Deep Context Resolution

by

Junnan Chen

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2018

© Junnan Chen 2018

Authors Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of contributions

This thesis includes work carried out in collaboration with Rui Qiao (peer, friend, ph.D. candidate of University of Waterloo).

I would like to acknowledge Rui Qiao's contribution on designing the pronoun resolution model and conducted the experiments of the pronoun resolution model. He provided with me a lot of advices during this project. In this thesis, I formulated the problem, carried out the literature review, designed and implemented the models, conducted the experiments on these models, performed the analysis of the experiment results.

Abstract

Conversations depend on information from the context. To go beyond one-round conversation, a chatbot must resolve contextual information such as: 1) co-reference resolution, 2) ellipsis resolution, and 3) conjunctive relationship resolution.

There are simply not enough data to avoid these problems by trying to train a sequence-to-sequence model for multi-round conversation similar to that of one-round conversation.

The contributions of this paper are: 1) We formulate the problem of context resolution for conversation; 2) We present deep learning models, including an end-to-end network for context resolution; 3) We propose a way of creating a huge amount of realistic data for training such models with good experimental results.

Acknowledgements

I would like to thank Docter Ming Li for his support and help during my study.

I would like to thank Rui Qiao, Haocheng Qin for their great help and advices at all times.

I would like to thank Doctor Jimmy Lin and Doctor Pascal Poupart for reviewing this thesis.

I would like to thank my family for raise me up.

Dedication

This is dedicated to my family.

Table of Contents

List of Tables x

List of Figures xi

1 Introduction 1

1.1 Background 1

1.2 Problem Definition 1

1.2.1 Context Resolutions 1

1.2.2 Context Resolution Detection and Completion 4

1.3 Contributions 6

1.4 Thesis Organization 6

2 Background 7

2.1 Convolution Neural Network 7

2.2 Recurrent Neural Network 8

2.2.1 RNN encoder-decoder 9

3 Related Work 11

4 Creating Training Data 13

4.1 Training Data Components 14

4.2 Training Data Construction 15

4.2.1 Data Collection 16

4.2.2 Keywords Detection 16

4.2.3 Data Label Generation 18

4.3 Addressing Issues In Constructed Data 18

4.3.1 Reducing False Negatives 19

4.3.2 Reducing False Positives 21

5 Context Resolution Models **22**

5.1 Co-reference and Ellipsis Detection Model 22

5.1.1 Sequence-CNN-pooling: 22

5.1.2 LSTM decoder: 25

5.1.3 Prediction: 26

5.1.4 Loss Function: 26

5.2 Pronoun Resolution Model 26

5.3 The End-to-end Model 27

5.3.1 Loss Function 27

6 Experiments **29**

6.1 Datasets And Formats 29

6.2 Implementing Details 30

6.3 Results 30

6.3.1 Co-reference And Ellipsis Detection Model 30

6.3.2 Pronoun resolution model 31

6.3.3 End-to-end Model 32

6.3.4 Selected Results 32

7 Conclusion and Discussion **34**

References **35**

APPENDICES **40**

.1 Training Data Creation Process 40

.1.1 Word labeling 40

.1.2 Word replacement 41

List of Tables

6.1 Number of sentences for each dataset in this paper. 29

6.2 Accuracy, Recall and Precision Rates of Co-reference and Ellipsis Detection models 30

6.3 Top K accuracies of Pronoun Resolution models 31

6.4 Accuracy, precision and recall rate, and top K resolution accuracies of the end-to-end model 32

6.5 Selected Examples 33

1 Examples of sentences and translations with labeled words (red) and other special words (blue) of nine types 44

2 Data size of each type, unit: 10000 46

List of Figures

1.1 Multi-round conversation breakdown 2

1.2 An illustration of breaking a multi-round conversation into complete single-round conversations. 4

1.3 An example of the detection and completion process. 5

2.1 Two layer CNN example 8

2.2 RNN structure 9

2.3 RNN Encoder-decoder Architecture 10

4.1 The flow chart of training data construction 15

4.2 An example of creating training data 17

4.3 Venn diagram: area **A** stands for false negative examples, area **B** stands for true positive examples, area **C** stands for false positive examples. 19

4.4 The person on the left actually asks for information about Pierre Trudeau when he says *When did he die?*. 20

5.1 Co-reference and Ellipsis Detection Model Architecture 23

5.2 Word and Pos-tag encoding layer 24

5.3 The Sequence-CNN-pooling structure 24

5.4 Embedding layer to Sequence-CNN-pooling layers 25

5.5 LSTM decoder layers 25

5.6 End to end Resolution Model Architecture 28

1 Classifications for Sentences Contain No Labeled Words 43
2 Classifications for Sentences Contain Labeled Words 43
3 Classifications for Sentences Contain Labeled Words Continued 43

Chapter 1

Introduction

1.1 Background

Conversation has helped to create human civilization and conversation will soon become a part of the platform for the next generation of human-computer interface.

For one-round conversation, the problem can be reasonably solved by a combination of traditional approach and a sequence-to-sequence model [39] trained on massive social network conversation data.

However, this approach does not generalize to more than one round conversation easily. We face the problem of co-reference resolution, ellipsis resolution, and conjunctive relationship resolution, or the multi-round conversation data requirement immediately leads to a geometric growth of training data: if we need $\Omega(N)$ data to train a one-round sequence-to-sequence conversation model, then potentially we need $\Omega(N^2)$ data to train two rounds, and $\Omega(N^k)$ for k rounds. This data requirement is problematic even for a single vertical domain.

1.2 Problem Definition

1.2.1 Context Resolutions

We wish to decompose a multi-round conversation or dialogue into many single rounds by loading the prior context relevant semantics to the current and independent sentence or question.

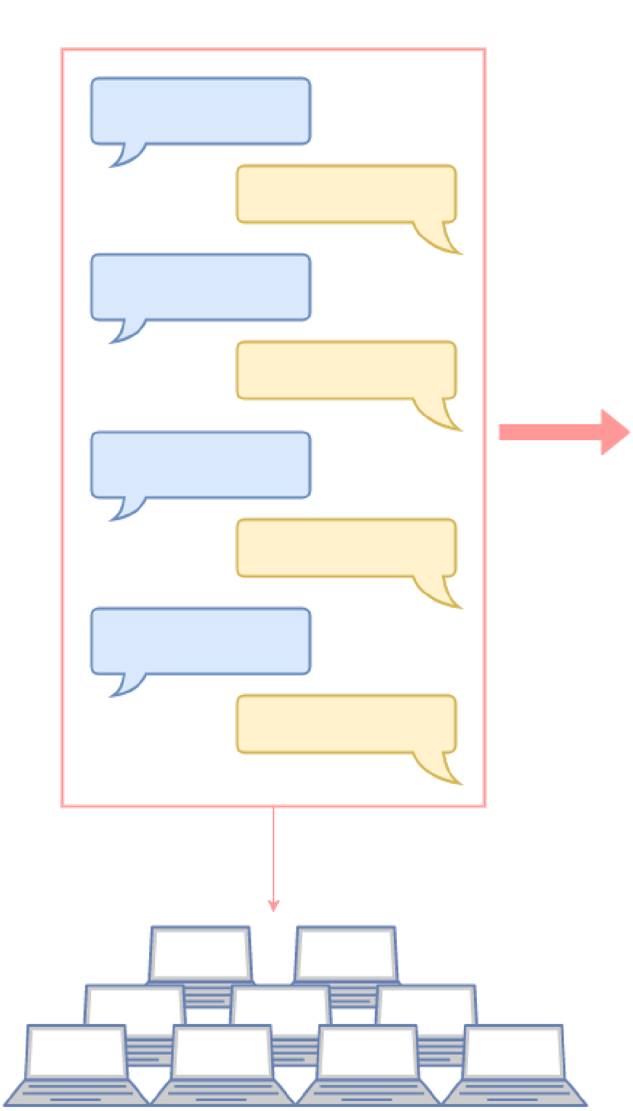

Figure 1.1: Multi-round conversation breakdown

Classical studies on co-reference, ellipsis, and conjunctions all appear in the context of conversations in different forms, and require different kinds of training data. We define three types of context resolution problems as follows.

Co-reference resolution: Co-reference in linguistics is defined as the phenomenon when two or more expressions in a text refer to the same person or thing; they have the same referent. In the case of conversation, we do not consider the same sentence co-reference because such co-reference can be taken care of by our one-round conversation mechanism. We also do not consider cataphora (when the anaphor appears after the antecedent) as it does not naturally happen in a conversation. We will particularly focus on co-references for names, time, locations, and noun phrases. For example, “it” in *-I went to dinner at Jim’s last night. -Was it delicious?* is a case of co-reference. Identifying “it” with “dinner” is the task of co-reference resolution.

Ellipsis resolution. Ellipsis happens more frequently in conversation than in a one-person monologue. Ellipsis in conversations means that the expression of an entity is omitted when the entity occurred in the previous conversation. For example, *What are the ingredients in McDonalds fries? Why taste bitter when cooling down?* Ellipsis resolution is to complete the second sentence to *Why McDonalds’ fries taste bitter when cooling down?*.

Conjunctive relation resolution: In texts, conjunctions help connect sentences. conjunctive relationship means two sentences are syntactically connected by conjunctive words[1]. The semantic meanings of these sentences are complete only when the sentences are joined together. While co-reference and ellipsis indicate a reference to some object from the preceding sentence, conjunctive generally indicate a reference to the entire preceding sentence. These conjunctions can be *coordinating*, such as for (reason), but/yet (contrast), and (addition), or (alternative), so (consequence); *subordinating*, such as after, although, as, as long as, because, before, which usually introduce some condition to the main sentence; or *correlative*, which are pairs such as both/and, whether/or, either/or, neither/nor, not/but, rather/than.

In the next round conversation, often speakers, sometimes losing such conjunctions, add or ask for: agreement, nouns, reasons, alternatives, conditions, or consequences. For example, **A**: What do you like to do? **B**: Read books. **A**: And? Or **A**: I hate books. In this latter case, A omitted conjunction “But”. We collectively call these “conjunctive relations”.

The goal is to convert a multi-round dialogue to many single-rounds, and solve the resolution problems so that these single-round sentences have complete and independent semantic meanings, so are answerable.

Converting the multi-round dialogue into single-round ones and changing the problem to “context resolution” problems significantly reduces the dimensionality of the problem.

This certainly does not solve all the problems, and to load all the semantics from the context into a single sentence often is impossible or extremely cumbersome. For example, consider a dialogue with two people discussing restaurants they liked and disliked for ten minutes, then one asks: “what is a restaurant that is liked by both of us?” However, we will cover a large fraction of practical cases. We will at least be able to handle co-reference cases such as: *I went to dinner at Jim's last night. Was it delicious?* and *Who is the prime minister of Canada? Who is his wife?*; ellipsis cases such as: *Will Tom go to the party? Unless invited.*, and *What is the population of Canada? What about China?*

1.2.2 Context Resolution Detection and Completion

In order to tackle the above context resolution problems in conversations, we furthur break down the problem into two tasks, as shown in Figure 1.2:

1 context resolutions detection and classification.

2 incomplete sentences completion.

