Reference Understanding in Dialogs with Contexts

A three-level Deep Neural Structure

Junnan Chen

Deep Learning Course Project

Introduction - Why Important

Dialog understanding has become more and more important in natural language processing area with the rise of chatting machines.

Understanding a single sentence can not fulfill the functionality of a chatting machines for the reason that reusing information from the previous contexts makes it very hard for machines to understand the real intention of the current sentence.

- -I went to dinner at Jim's last night.
- -Is **it** delicious?

The word *it* in the second sentence could confuse the chatting machine if given without the context.

The Trend of Chatting AI

What are the big guys doing?

- October 2011, Apple: Hi Siri!
- May 2012, Samsung: S Voice.
- April 2013, Microsoft: Cortana.
- November 2014, Amazon: Alexa.
- May 2015, Google: Okay Google!
- March 23, 2016. 16-hour Tay.
- April 2017, Samsung: Bixby.

Artificial intelligence chatterbot, intelligent personal assistant, virtual personal assistant.

Introduction - Why Difficult

Robots need to think like a human in order to talk like a human.

- Knowledge-based: Who is the half blood prince?
- Context-based: I like him as well. What about Waterloo?
- Informal language usage: Drop me a line!

Introduction - Why Difficult

Referring: One type? Multiple types? Not referring?

- -I went to dinner at Jim’s last night.
- -Is **it** delicious?

The word *it* could be referring many types of nouns or not referring to anything at all.

Omitting: entities that was mentioned in the previous context.

- -What song did she sing yesterday?
- -She sang her sweeteast.
- -Will Tom go to the party?
- -Unless invited.

Introduction -Why Difficult

Lack of Training Data

Traditionally Speaking: Adequate well labeled sentences, where words that refers to some entity, positions where entities are omitted and the types of these entities are labeled is defined as a labeled sentence. Not published yet according to our knowledge.

Another idea: Unsupervised learning based on plenty of conversations. Not practical.

The Reference Understanding Task

Based on the above facts, we define the reference understanding task as followed.

Given one sentence from a dialog, we are to detect

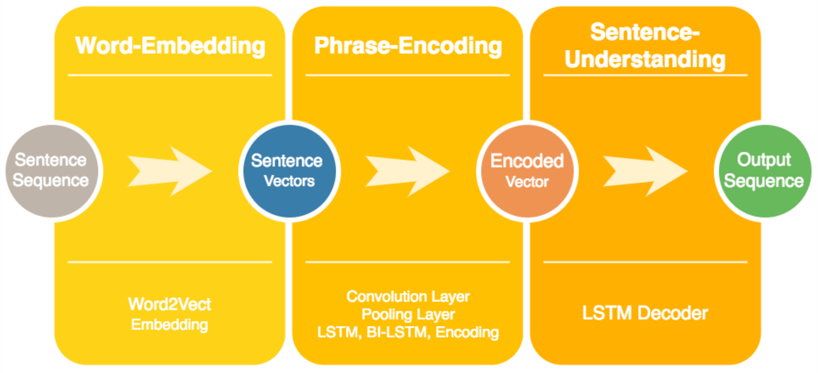
- words that are referring to some entity from the previous contexts and figure out the type of the entity.
- position of omitted entity from the previous sentences in the current sentence and figure out the type of the entity.

The Key to Tackle The Problem

Reference understanding requires a high level of sentence understanding due to the fact that, in most cases, subjects and objects are replaced or omitted.

- 3-level hierarchical neural network structure to simulate how human brain works.
- Combination of 4 types of DNN assembled according to their features.

The Model Overview



- **Word level**: Extract the meaning of each individual word.
- Phrase level: Construct phrases and extract meaningful ones.
- Sentence level: Classify and Recognize references based on the understanding of the entire sentence.

Word-Embedding Level - Classic neural language model

The classic neural language model proposed by Bengio et al consists of a one-hidden layer feed-forward neural network that predicts the next word in a sequence.

