically, answer sentence selection from question-answer pairs to provide additional indirect feedback and supervision for relation extraction task.

CHAPTER 6: CONCLUSION

We present a novel study on indirect supervision (from question-answering datasets) for the task of relation extraction. We propose a framework, ReQuest, that embeds information from both training data automatically generated by linking to knowledge bases and QA datasets, and captures richer semantic knowledge from both sources via shared text features so that better feature embeddings can be learned to infer relation type for test relation mentions despite the noisy training data. Our experiment results on two datasets demonstrate the effectiveness and robustness of ReQuest. Interesting future work includes identifying most relevant QA pairs for target relation types, generating most effective questions to collect feedback (or answers) via crowd-sourcing, and exploring approaches other than distant supervision [52, 53].

REFERENCES

[1] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL/IJCNLP*, 2009.

[2] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *ECML/PKDD*, 2010.

[3] R. Hoffmann, C. Zhang, X. Ling, L. S. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *ACL*, 2011.

[4] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *ACL*, 2016.

[5] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "Dbpedia spotlight: shedding light on the web of documents," in *I-Semantics*, 2011.

[6] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *EMNLP*, 2011.

[7] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *ACL*, 2014.

[8] Y. S. Chan and D. Roth, "Exploiting background knowledge for relation extraction," in *COLING*, 2010.

[9] N. Nguyen and R. Caruana, "Classification with partial labels," in *KDD*, 2008.

[10] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.

[11] J. Rao, H. He, and J. J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in *CIKM*, 2016.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[13] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[14] X. Ling and D. S. Weld, "Fine-grained entity recognition," in *AAAI*, 2012.

[15] J. Ellis, J. Getman, J. Mott, X. Li, K. Griffitt, S. M. Strassel, and J. Wright, "Linguistic resources for 2013 knowledge base population evaluations," in *TAC*, 2014.

[16] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han, "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *WWW*, 2017.

[17] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? a quasi-synchronous grammar for qa," in *EMNLP-CoNLL*, 2007.

[18] Z. Wang and A. Ittycheriah, "Faq-based question answering via word alignment," *arXiv preprint*, vol. arXiv:1507.02628, 2015.

[19] M. Tan, C. d. Santos, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *arXiv preprint*, vol. arXiv:1511.04108, 2015.

[20] C. N. dos Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," vol. arXiv:1602.03609, 2016.

[21] X. Yao, B. V. Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in *NAACL*, 2013.

[22] X. Yao, B. V. Durme, and P. Clark, "Automatic coupling of answer extraction and information retrieval," in *ACL*, 2013.

[23] R. J. Mooney and R. C. Bunescu, "Subsequence kernels for relation extraction," in *NIPS*, 2005.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.

[25] M. R. Gormley, M. Yu, and M. Dredze, "Improved relation extraction with feature-rich compositional embedding models," in *EMNLP*, 2015.

[26] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths." in *EMNLP*, 2015.

[27] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin, "Improved relation classification by deep recurrent neural networks with data augmentation," *arXiv preprint*, vol. arXiv:1601.03651, 2016.

[28] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *SemEval@ACL*, 2010.

[29] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," *arXiv preprint*, vol. arXiv:1601.00770, 2016.

[30] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel, "The automatic content extraction (ace) program - tasks, data, and evaluation," in *LREC*, 2004.

[31] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *ACL*, 2014.

[32] N. Bach and S. Badaskar, "A review of relation extraction," in *Literature review for Language and Statistics II*, 2007.

[33] M. Surdeanu and H. Ji, "Overview of the english slot filling track at the tac2014 knowledge base population evaluation," in *TAC*, 2014.

[34] D. Savenkov, W.-L. Lu, J. Dalton, and E. Agichtein, "Relation extraction from community generated question-answer pairs," in *NAACL*, 2015.

[35] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *EMNLP*, 2011.

[36] J. Ebrahimi and D. Dou, "Chain based rnn for relation classification," in *NAACL*, 2015.

[37] C. Quirk and H. Poon, "Distant supervision for relation extraction beyond the sentence boundary," *arXiv preprint*, vol. arXiv:1609.04873, 2016.

[38] X. Han and L. Sun, "Global distant supervision for relation extraction," in *AAAI*, 2016.

[39] W. Xu, R. Hoffmann, L. Zhao, and R. Grishman, "Filling knowledge base gaps for distant supervision of relation extraction," in *ACL*, 2013.

[40] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *IJCAI*, 2007.

[41] H. Poon and P. M. Domingos, "Joint unsupervised coreference resolution with markov logic," in *EMNLP*, 2008.

[42] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *EMNLP*, 2015.

[43] K. Narasimhan, A. Yala, and R. Barzilay, "Improving information extraction by acquiring external evidence with reinforcement learning," in *EMNLP*, 2016.

[44] P. H. Kanani and A. McCallum, "Selecting actions for resource-bounded information extraction using reinforcement learning," in *WSDM*, 2012.

[45] D. Savenkov and E. Agichtein, "When a knowledge base is not enough: Question answering over knowledge bases with external text data," in *SIGIR*, 2016.

[46] A. Ittycheriah, M. Franz, W.-J. Zhu, A. Ratnaparkhi, and R. J. Mammone, "Ibm's statistical question answering system," in *TREC*, 2000.

[47] D. Elworthy, "Question answering using a large nlp system," in *TREC*, 2000.

[48] M. Khalid and S. Verberne, "Passage retrieval for question answering using sliding windows," in *IRQA@COLING*, 2008, pp. 26–33.

[49] C. Wade and J. Allan, "Passage retrieval and evaluation," in *Tech. Reports of DTIC*, 2005.

[50] C. L. A. Clarke, G. V. Cormack, D. I. E. Kisman, and T. R. Lynam, "Question answering by passage selection (multitext experiments for trec-9)," in *TREC*, 2000.

[51] A. Corrada-Emmanuel, W. B. Croft, and V. Murdock, "Answer passage retrieval for question answering," in *Tech. Reports of CIIR UMass*, 2003.

[52] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *NAACL*, 2013.

[53] Y. Artzi and L. S. Zettlemoyer, "Weakly supervised learning of semantic parsers for mapping instructions to actions," *TACL*, vol. 1, pp. 49–62, 2013.