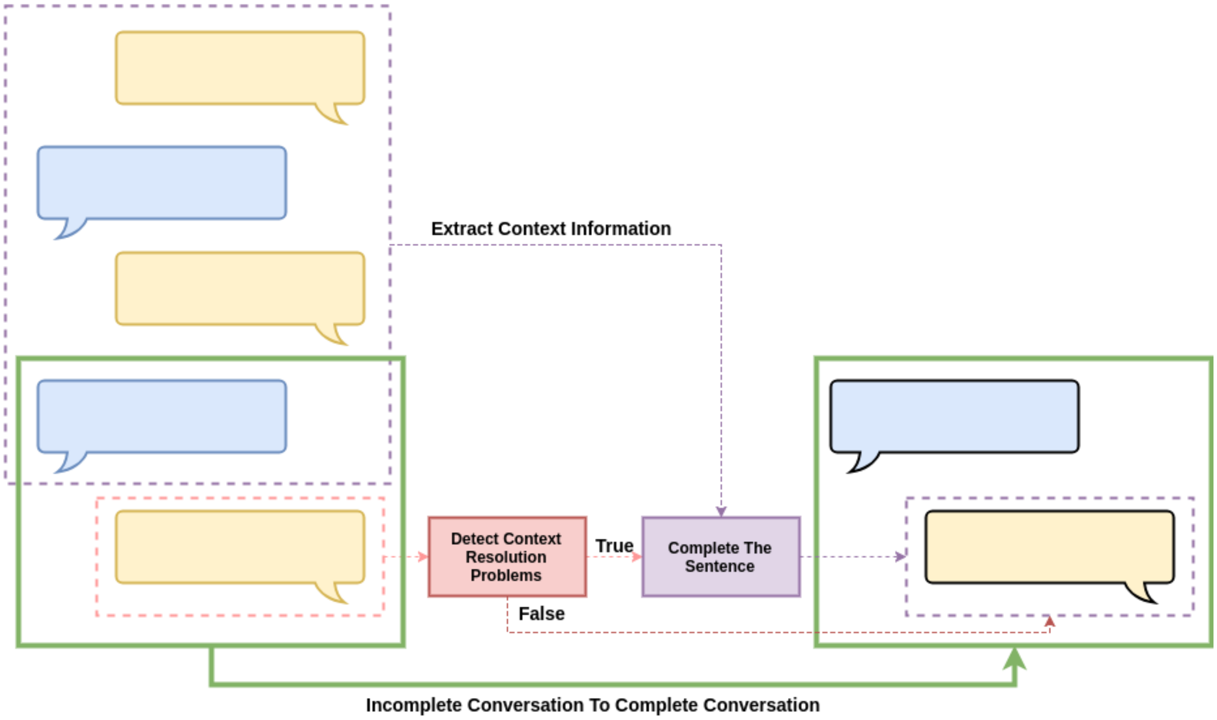
Extract Context Information
Detect Context Resolution Problems
True
Complete The Sentence
False
Incomplete Conversation To Complete Conversation

Figure 1.2: An illustration of breaking a multi-round conversation into complete single-round conversations.

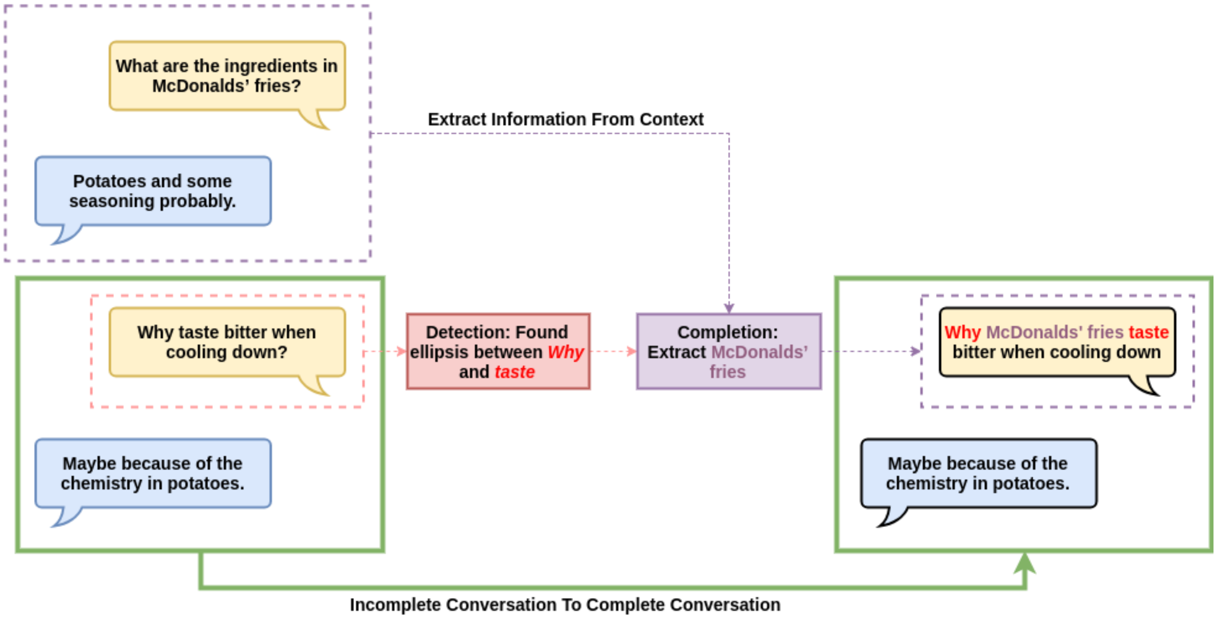
What are the ingredients in McDonalds' fries?
Potatoes and some seasoning probably.
Extract Information From Context
Why taste bitter when cooling down?
Maybe because of the chemistry in potatoes.
Detection: Found ellipsis between Why and taste
Completion: Extract McDonalds' fries
Why McDonalds' fries taste bitter when cooling down
Maybe because of the chemistry in potatoes.
Incomplete Conversation To Complete Conversation

Figure 1.3: An example of the detection and completion process.

For any sentence from a conversation, we first need to detect whether there exists co-reference or/and ellipsis in the sentence, the types and positions where those context related problem occurs in the sentence. There are four types of co-reference and ellipsis we particularly focus on: **people, time, locations and noun phrases**. From the previous example in Fig 1.3, an example of ellipsis occurs in *Why taste bitter when cooling down?*, after the word *Why* and before the word *taste*. Next, the missing part of the sentence needs to be located from the contexts. Therefore, The noun phrase *McDonalds fries* in the context *What are the ingredients in McDonalds fries?* is our target to complete the conversation.

To tackle the first task, we build a deep neural network model to

1 detect the co-reference and/or ellipsis in a given sentence

2 locate the position of them

3 classify the type of co-reference and ellipsis

For the second task, if the type of context resolution is one of people, time and locations, the related entity could be easily retrieved from the context with the help of a Named

Entity Recognizer such as [13]. The task becomes nasty when it comes to noun phrase type, for the reason that there could be multiply noun phrases in the previous sentence. Therefore, we build a deep neural network model to select the best matching noun phrase of a given incomplete sentence by scoring all the potential noun phrases with the sentence.

1.3 Contributions

The contributions of this thesis are: 1) We initiate a comprehensive study of context resolution in conversation, where we defined a method to breakdown multi-round conversations into single-round conversations and reduced the dimensionality of the problem. To our knowledge, this is the first attempt at systematically defining and solving the problem of contextual resolution in conversation.

2) We present deep learning models for context resolution.

3) We propose our novel method of generating a huge amount of realistic data for training such models with good experimental results.

1.4 Thesis Organization

Chapter 2 and Chapter 3 introduce the background knownledge and related work of context resolution. Chapter 4 introduces the method on constructing training data. Chapter 5 and Chapter 6 introduce the deep neural network models we proposed and the experiment results. Chapter 7 introduces the conclusion and discussion of this thesis.

Chapter 2

Background

2.1 Convolution Neural Network

Convolution Neural Networks(CNN) efficiently captures local features. Ever since LeCun et al.,[27] first applied backpropagation and gradient-based learning to train CNN and succeded in document recognition, CNN has become one of the most widely used neural networks on various areas such as Image Recognition [24], Speech Recognition [4] and Natural Language Processing [14].

CNN preserve the spatial structure of the input matrix. For example, as shown in Figure 2.1, the input is a $(28 \times 28 \times 3)$ matrix, which means the height and width of the input is 28 and the depth of which is 3. Then, we employ a "filter" of size $(5 \times 5 \times 3)$. We convolve the filter with the input by sliding the filter both vertically and horizontally over the input and compute the dot products. Therefore, after "convolving", we result in a $(24 \times 24 \times 1)$ matrix. Multiple filters could be applied on the same input to extract different features. For example, if we apply 8 filters, the result will be of size $(24 \times 24 \times 8)$. Activation functions such as ReLU and sigmoid are commonly applied on top of the dot products.

Pooling layers basically downsample the result and make the representation smaller. For the previous example, if we apply a max pooling layer with (3×3) filters with stride 3 only on the height and width, we will end up in the result of size $(8 \times 8 \times 8)$.

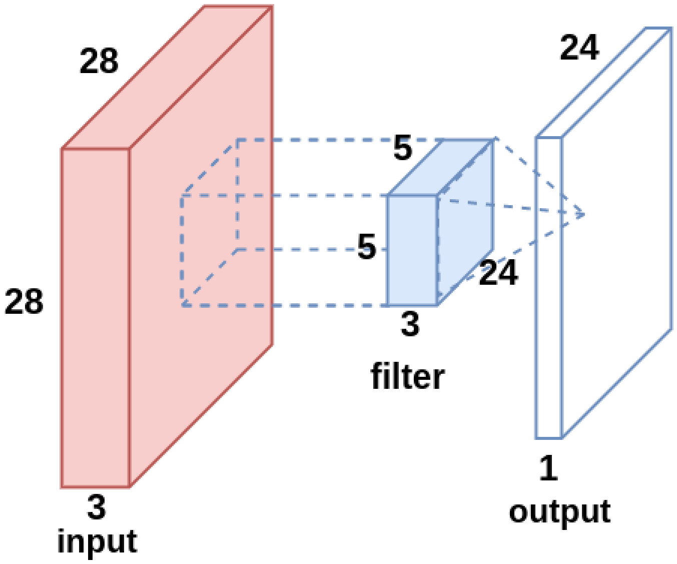

Figure 2.1: Two layer CNN example

2.2 Recurrent Neural Network

Recurrent Neural Networks(RNN) has shown promising results on processing sequencial data. It has been applied on Image Generation [18], Sequence-to-Sequence learning [43] and language modeling [42].

A general RNN cell has the following structure as shown in Figure 2.2. The key to handling sequential data is that, every time an input is fed to the RNN cell, the RNN cell will compute and update its hidden state which will be fed back into the model the next time a new input is fed. We denote a sequence of vectors as $\mathbf{x}$, the input vector at each time step as x_t. We can formularize a RNN cell as:

$$h_t = f_W(h_{t-1}, x_t)$$

, where h_t denotes the hidden state at time step t, and f_W is the recurrent activation function with parameters W of the RNN cell. Different RNN cells have different functions and compute y_t according to W differently.