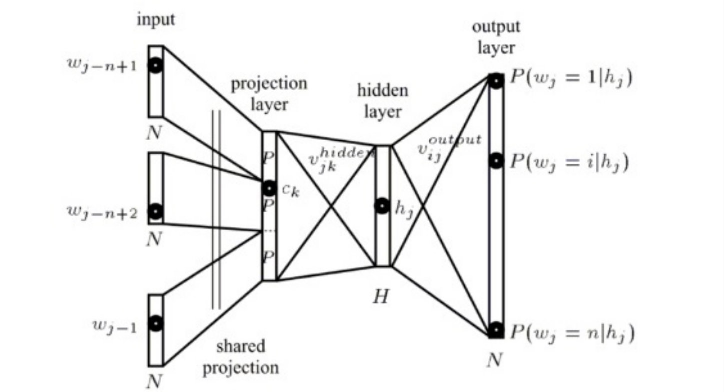


Figure 1: Bengio et al. a neural language model.2006

Word-Embedding Level - Word2Vec

In the embedding layer, each word in the sentence will be projected into a space with lower dimension.

- Provides general semantic relationships
- Less computation costs

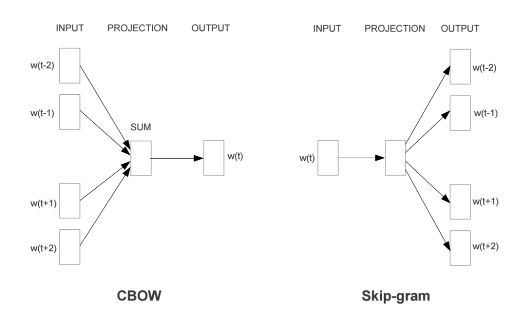


Figure 2: Mikolov et al, Efficient Estimation of Word Representations in Vector Space. 2013.

Word-Embedding Level

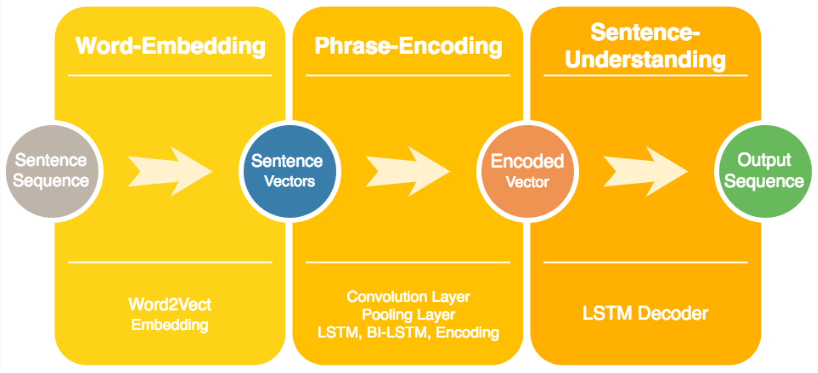
Input Sequence: $X = (X_1, .., X_n)$, $X \in \mathbb{R}^{n \times d_{vocabulary}}$

Embedding: A matrix $M_{embedding} \in \mathbb{R}^{d_{vocabulary} \times d_{embedding}}$.

A word, X_i: one point from the a space with dimension of the entire vocabulary size, $d_{vocabulary}$.

Embedding layer: $\mathrm{E}(X) = X * M_{embedding}$.

The Model Overview



- Word level: Extract the meaning of each individual word.
- **Phrase level**: Construct phrases and extract meaningful ones.
- Sentence level: Classify and Recognize references based on the understanding of the entire sentence.

The Phrase Understanding Level

Phrases consist of consequent words or discontinuous words.

- a bunch of, good game, have a nice day.
- have not … yet, if … then.

Different phrases can share similar meaning.

- He weighs more than I do.
- He is heavier than me.

Phrase could be in many positions in a sentence.

- I will eat when I feel hungry.
- When I feel hungry I will eat.

Phrases are patterns in sentences. Understanding phrases is similar to detecting beaks in pictures.

The Phrase Understanding Level - Finding Patterns

Convolution Neural Networks (CNN) combined with Pooling has been proved successful in image classification (krizhevsky2012imagenet), face recognition (lawrence1997face), sentence classification (kim2014convolutional) and many other areas.

There are a certain number of filters of certain sizes in CNN. Each filter goes through the entire input to detect a certain pattern. CNN has good performance on finding patterns while shrink the size of parameters resulting from sharing filters. There are plenty of patterns exists in natural languages.

we apply CNN+Pooling in our model in order to seek the underlying patterns in the input sentence.