Long Short Term Memory networks (LSTM) are a type of RNN introduced by [21]. In LSTM, a sigmoid layer called "forget gate" is employed to avoid useless history information. The LSTM cell has the following recurrent activation function.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

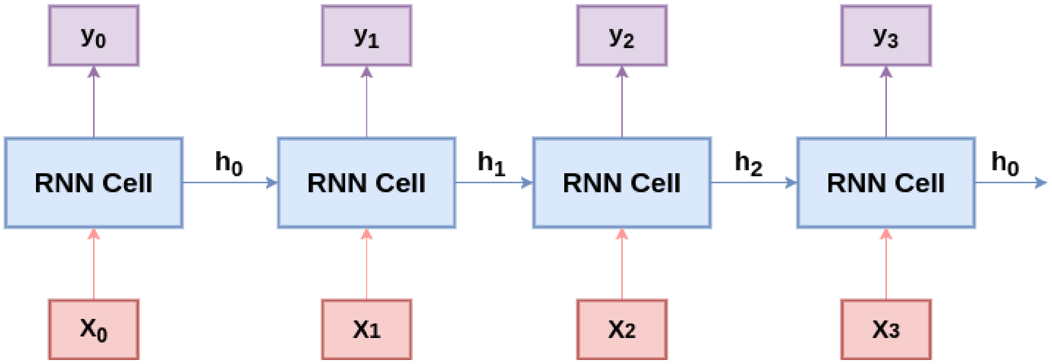

Figure 2.2: RNN structure

$$
\begin{gathered}
i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t = \sigma(W_O \cdot [h_{t-1}, x_t] + b_O) \\
h_t = o_t * tanh(C_t)
\end{gathered}
$$

, where x_t is the input vector, $W_f, W_i, W_C, W_O, b_f, b_i, b_C, b_O$ are weight matrices and bias vector parameters of the LSTM cell, C_t is the cell state vector and h_t is the output vector of the LSTM cell.

2.2.1 RNN encoder-decoder

The RNN encoder-decoder was first introduced by [10], the idea is to model an entire process, such as machine translation [10] and conversation generation [40], through a neural network. An RNN encoder-decoder consists of two RNN cells, as shown in Figure 2.3. The RNN cell that reads the input sequence one at a time is the encoder. The final hidden state h_N of the encoder contains the compressed information of the entire input sequence. The RNN decoder decodes the information from h_N. There are many variances on the decoder, [5] uses the weighted sum of the encoder's output at each time step as the input of the decoder, [10] uses h_N as attention and feed it to the decoder at each time step along with the output of the decoder at last time step.

The RNN encoder-decoder provides a possibility for end-to-end training where all the parameters are optimized at the same time. It is particularly useful for context resolution since it has the power to expliote and understand the contexts.

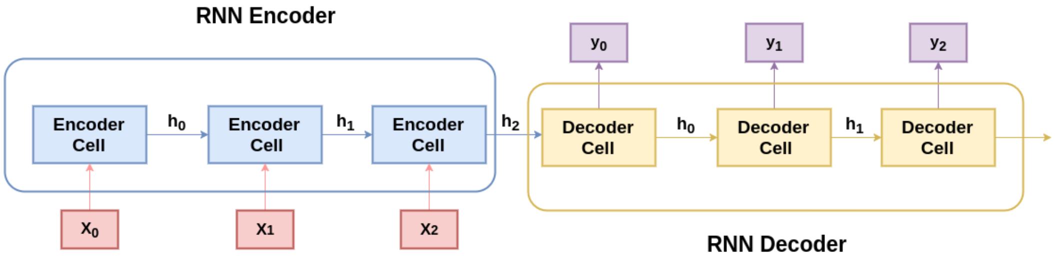

Figure 2.3: RNN Encoder-decoder Architecture

Chapter 3

Related Work

For text, there has been much classical or linguistic theoretical work on co-reference resolution and some work on ellipsis resolution, as well as conjunctions in texts. Some used deep learning for co-reference in text [47], with limited datasets, applications, and models.

There are three main aspects that distinguish our study from existing work. First, conversation is quite different from text. In conversation, we do not consider co-reference or ellipsis in the same sentence (these are usually resolved by cQA, grammar rules, or an lLSTM), neither do we consider cataphora, where the cataphor precedes the expression to which it refers. The conjunctive relation resolution is more subtle, as in text there are conjunction words, but in conversation they are sometimes naturally missing.

Second, we have created extensive training data to enable our learning. In text, there are two small well-labeled datasets: the MUC and CoNLL-2011 shared task dataset. These are far from enough and do not suit the conversation task. Our constructed data, however, do not have context information during the training phase. That is, for co-reference, we only learn the "reference" but do not have the "referenced entity" in the training data.

Last, other than classical studies of "ellipsis resolution", we have not found machine learning based research on "ellipsis resolution" or "conjunctive relation resolution", even in text. Texts are typically one personal monologue and the conjunction words logically connect two sentences, thus there is no need to do "conjunctive relation resolution".

Orthogonal to our research, there are other approaches using generative adversarial network and reinforcement learning, for example, to avoid topics that would end a conversation [28, 29].

In the following sections, we introduce neural networks that we applied in our models, including word embedding, convolution neural networks, recurrent neural networks, and their variances.

Chapter 4

Creating Training Data

Context resolution requires large scale and annotated training data. Obtaining such a data set is key to this project. We now describe our idea of how to obtain such data, via one concrete example.

The ideal training data for co-reference is a collection of small conversations, with proper labels, such as follows:

A: Who is Pierre Trudeau?
B: Justin's father.
A: When did he die?
Label: When did Pierre Trudeau die?

Thus, using this ideal data set to train a neural network, we will be able to complete "When did he die" to "When did Pierre Trudeau die?" Then a one-round conversation engine can continue the conversation smoothly.

Although we do not have such ideal data sets, we do have many sentences such as "When did Pierre Trudeau die?" That is, different from normal learning tasks where putting labels on the data is a major problem, here, we have many labels, but no data. However, observing that the "data" (When did he die) in fact is not that different from the "label" (When did Pierre Trudeau die), and with careful manipulations, we can construct the data:

When did [he] die?

While this is an easy case, many other cases are more complicated and to be discussed next.

4.1 Training Data Components

First, we introduce the core components of our dataset. In order to precisely detect co-reference and ellipsis, our dataset consists of three main parts: positive co-reference, positive ellipsis, and negative examples.

Positive co-reference examples are positive data of co-reference cases. They are sentences that contain pronouns that refer to entities from the context. Pronouns that have referring functions are labeled. Positive co-reference examples give our model the ability to detect real co-reference cases.

Positive ellipsis examples are sentences that contain ellipsis cases. Entities from the context are omitted in these sentences leaving the sentence incomplete. The positions of the missing entities are labeled. Positive ellipsis examples give our model the ability to detect real ellipsis cases.

Negative examples are complete sentences and have complete semantic meanings. They have two main purposes: 1) to work as control groups to help the model distinguish real co-references and ellipsis from those complete sentences with similar structures, 2) and to provide a large variety of natural language patterns.

There are two types of negative examples. The first type is negative examples for co-reference. Pronouns in natural languages have multiple meanings. For example:

1. it seems like you have not slept yet.
2. it belongs to Jim.

In the first sentence, “it” does not have co-reference meaning. While in the second sentence, “it” does. Sentences that contain pronouns, but do not have reference meanings are included in this part. These sentences give our model the ability to distinguish pronouns and reference words that have co-reference functionality from those that do not.

The second type is general negative examples. These sentences are complete sentences and have a large variety of patterns. They serve as both the negative examples of ellipsis cases and general language pattern providers. These sentences provide our model with complex natural language patterns and give our model the ability to distinguish between complete sentences and incomplete sentences.

4.2 Training Data Construction

Next, we introduce how we create the above three components. There are four phases in the data construction process: data collection, keywords detection, data modification, and data label generation. Figure 4.1 shows the workflow of the training data construction process. A full process of constructing the Chinese dataset with details is in Appendix.1.

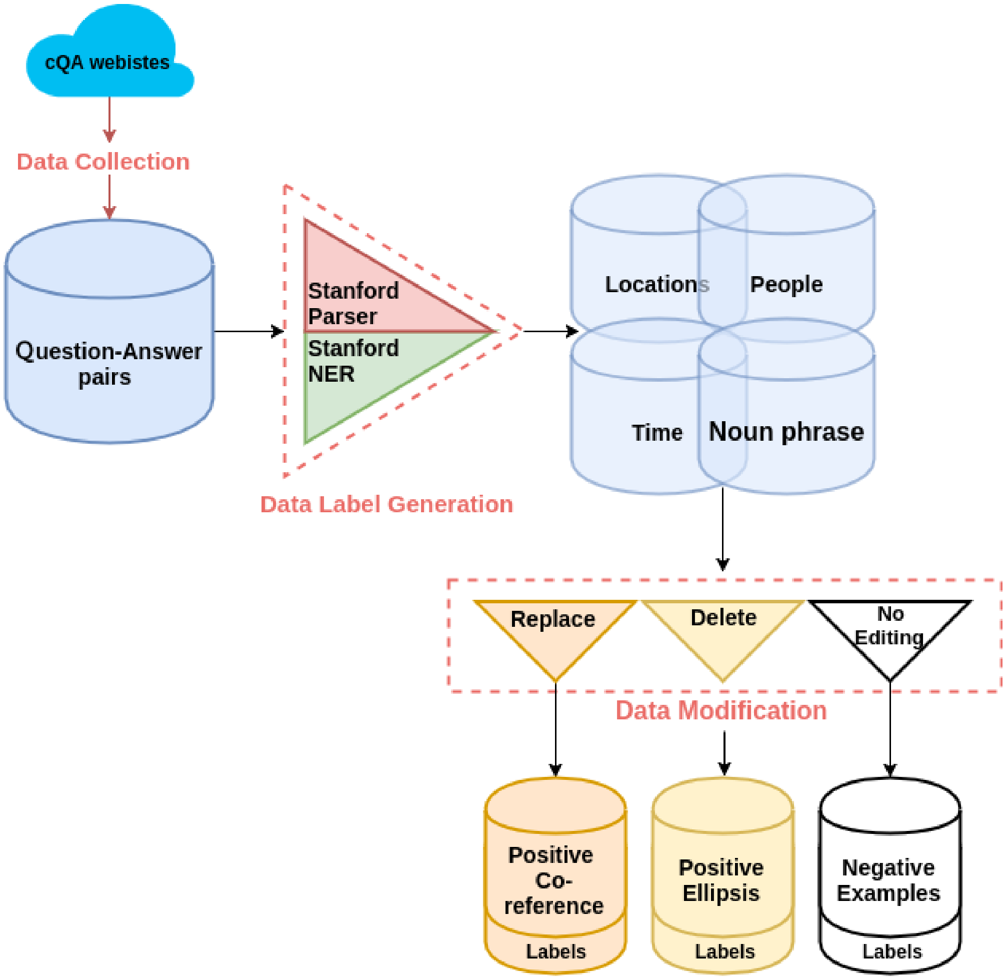

Figure 4.1: The flow chart of training data construction

4.2.1 Data Collection

Sentences in dialogues have the features of being short and containing only one or two entities. Language data from Community Question Answering (cQA) websites fit our purpose perfectly since 1) these questions and answers tend to be short and precise; 2) large user groups provide a huge corpus data; 3) these single round question-answering dialogues share some language features with chatting dialogues.

Initially, QA pairs from the internet are collected. These are our *raw data*. These raw data are mostly precise, complete, short, and independent sentences and contain no co-references to the context.

Datasets are constructed out of these raw data. We will focus on four types of most common entities: location, time, people, and noun-phrases. A large number of co-reference, ellipsis, and conjunctive relationships occur to them. Therefore, we construct four datasets in total. Each dataset contains training data for a certain type of entity.

4.2.2 Keywords Detection

First, we need to detect and label words that refer to *time*, *location*, *people* or *noun phrases*. We parse questions using the Stanford Parser [38] to generate syntax trees annotated with POS tags. The POS-tags provide syntactic information that helps guide the data generation rules. Then, we use the Stanford Named Entity Recognizer (NER) [13] to tag tokens that refer to *time*, *location* or *people* entities. We call these words *marked words*. The positive examples and negative examples are then randomly sampled to keep the ratio balanced.

Data Modification

Our goal is to transform short sentences from dialogues into positive examples of coreference and ellipsis. The main challenge in generating those is to identify segments that can be omitted or replaced with a pronoun so that the resulting sentence is both grammatical and natural. Our method modifies complete sentences into sentences that contain co-reference or ellipsis according to syntactic patterns.

Create positive co-reference examples: Since pronouns in co-reference sentences actually refer to an entity from the context, we can reverse the process and create co-reference cases by replacing entities with pronouns in sentences. It is feasible also because for a certain entity type (e.g. time), the corresponding pronouns are limited. Therefore, we create positive co-reference examples by replacing the marked words in the sentence by certain pronouns.

Create positive ellipsis examples: For the same reason as above, the process of understanding ellipsis could be reversed. We can create ellipsis cases by omitting entities in sentences. Therefore, we create ellipsis cases by deleting the marked words in the sentence directly.

The above two modifications could result in some sentences that do not make sense. However, since our raw data consists of a large variety of sentence patterns, refining the grammar constraints could limit this disadvantage greatly.

Create negative examples: Because of the functionality of negative examples as stated above, these sentences are complete sentences. In order to provide enough language patterns, negative examples are randomly sampled out of raw data. In addition, a number of complete sentences that contains pronouns already are added. It could enhance our model's ability to distinguish real co-reference and "fake" co-reference.

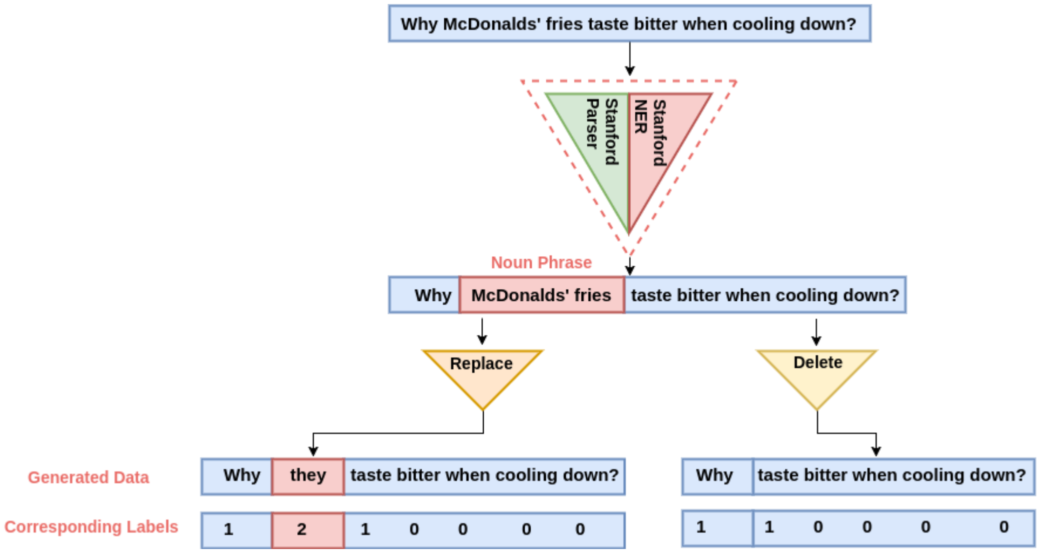
Why McDonalds' fries taste bitter when cooling down?
Stanford Parser
Stanford NER
Noun Phrase
Why
McDonalds' fries
taste bitter when cooling down?
Replace
Delete
Generated Data
Why
they
taste bitter when cooling down?
Why
taste bitter when cooling down?
Corresponding Labels
1
2
1 0 0 0 0
1
1 0 0 0 0

Figure 4.2: An example of creating training data

4.2.3 Data Label Generation

Given a modified sentence s consisting of m words, denotes as

$$\mathbf{s} = (s_1, \dots, s_m)$$

, the label is a sequence with the same length as s, as

$$\mathbf{l} = (l_1, \dots, l_m)$$

Each element in $\mathbf{l}$ corresponds to a word in s and indicates the type of its corresponding word. There are three types:

- **type 0:** the word is not referring to any entity,
- **type 1:** the word is next to an omitted or replaced entity,
- **type 2:** the word is referring to an entity.

Figure 4.2 shows an example of the data creation procesure. Note that there could be two or more consequent 2s in one label for the reason that a co-reference case could be more than one word.

Therefore, two consequent 1s in one label indicates an ellipsis occurance between these two positions. One 1 followed by one or more 2s, followed by zero or one 1 in one label indicates an co-reference occurance there.

4.3 Addressing Issues In Constructed Data

Constructed data are not exactly the same as real data. Due to the fact that a group of grammar rules is applied when constructing the data, the most common “hidden danger” is that there are differences between the real data and the constructed data, where some real co-reference and ellipsis cases are either not covered or categorized incorrectly. Therefore, we face the problem that the performance of our model will decrease when facing user-generated real data. In this section, we will introduce how we address the issues which are related to the differences between the constructed data and real data, in order to 1) generalize the constructed data to make it more realistic, and 2) reduce the impact on the performance of our deep neural network models due to those differences.

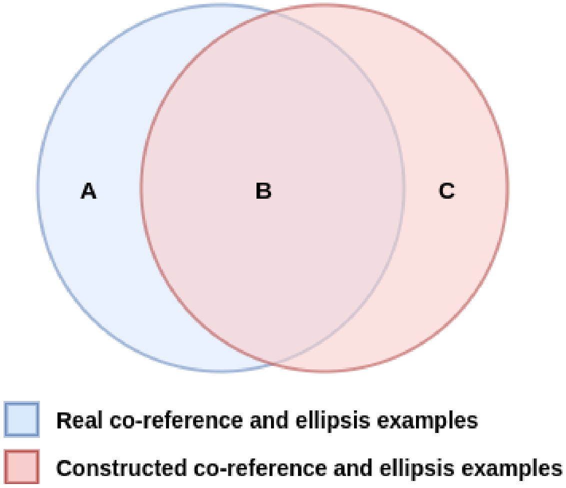
A
B
C
Real co-reference and ellipsis examples
Constructed co-reference and ellipsis examples

Figure 4.3: Venn diagram: area **A** stands for false negative examples, area **B** stands for true positive examples, area **C** stands for false positive examples.

First, we define **false negative** and **false positive** regarding the differences between the constructed data and real data as follows, see Figure 4.3. False negative examples are the real data that do contain co-reference or ellipsis but are not included in the constructed dataset. False positive, on the contrary, means that real data that do not contain co-reference or ellipsis but categorized as they do in the constructed dataset.

4.3.1 Reducing False Negatives

The key here is to distinguish "real" co-reference/ellipsis among the corpus. Therefore, first, we enlarged the coverage of our constructed data and reduce the portion of uncovered real cases. Intuitively, we want area **B** in Figure 4.3 as large as possible. As humans, when we come up with a sentence that has co-reference or ellipsis in it, most of the time we actually know the complete and exact meaning of the sentence in our brains. See the example in Figure 4.4, this provides the insight that a part of the co-reference and ellipsis occurrence could actually be transformed from complete sentences. Therefore, we tried to mimic the procedure when humans actually transform a sentence into co-reference and ellipsis sentences in their brains by a large group of grammar rules that follows our language nature, so that this part of real data widely covered in the constructed data. For example, in the Chinese dataset, nine categories of "candidates" that could be transformed into co-reference and ellipsis data are proposed and each with detailed syntactic rules to keep the transformation as smooth and realistic as possible. Applying such rules on our 300 million collected raw data provides the prosperousness in the constructed data.

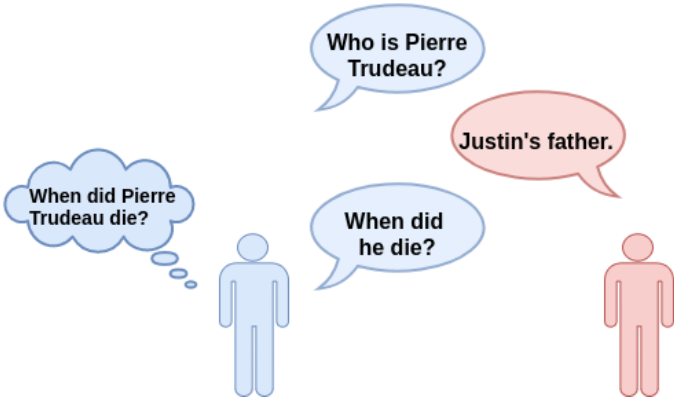

Figure 4.4: The person on the left actually asks for information about Pierre Trudeau when he says *When did he die?*.

However, despite the part of data that could be transformed from complete sentences naturally with grammar rules, there still exists real examples that violate those assumptions. For instance, gender and number agreements could be very tricky to cover fully. Although there exists a numerous number of syntactic forms of co-reference and ellipsis in natural linguistics, the semantics and sentiments in the sentences are similar. Given an unfamiliar syntactic form, the human can still understand the sentence base on the meanings of the words in a sentence. For instance, “Saw the trailer? Pretty cool right?” is a syntactic form that is difficult to cover by grammar rules. However, with the help of “saw”, ”trailer” and “cool”, we can inference that there occurs an ellipsis in a movie or a television show.

Therefore, the key to making the model have good performance on real cases that are not covered by the grammar rules is to fit the model on the semantics and sentiment meanings as well as the syntactic of the sentences, and most importantly, to avoid overfitting on the grammar rules we generated. To achieve this, we first employ a large amount of corpus to provide the prosperousness in semantics. For the Chinese dataset, the 300 million collected sentences are rich in semantics and sentiments. And also, an advantage of using deep neural network model is that we can introduce prior sentiment information by using pre-trained word embedding [31]. Secondly, to enhance the richness of semantics in the constructed data, the grammar rules are strictly syntactic based. That is, instead of writing grammars based on words, grammars are based on the syntactic roles (e.g. subjective, objective, pronoun). In this way, the constructed data will keep the richness in the original corpus without filtering out sentences that do not contain certain words.

4.3.2 Reducing False Positives

False positives examples are examples are cases that have similar syntactic structure, but naturally do not have co-reference or ellipsis inside. The challenge here is to make the model learn to distinguish positive and negative cases given syntactically similar sentences. For example, “it seems like you have not slept yet.” and “it belongs to Jim” all have “it” followed by a verb. The first “it” does not have co-reference meaning while the second does. Therefore, the problem could be addressed by adding negative training data. Adding a certain number of sentences like the first sentence in the example to the dataset makes the dataset balanced. Selecting negative examples is very important in constructing the dataset. For the Chinese dataset, we first carefully select complete sentences that have similar syntactic structures to our “candidate” structures and keep the ratio of negative and positive examples 1 : 1. In addition, these sentences also provide a large variety of natural language patterns which also benefits in reducing the false negatives.

Chapter 5

Context Resolution Models

5.1 Co-reference and Ellipsis Detection Model

The Co-reference and Ellipsis Detection model is to predict the type of each word suggested in Section 4.2.3 of a given sentence $\mathbf{s} = \{s_1, \ldots, s_n\}$ and its corresponding POS-tags $\mathbf{t} = \{t_1, \ldots, t_n\}$. We formulate the model as follows:

$$\{d_1, \ldots, d_n\} = \mathcal{F}(\mathbf{s}, \mathbf{t})$$

where $\mathcal{F}$ indicates our model, d_i indicates the probability distribution over the three types of word s_i. Initially, $\mathbf{s}$ and $\mathbf{t}$ could be of various length. They are first zero-padded into a fixed length m. We denote the padded sentence as $\mathbf{s} = \{s_1, \ldots, s_m\}$, where s_i is a word. We denote the padded POS-tag sequence as $\mathbf{t} = \{t_1, \ldots, t_m\}$ as well. Our model has the following components.