The Phrase Understanding Level - CNN Construction

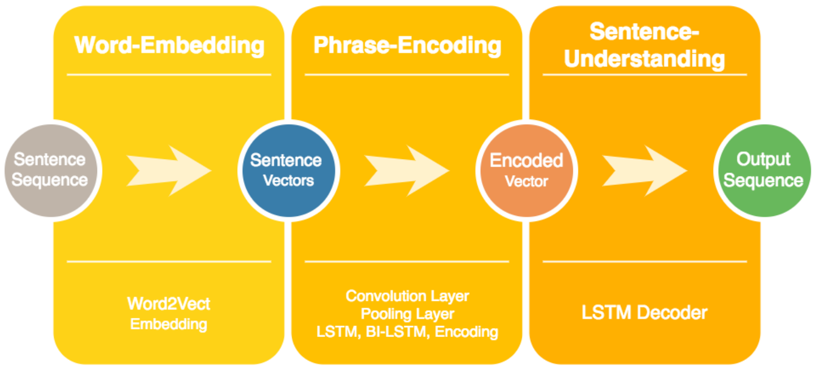
- First CNN layer F_1: f_1 number of filters of size $(3 \times d_{embedding})$
- Second CNN layer: f_2 number of filters of size $(4 \times f_1)$
- Pooling layer: pooling size of 3.

F_1: patterns consists of 3 words, which means every consequent 3-word phrase is represented by a vector with dimension of f_1.

F_2: patterns consists of every consequent 4-phrase is represented by a vector with dimension of f_2.

Not all of F_2 contains useful information. We apply a maxpooling layer to eliminate useless information and compress the phrase information

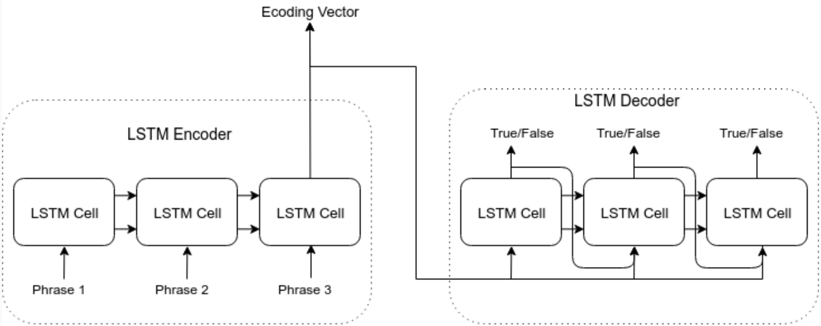
The Model Overview



- Word level: Extract the meaning of each individual word.
- Phrase level: Construct phrases and extract meaningful ones.
- **Sentence level**: Classify and Recognize references based on the understanding of the entire sentence.

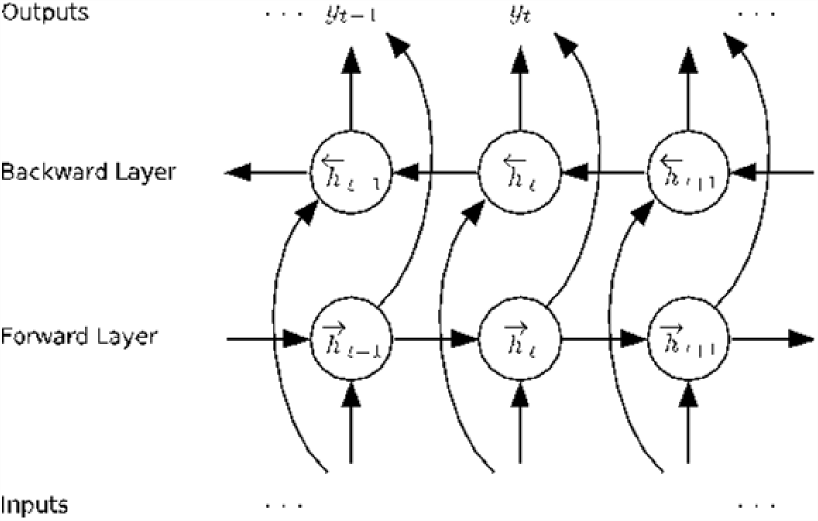
Sentence Understanding Level: LSTM Encoder-decoder

The information of the phrases we selected from the last step needs to be compressed. We take LSTM's advantages of memorizing history information. An encoder-decoder is adapted here.