Word and POS-tag encoding: As shown in 5.2, we first apply a 200-dimensional embedding [31] to $\mathbf{s}$ and a 15-dimensional embedding to $\mathbf{t}$. Let $\mathbf{s^e} = \{s^e_1, \ldots, s^e_m\}$ and $\mathbf{t^e} = \{t^e_1, \ldots, t^e_m\}$ be the embedded representations. As suggested by [16], we also include the position embeddings in the model, denoted as $\mathbf{p} = (p_1, \ldots, p_m)$. The word embeddings and positional information are incorporated together as $\mathbf{e} = \{s^e_1 + p_1, \ldots, s^e_m + p_m\}$. Finally $\mathbf{t^e}$ and $\mathbf{e}$, are concated together as the final output: $\mathbf{p} = \{p_1, \ldots, p_m\}$, where $p_i = [s^e_1 + p_1, t^e_1]$.

5.1.1 Sequence-CNN-pooling:

Recently, Convolutional Neural Networks (CNN) based models have shown promising results in sentiment analysis [22] and translation [16]. Inspired by the recent success of

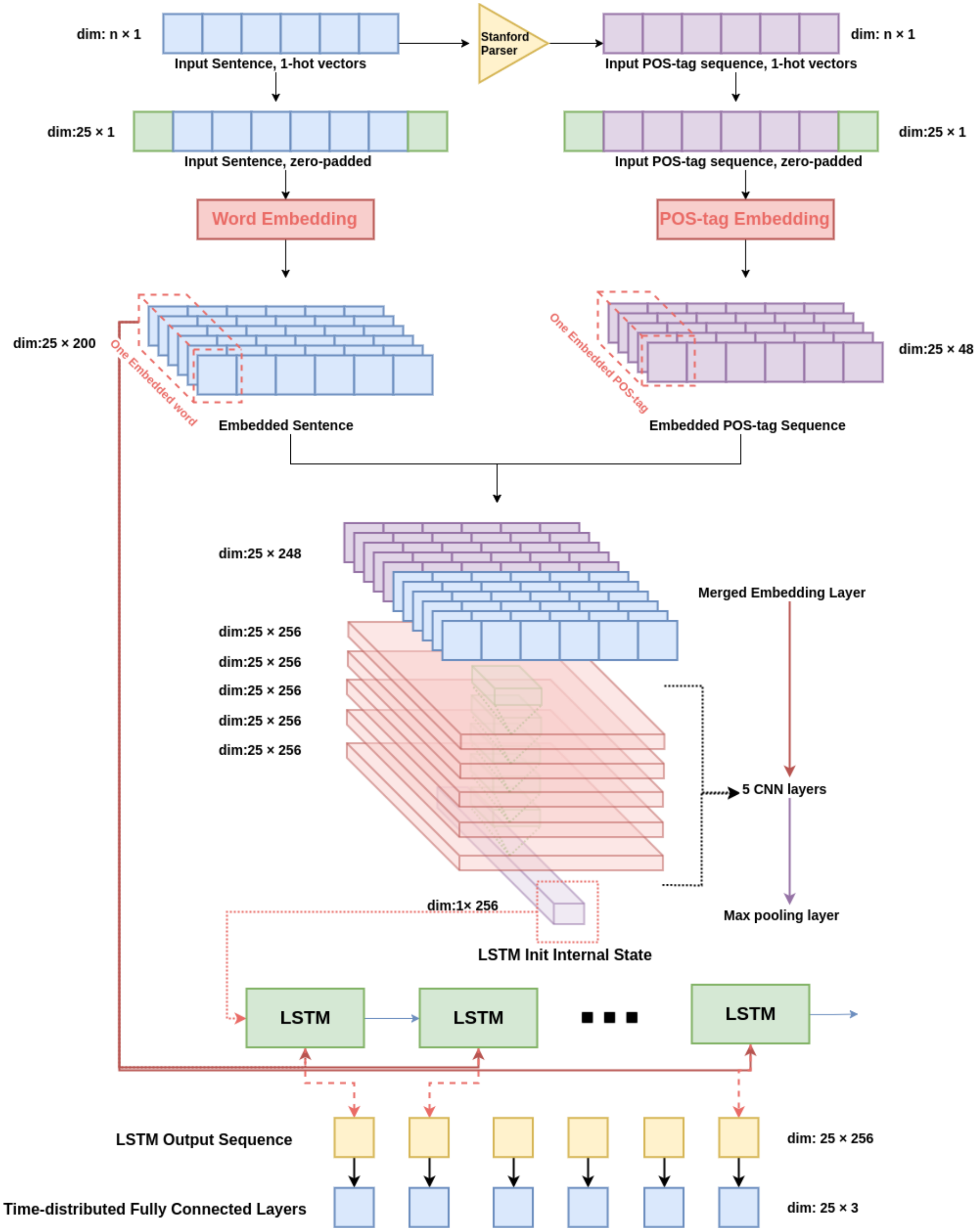

Figure 5.1: Co-reference and Ellipsis Detection Model Architecture

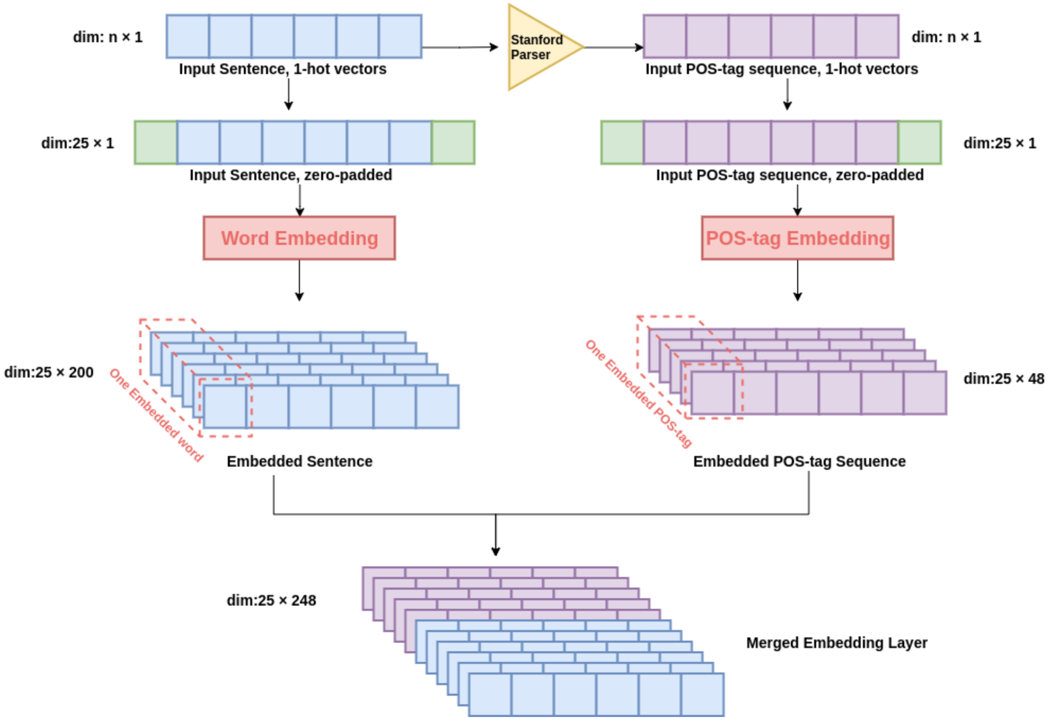

Figure 5.2: Word and Pos-tag encoding layer

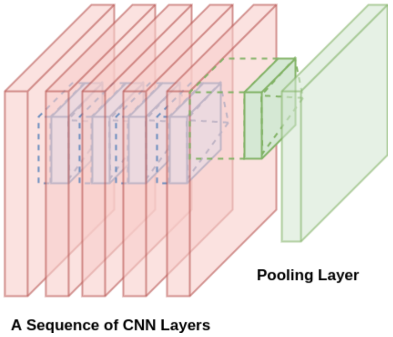

Figure 5.3: The Sequence-CNN-pooling structure

Convolutional sequence-to-sequence model [16]. As shown in Figure 5.3, we apply a stack of five convolution layers followed by a global max pooling layer on top of the word and POS-tag encodings to extract underlying patterns in the sentence. The convolution layers have 512 filters of size 3. We use gated linear units (GLU) [11] as the activation function, and we included residual connections to reduce training difficulty [20]. The output is denoted as $\mathbf{h}_{\text{phrase}} = \mathbf{CNN}(\mathbf{p})$ which contains the meaning of every phrases consist of consequent seven words.

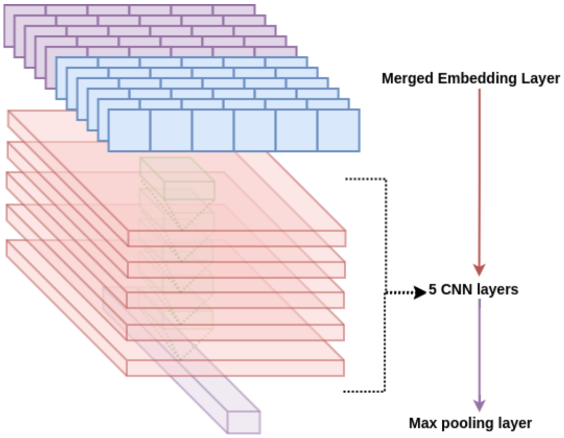

Figure 5.4: Embedding layer to Sequence-CNN-pooling layers

5.1.2 LSTM decoder:

Next, as shown in Figure 5.5, $\mathbf{h}_{\text{phrase}}$ is set as the initial state of an LSTM-decoder. The embedded sentence $\mathbf{s}^{\mathbf{e}}$ is then fed into the decoder one word at a time. We denote the output of the decoder as $\mathbf{h}_{\text{decoded}} = \{h_1, \ldots, h_m\}$ where h_i is the output at timestep i.

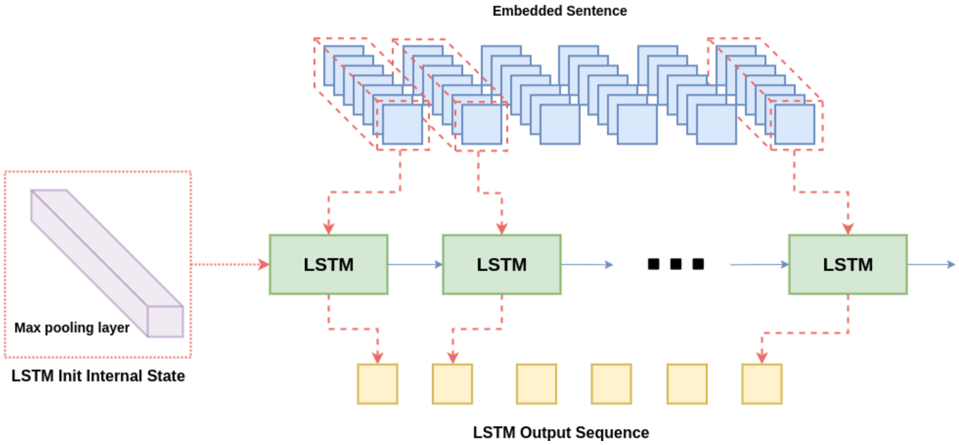

Figure 5.5: LSTM decoder layers

5.1.3 Prediction:

Finally, $\mathbf{h}_{\text{decoded}}$ is fed to a time-distributed fully connected layer, which means each h_i is fed to a fully connected layer $\mathcal{FC}_i$. Each fully connected layer has 3 output units. We denote the final output of a time-distributed fully connected layer as $\mathbf{d} = \{d_1, \ldots, d_n\}$, where d_i indicates the probability distribution over the three types of word s_i.

5.1.4 Loss Function:

To train this model, we apply cross entropy loss. We denote the real distribution of each word in the sentence as $\mathbf{y} = (y_1, \ldots, y_m)$, where $y_i \in \mathbb{R}^3$. The loss is as follows:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{f_{\text{class}}} y_{ij} \log(d_{ij})$$

5.2 Pronoun Resolution Model

The Pronoun Resolution model is to estimate the ”appropriateness” of inserting a noun phrase into the orginal sentence. For instance, the score of “*[slot] compete freely with each other*“ and “*companies*“ is 4.0449, while the score of the same sentencec and “*materials*“ is -0.0021. Similarly to the detection model, given a sentence $\mathbf{s^{marked}} = \left\{s_1^{marked}, \ldots, s_n^{marked}\right\}$ with marked slots where noun phrases are omitted or replaced, where s_i^{marked} could either be a word or a marked slot, its corresponding POS-tags $\mathbf{t^{marked}} = \left\{t_1^{marked}, \ldots, t_n^{marked}\right\}$, and a candidate noun phrase $\mathbf{n} = \{n_1, \ldots, n_k\}$. we formulate the model as follows:

$$g = \mathcal{F}(\mathbf{s^{marked}}, \mathbf{t^{marked}}, \mathbf{n})$$

where g is the score of appropriateness.

First, $\mathbf{s^{marked}}$ and $\mathbf{t^{marked}}$ are fed into the **Word and POS-tag encoding** and the **Sequence-CNN-pooling** components in Section 5.1. We denote the output as $\mathbf{p^{marked}_{s,t}}$. Next we feed the 200-dimensional word embeddings of $\mathbf{n}$ to the **Sequence-CNN-pooling** component and denote the output as $\mathbf{p_n}$. Finally, $\mathbf{p^{marked}_{s,t}}$ and $\mathbf{p_n}$ are concated and fed to a multilayer perceptron (MLP), which consists of two hidden layers, each with 128 rectified linear unit [32]. The output $g = \mathcal{F}(\mathbf{s}, \mathbf{n})$ is a scalar score for the match between the input sentence and noun phrase.

Loss Function: To train this model we apply hinge loss. During training for each sentence $\mathbf{s}$ we know the correct corresponding noun phrase $\mathbf{n}^{+}$; meanwhile we random sample a noun phrase $\mathbf{n}^{-}$ as the negative sample. Then the hinge loss is defined as:

$$\mathcal{L}_{\text{hinge}} = \max\{0, \Delta + \mathcal{F}(\mathbf{s}^{\mathbf{marked}}, \mathbf{t}^{\mathbf{marked}}, \mathbf{n}^{-}) - \mathcal{F}(\mathbf{s}^{\mathbf{marked}}, \mathbf{t}^{\mathbf{marked}}, \mathbf{n}^{+})\}$$

By optimizing the parameters to minimize $\mathcal{L}$hinge, we encourage the model to learn that $\mathcal{F}(\mathbf{s}^{\mathbf{marked}}, \mathbf{t}^{\mathbf{marked}}, \mathbf{n}^{+})$ should be at least $\mathcal{F}(\mathbf{s}^{\mathbf{marked}}, \mathbf{t}^{\mathbf{marked}}, \mathbf{n}^{-})$ plus a margin Δ.

5.3 The End-to-end Model

Based on the above two individual models, we further propose an end-to-end model where the detections and resolutions are done simultaneously:

$$[\{d_1, \ldots, d_n\}, g] = \mathcal{F}(\mathbf{s}, \mathbf{t})$$

The idea is straight-forward. $\mathbf{s}$ and $\mathbf{t}$ are processed by **Word and POS-tag encoding**, **Sequence-CNN-pooling**, **LSTM encoder-decoder** and **Prediction** components the same as in Section 5.1 to generate the output distributions $\mathbf{d}$. The output of **Sequence-CNN-pooling** during this step is denoted as $\mathbf{p_{s,t}}$. Then, $\mathbf{n}$ is processed the same as in Section 5.2. Following the same procesure in Section 5.2 only replacing $\mathbf{p}_{\mathbf{s,t}}^{\mathbf{marked}}$ with $\mathbf{p_{s,t}}$, the score g is generated.

5.3.1 Loss Function

To train this model, we calculate the hinge loss $\mathcal{L}_{\text{hinge}}$ and the cross entropy loss $\mathcal{L}_{\text{cross-entropy}}$ samely as above. The two losses are added up with a coefficient for balancing:

$$\mathcal{L} = \mathcal{L}_{\text{hinge}} + f_{\text{balance}} * \mathcal{L}_{\text{cross-entropy}}$$

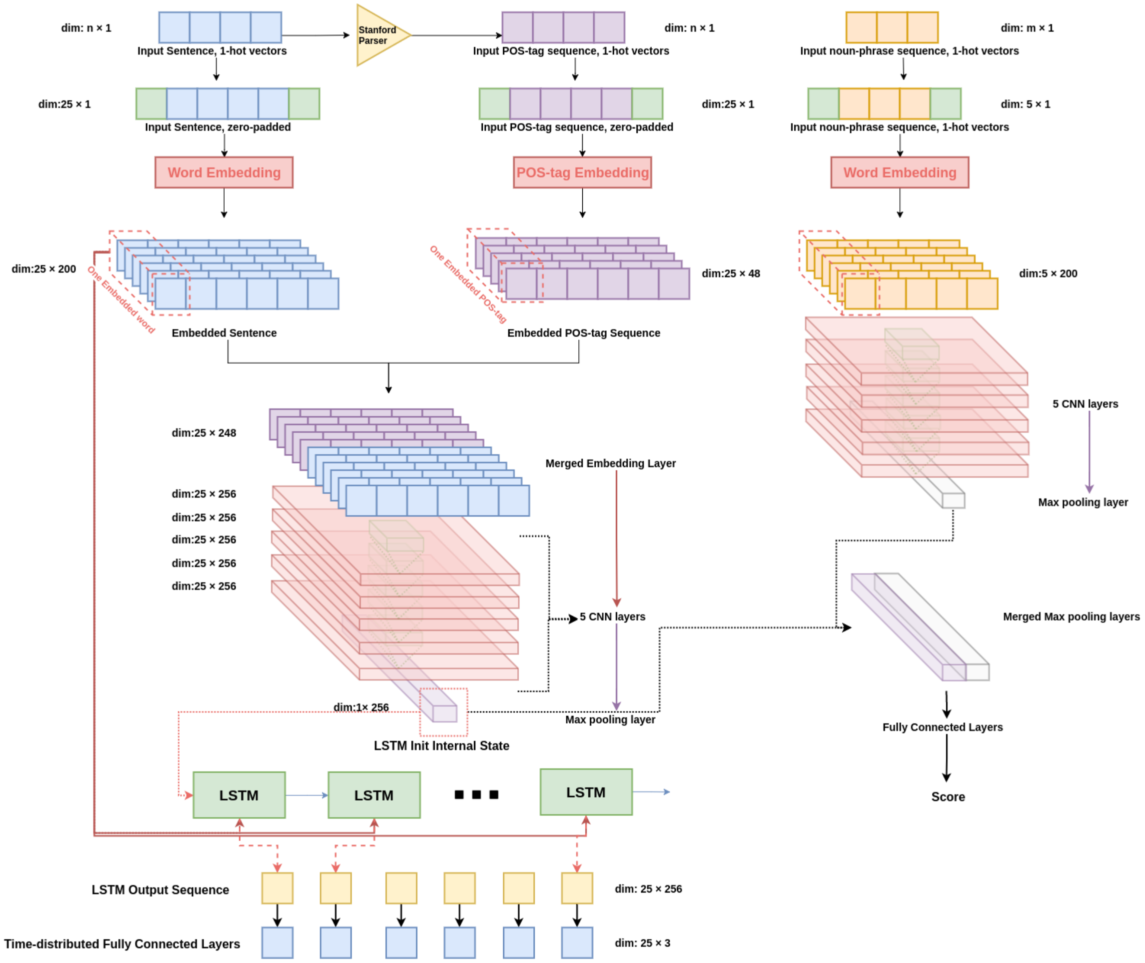

Figure 5.6: End to end Resolution Model Architecture

Chapter 6

Experiments

6.1 Datasets And Formats

All of our models are language independent. We ran experiments on Chinese datasets. Our Chinese dataset is made from data collected from Chinese cQA websites including BaiduZhidao, SosoWenwen, which contains over 300,000,000 QA pairs. We generated *time*, *location*, *people* and *noun phrase* examples according to the breakdown in Table 6.1. Because Chinese language is graphically based, Jieba [41], a Chinese segmentor which segments a sentence into a sequence of words, is applied on each sentence prior to our data generation.

Each dataset is divided into two parts, the training data and the testing data, at the ratio of 9:1; the testing data is completely out-sampled from the training data. The Coreference and Ellipsis Detection Model is trained and tested on all four datasets. The Pronoun Resolution Model and End-to-end Model are tested on the noun phrases dataset.

	Dataset	**Negative**	**Ellipsis**	**Coreference**	**Total**
Chinese	Noun phrase	1 000 000	800 000	1,200 000	3,000 000
	Location	1 000 000	200 000	750 000	1 950 000
	People	1 000 000	990 000	601 000	1 700 000
	Time	750 000	20 0000	500 000	1 450 000

Table 6.1: Number of sentences for each dataset in this paper.

6.2 Implementing Details

The Chinese word embeddings are pre-trained using word2vec [31], with the raw data of our corpus. The models are trained by the Adam algorithm [23] and with a learning rate of 3×10^{-4}.

6.3 Results

6.3.1 Co-reference And Ellipsis Detection Model

Sentences as long as POS-tag sequences are fed to the model. The model outputs the positions of co-reference and ellipsis if they exist. More specifically, the model predicts the probability distributions for each word in the sentence, and the word is then classified to the type with the highest probability. We denote the output as $\mathbf{d}$, where $\mathbf{d} = \{d_1, \ldots, d_m\}$, the real type as $\mathbf{l} = \{l_1, \ldots, l_m\}$. Under this setting, we define the follows:

- true positive data point (TP): $\forall i = 1, \ldots, m$, st. $d_i = l_i$ and $\exists i = 1, \ldots, m$ st. $l_i = 1 \vee l_i = 2$.
- true negative data point (TN): $\forall i = 1, \ldots, m$, st. $d_i = l_i$ and $\forall i = 1, \ldots, m$ st. $l_i = 0$.
- false positive data point (FP): $\exists i = 1, \ldots, m$ st. $d_i \neq l_i$ and $\exists i = 1, \ldots m$ st. $d_i = 1 \vee d_i = 2$.
- false negative data point (FN): $\exists i = 1, \ldots, m$ st. $d_i \neq l_i$ and $\forall i = 1, \ldots, m$ st. $l_i = 0$.

The accuracy, precision, and recall rate are then calculated accordingly. The experiment results are in Table 6.2.

	Dataset	**Accuracy**	**Precision**	**Recall**
Chinese	Noun phrase	93.2%	92.7%	96.9%
	Location	95.6%	95.3%	95.7%
	People	96.1%	92.9%	97.5%
	Time	93.8%	91.1%	95.7%

Table 6.2: Accuracy, Recall and Precision Rates of Co-reference and Ellipsis Detection models

The high accuracies indicate the strong ability to distinguish positive examples and negative examples. The slightly higher recall rate than precision indicates the model tends to treat potentially words as positive and retrieve more potentially positive candidates, which meets our requirements in this field properly.

6.3.2 Pronoun resolution model

The model is trained and tested on noun phrase datasets. The model is not tested on time, people, and location datasets because the resolutions in these datasets are straightforward. The corresponding named entity could be retrieved easily from the context according to its specific type (time, people or location). On the contrary, noun phrases are ambiguous and difficult to resolve with simple mechanisms.

We evaluate the model's performance on the test set. For each sentence in the test set, we feed it into the model together with the correct noun phrase and nine randomly sampled noun phrases. The model outputs the 'appropriateness' scores for all 10 noun phrases and we choose the one with the highest score as the model's prediction. Under this setting, a naive model that outputs random scores will have an overall accuracy close to 10%. Our Pronoun Resolution Model's (PRM) performance is listed in Table 6.3.

Dataset	**Model**	$K = 1$	$K = 2$	$K = 3$
Chinese	PRM	67.2%	84.8%	91.2%
	PRM with _PAD	69.1%	85.2%	91.2%
	Bigram	22.8%	37.1%	48.2%

Table 6.3: Top K accuracies of Pronoun Resolution models

When integrating the PRM with Reference Identifying Model(RIM), we find that sometimes RIM predicts a word in a sentence to be a reference when it is not. This requires our PRM to have the ability to predict that nothing fits for a _PRONOUN slot. To achieve that we create a special token _PAD, representing the null string. Then we modify the hinge loss to be:

$$\begin{aligned}\mathcal{L}_{\text{hinge}} = \max\{0, \Delta + \mathcal{F}(\mathbf{s}, \mathbf{n}^-) - \mathcal{F}(\mathbf{s}, \mathbf{n}_0)\} + \\ \max\{0, \Delta + \mathcal{F}(\mathbf{s}, \mathbf{n}_0) - \mathcal{F}(\mathbf{s}, \mathbf{n}^+)\}\end{aligned}$$

Where $\mathbf{n}_0$ represents the word embedding for _PAD. At inference time we can input _PAD along with other candidate noun phrases to PRM. If _PAD token has the highest score, that means nothing should be fit into the reference slot given by RIM. We trained PRM again with the aforementioned modifications on the same training data set. Surprisingly, we find that the top 1 accuracy on the test set improves by 1.86% and we cannot yet justify why. For comparison, we experimented ranking the candidates based on bigram frequency. The result is shown in Table 6.3.

6.3.3 End-to-end Model

The end-to-end model is also tested only on the noun phrase dataset for the same reason as above. This model is trained with the original sentence as well as the correct noun phrase and 9 random sampled noun phrases. As for outputs, it has two parts, the co-reference and ellipsis detection of the sentence, and the score of 'appropriateness'. The experiment results of the end-to-end model are shown in Table 6.4.

Dataset	**Accuracy**	**Precision**	**Recall**	K**=1**	K**=2**	K**=3**
Chinese	93.8%	95.5%	95.3%	70.1%	82.9%	89.0%

Table 6.4: Accuracy, precision and recall rate, and top K resolution accuracies of the end-to-end model

Comparing the end-to-end model with the detection model, for the Chinese dataset, we found that the end-to-end model has improvements of 0.6% on accuracy and 2.8% on precision. The recall rate has dropped by 1.6%. The result shows that involving candidate phrase information, the ability of detecting the correct co-reference and ellipsis is improved.

Comparing to end-to-end model with the PRM, for Chinese dataset, we found that the top 1 accuracy is slightly improved by 2.9%, while top 2 and top 3 accuracies are dropped by 1.9% and 1.2%. The drops are expected as the position information of co-reference and ellipsis are not given. The accuracy of resolution is based on the accuracy of detection in the end-to-end model.

6.3.4 Selected Results

Table 6.5 presents several conversations resolution examples by our models. Symbols in the sentences have the following meanings. △ indicates eclipses. Blue words indicate co-references followed by the type of co-referencing in square brackets.

Detection	**A**:加拿大总理是谁? (Who is the prime minister of Canada?) **B**: 贾斯汀鲁铎. (Justin Trudeau.) **A**:他[People]老爸是谁? (Who is his[People] father?)
Resolution	贾斯汀鲁铎老爸是谁? (Who is Justin Trudeau’s father?)
Detection	**A**: 绵中离我们近吗, 我朋友去那里[Location]上学. (How far is the Mian High school? My friend is studying there[Location].) **B**: 打车过去△[Location]十分钟 (Ten munites cab △[Location].) **A**: 打车钱他[People]报销我就去△[Location] (I will go △[Location] if he[People] reimbursed me the cab fee.)
Resolution	**A**: 绵中离我们近吗, 我朋友去绵中复读 (How far is the Mian High school? My friend is studying in Mian High school.) **B**:打车过去绵中十分钟. (Ten munites cab to Mian High school.) **A**:打车钱我朋友报销我就去绵中. (I will go to Mian High school if my friend reimbursed me the cab fee.)
Detection	**A**:麦当劳的薯条是什么做的?为什么△[Noun phrase]凉了吃会涩呢? (What are the ingredients in McDonalds’ fries? Why △[Noun phrase] taste bitter when cooling down?)
Resolution	为什么麦当劳的薯条凉了吃会涩呢? (Why McDonalds’ fries taste bitter when cooling down?)
Detection	**A**:法国人喜欢吃生一点的牛排. (The French prefer steaks rare.) **B**:△[People]为什么不吃熟的△[Noun phrase] (△[People]Why not eat well-done ones△[Noun phrase]?
Resolution	法国人为什么不吃熟的牛排? Why not the French like well-done steaks?

Table 6.5: Selected Examples

Chapter 7

Conclusion and Discussion

Currently, most chatting robots depend on grammar rules to handle multiple-round conversation for special cases. Many people even think the multiple-round conversation can only be implemented in vertical domains by such special custom rules. We have argued that a multi-round conversation can be decomposed into single-round conversations together with context resolution.

We have systematically defined the context resolution problem for conversation and initiated a comprehensive study of this problem. We have demonstrated how to create training data to train an end-to-end deep learning network to solve our problem with high accuracy, for a large sub-class of our problem.

This study leads to many open studies. Our work could be extended to wider contextual domains, including more conjunctive relations and more careful linguistic studies of conjunctive relations in conversations. Our work could be extended to wider language domains, since our framework is language independent. Studies could go beyond context resolution and include semantics from conversation history. At the application level, an end-to-end Question-Answering system could be formed by combining our work with QA works, for example, that of [8]. Moreover, beyond context-resolution, multi-round chatting by a chatbot certainly involves many other aspects, such as avoiding conversation-ending topics [29] and consistency [36]. Eventually all these should be integrated to an end-to-end system with the context resolution models.

References

[1] https://www.thefreedictionary.com/conjunctively.

[2] The pronoun resolution model is proposed by rui qiao. the work of his is not published.

[3] Rsvp dodo, the first chatting robot with rsvp brain in the world. http://dodo.rsvp.ai/, 2017.

[4] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Shuanhu Bai, Horng Jyh Paul Wu, Haizhou Li, and Gareth Loudon. System for chinese tokenization and named entity recognition, October 30 2001. US Patent 6,311,152.

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[8] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[9] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[11] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[13] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[14] Jenny Rose Finkel and Christopher D Manning. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334. Association for Computational Linguistics, 2009.

[15] Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574, 2005.

[16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[17] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[18] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[19] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[22] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[23] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[26] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

[29] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[30] Jun Li, Jinxian Pan, Chen Ye, Yong Huang, Danlu Wen, and Zhichun Wang. Linking entities in chinese queries to knowledge graph. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 590–597. Springer, 2015.

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[32] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[33] Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In *EMNLP*, pages 548–554, 2015.

[34] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[35] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics, 2011.

[36] Qiao Qian, Minlie Huang, and Xiaoyan Zhu. Assigning personality/identity to a chatting machine for coherent conversation generation. *arXiv preprint arXiv:1706.02861*, 2017.

[37] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[38] Christopher D. Manning Roger Levy. Is it harder to parse chinese, or the chinese treebank? pages 439–446. Association for Computational Linguistics, 2003.

[39] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784, 2016.

[40] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

[41] Junyi Sun. Jieba chinese text segmentation module. `https://github.com/fxsjy/jieba`, 2012.

[42] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[43] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[44] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

[45] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[46] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[47] Jheng-Long Wu and Wei-Yun Ma. A deep learning framework for coreference resolution based on convolutional neural network. In *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*, pages 61–64. IEEE, 2017.

[48] Qing Xia, Xin Yan, Zhengtao Yu, and Shengxiang Gao. Research on the extraction of wikipedia-based chinese-khmer named entity equivalents. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 372–379. Springer, 2015.

[49] Borui Ye. Query similarity for community question answering system based on recurrent encoder decoder. 2017.

[50] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics, 2002.

APPENDICES

.1 Training Data Creation Process

In this section, we will introduce the procesure we follow to create our training data.

.1.1 Word labeling

Initially, we have collected over 300,000,000 Question-Answer pairs from Chinese cQA websites including BaiduZhidao[1], SosoWenwen[2] and others. All our training data are created out of these collected sentences. The following steps are applied.

- Sentences of length longer than 25 words are removed. The leftover sentences usually contain 6-7 words in our new data set D.
- 9 types of sentences, to be defined in Section .1.2, are then selected from D, for each of the following 4 datasets. Of these 9 types, types 1,2,9 do not contain labeled words, types 1-4 will serve as negative examples and type 2 is essential "everything else in D" and thus is trimmed to the current sizes in Table 2. Type 9 contains sentences that already have co-referents themselves. Types 5-8 will be used to generate positive examples (by replacing a word as a case of co-reference or by removing a word as a case of eclipse or conjunction). All other types, except for type 2, keep their original sizes.
  - Time dataset: we denote the time-words as "labeled words".
  - Location dataset: we denote the location-words as the "labeled words".

[1] https://zhidao.baidu.com/
[2] https://wenwen.sogou.com/

  - People dataset: we denote the people-words as the “labeled words”.
  - Noun phrase dataset: we denote the noun phrases as the “labeled words”.

The final data set contains 8003752 sentences, detailed in Table 2. The sizes of above 4 datasets are also given in Table 2. Note that there exist some overlaps among these four datasets: if a sentence S contains a time-word and a location-word then S belongs to both Time dataset and Location dataset; however, in the Time dataset, the time-word in S is the only labeled word, and in the Location dataset, the location-word S is the only labeled word. Ditto for People and Noun phrase datasets.

.1.2 Word replacement

At the beginning of Section 4, we have replaced “Trudeau” by “he” to show how to construct the training data. Other replacement rules are tedious, although not beyond control, and they will be defined in this section. As these are Chinese language-specific details and using the same idea, a non-Chinese reader can safely skip the rest of this section.

We now explain how to replace the entities we found from the previous step. We divide the sentences into nine different types in each of the 4 domains: time, people, location, noun phrase. In order to define these nine type of sentences, we first, need to introduce our Reference Word (RW) lists for each the four datasets.

Let X=Noun phrase, Location, People, or Time. Consider a sentence S_X in domain X. RW_4^X includes a complete list of co-reference words for in S_X. RW_3^X contains the list of pronouns or similar we will use to replace corresponding entities in S_X. When a word in RW_1^X appears before a labeled word or a word in RW_2^X appears after a labeled word, then they are not a real pronoun and do not serve as a co-referent, in S_X.

- **Noun phrase:**
  - RW_1^N, RW_2^N, RW_3^N, RW_4^N = {”这” (this, it),”那” (that),”这个” (this one), ”那个” (that one), ”这些” (these),”那些” (those) }
- **Location:**
  - RW_1^L, RW_2^L = {”这” (here),”那” (there),”这儿” (here),”那儿” (there), ”这个” (here),”那个” (there), ”这里” (here), ”那里” (there), ”这边” (here), ”那边” (there)},

- RW_3^L = {"这" (here),"那" (there), "这个" (here), "那个" (there) },
- RW_4^L = {"这儿" (here), "那儿" (there), "这里" (here), "那里" (there), "这边" (here), "那边" (there), "这地方" (here), "那地方" (there), "这个地方" (here), "那个地方" (there)}

- **People:**
  - RW_1^P = {"这" (this),"那" (that), "这个" (this one), "那个" (that one), "这些" (these), "那些" (those)},
  - RW_2^P = {"她" (she, her,), "他" (he, him, his), "它" (it, its), "他们" (they, their), "她们" (they, their), "它们" (they, (their)},
  - RW_3^P = {"这" (this), "那" (that), "这个" (this one), "那个" (that one), "这些" (these),"那些" (those)},
  - RW_4^P = {"她" (she), "他" (he), "他们" (they, them), "她们" (they, them), "这个人" (this person), "那个人" (that person), "这些人" (these people), "那些人" (those people), "这人" (this person), "那人" (that person), "那谁" (that guy)}

- **Time:**
  - RW_1^T, RW_2^T = {"这" (this time point, it), "那" (that time point, then), "这个" (this time point),"那个" (that time point), "这时" (this time), "那时" (that time, then),"这时候" (this time), "那时候" (that time), "这天" (this day), "那天" (that day), "那年" (that year)},
  - RW_3^T, RW_4^T = {"这时" (this time), "那时" (that time, then), "这时候" (this time), "那时候" (that time), "这天" (this day), "那天" (that day), "这个时候" (this moment), "那个时候" (that moment), "那年" (that year)}

These four sets of reference words provide the flexibility of modification and prevent generating incomplete sentences. Next, as shown in Figure 1 and Figure 3, in each of the four domains $X \in = \{$Time, People, Location, Noun Phrase $\}$, we search for the sentences of nine types according to the following rules.

The sentence, S^X for domain X is considered a sequence of words:

$$S = \{w_1, w_2..., w_n\}$$

where we will use a $*$ to denote a labeled word, $w*$. We use w^l to denote the word left to word w, and w^r the word on the right. Noun denotes the set of all nouns. Conj denotes

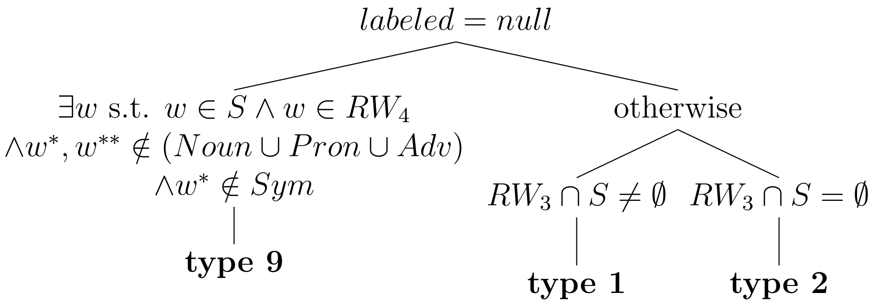
$labeled = null$
$\exists w$ s.t. $w \in S \land w \in RW_4$
$\land w^*, w^{**} \notin (Noun \cup Pron \cup Adv)$
$\land w^* \notin Sym$
type 9
otherwise
$RW_3 \cap S \neq \emptyset$
$RW_3 \cap S = \emptyset$
type 1
type 2

Figure 1: Classifications for Sentences Contain No Labeled Words

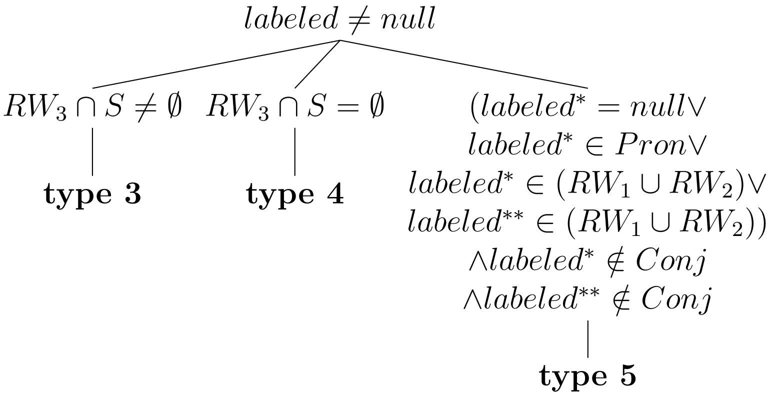
$labeled \neq null$
$RW_3 \cap S \neq \emptyset$
$RW_3 \cap S = \emptyset$
$(labeled^* = null \lor$
$labeled^* \in Pron \lor$
$labeled^* \in (RW_1 \cup RW_2) \lor$
$labeled^{**} \in (RW_1 \cup RW_2))$
$\land labeled^* \notin Conj$
$\land labeled^{**} \notin Conj$
type 3
type 4
type 5

Figure 2: Classifications for Sentences Contain Labeled Words

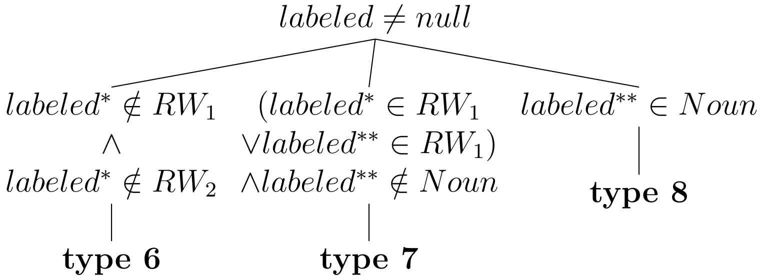

Figure 3: Classifications for Sentences Contain Labeled Words Continued

the set of all conjunction words. Pron denotes the set of all pronouns. Adv denotes the set of all adverbs. Sym denotes the set of all punctuations.

We present examples for each type in Table 1, where special words (e.g. labeled words) are in special colors and followed by its label in brackets.

Type	**Dataset**	**Sentence and Translation**
Type 1	Noun Phrase	我都醒了你这(RW_3^N)是还没睡呐. I woke up already, it seems like you have not slept yet.
	Time	那(RW_3^T)你毕业了想做什么呢 Then what is your plan after graduation.
Type 2	People	中国自古以来谁的武功最高啊? Who is the top martial artist of all times in China?
Type 3	People	那(RW_3^P)小编肯定不是东北 银(labeled words). So, that the editor must not be a northeastern folk.
Type 4	Time	狗狗一年四季(labeled word)都伸舌头是为什么 Why do dogs stick their tongues out all year round?
Type 5	Location	这(RW_1^L)北京(labeled word)的霾被你赶上了哈哈. This Beijing fog caught you!
	Noun Phrase	葡萄酒(labeled word)放久了还能再喝吗? Is wine stored a long time okay to drink?
Type 6	Location	西直门(labeled word)有啥好吃的吗? Any great food around Xizhimen?
Type 7	People	那个(RW_1^P)小罗伯特唐尼非常帅! That Robert Downey Jr is very handsome.
Type 8	People	杜莎夫人(labeled word)蜡像馆(noun)有学生票吗? Does Madame Tussauds wax museum have student tickets?
Type 9	Noun Phrase	因为那(RW_4^N)是大年夜. Because that was New Year's Eve.

Table 1: Examples of sentences and translations with labeled words (red) and other special words (blue) of nine types

Sentences are divided into the above nine types according to their syntaxes and semantics. Different modification rules are applied to different types in order to ensure that the modified sentences are proper sentences.

Sentences belong to **types 1-4**, are complete sentences and have complete semantic meanings. Even though sentences from **type 3** and **type 4** contain reference words, these words do not have co-reference funtionality. The first example in Table 1, the word 这 **(this)** in 我都醒了你[这][RW_3]是还没睡呐(I woke up already, it seems like you have not slept yet.) does not refer to any context.

These complete sentences are divided into the above four types in order to balance the data ratios. These sentences give our model the ability to distinguish reference words that have co-reference functionality from those not.

In sentences belong to **type 5**, labeled words could be omitted and the leftover parts of these sentences still make sense. In the 6th example in Table 1, when **北京** Beijing is removed from [**这**][RW_1][**北京**][labeled word]的霾被你赶上了哈哈(This Beijing fog caught you!), the leftover part, [**这**][RW_1]的霾被你赶上了哈哈(The fog here caught you!) is still a good sentence and this is clearly an example of eclipse. These sentences make good data for the eclipse detection and some conjunctive relations.

In sentences belong to **type 6** and **type 7**, labeled words could be replaced by reference words and the modified sentences still make sense. In the 8th example in Table 1, [**西直门(Xizhimen)**] in [**西直门**]有啥好吃的吗? (Any great food around Xizhimen?) could be replaced by [那儿](there) and 那儿有啥好吃的吗(Any great food there) is a typical co-reference case. While in **type 8**, the labeled words are used as adjectives and could be replaced by reference words followed by a ”的” (of/’s). For example, [**杜莎 夫人**(Madame Tussauds)] in [**杜莎 夫人**][labeled word][**蜡像馆**][noun]有学生票吗(Does Madame Tussauds wax museum have student tickets?). could be replaced by [她(she)] followed by a ”的(’s)”[3]. 她的[**蜡像馆**][noun]有学生票吗 (Does her wax museum have student tickets?) is well-form sentence with co-reference. These sentences are good data for co-references.

Sentences belong to **type 9** contain co-references already.

Finally, we edit each sentence according to the following rules.

- **type 1, 2, 3, 4**: No editing.
- **type 5**: Delete the labeled words. Label the position of the deleted words.
- **type 6**: Replace the labeled words with words from RW_3, and label the position of the replacement.
- **type 7**: Replace the labeled words with words from RW_4, and label the position of the replacement.
- **type 8**: Replace the labeled words with words from RW_3 or RW_4 and add ”的” right after, and label the position of the replacement.
- **type 9**: No editing. Label the position of words from RW_4.

The data sizes of each type are listed in Table2.

[3] In Chinese, [她(she)的(’s)] means *her*.

	Noun phrase	**Location**	**People**	**Time**
type 1	10	52	53	0
type 2	25	35	40	59.9
type 3	45	3	2	0.1
type 4	20	10	5	15
type 5	80	20	9.9	20
type 6	116.5	62.4	20	47.5
type 7	0	0	28	0
type 8	0.5	9	2	2
type 9	3	3.6	10.1	0.5

Table 2: Data size of each type, unit: 10000