Sentence Understanding Level: Bi-directional LSTM

The encoder could also be Bidirectional LSTM cells.

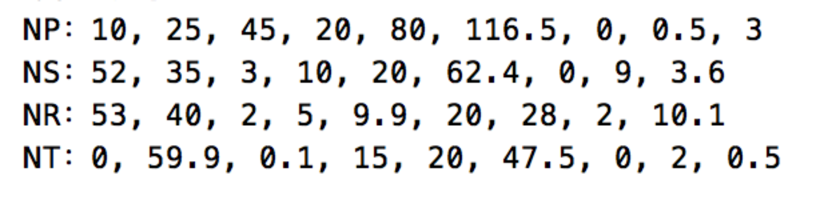


The Objective

A distributed softmax layer follows the LSTM decoder. That is, for each $h'_i \in h'_1, h'_2, ..., h'_n$, there are softmax functions applied, indicating the probability of whether $word_i$ is a pronoun or followed by an omitted entity. Therefore, the objective is to minimize the categorical cross-entropy between the true label and the predicted probability.

Data Source and Data Size

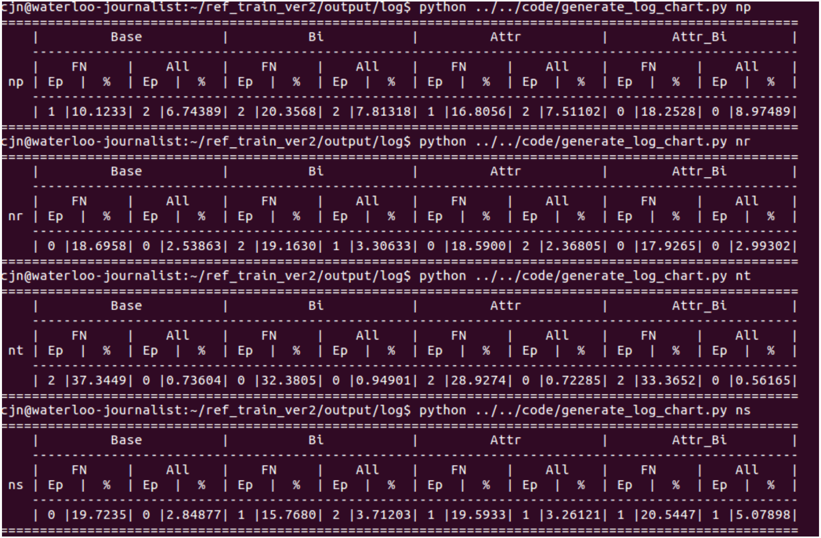
- Data collected from cQA website, 8003752 sentences.
- Data labeling: StanfordParser, nounphrase, time, location, name.
- Entity replacement and deletion.



```
NP: 10, 25, 45, 20, 80, 116.5, 0, 0.5, 3
NS: 52, 35, 3, 10, 20, 62.4, 0, 9, 3.6
NR: 53, 40, 2, 5, 9.9, 20, 28, 2, 10.1
NT: 0, 59.9, 0.1, 15, 20, 47.5, 0, 2, 0.5
```

Figure 3: Data size: unit size is 10,000

Experiments and Results



```
cjn@waterloo-journalist:~/ref_train_ver2/output/log$ python ../../code/generate_log_chart.py np
=====================================================================================================
    |         Base          |          Bi           |         Attr          |        Attr_Bi        |
    -------------------------------------------------------------------------------------------------
    |    FN     |    All    |    FN     |    All    |    FN     |    All    |    FN     |    All    |
 np | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  |
    -------------------------------------------------------------------------------------------------
    | 1 |10.1233| 2 |6.74389| 2 |20.3568| 2 |7.81318| 1 |16.8056| 2 |7.51102| 0 |18.2528| 0 |8.97489|
=====================================================================================================
cjn@waterloo-journalist:~/ref_train_ver2/output/log$ python ../../code/generate_log_chart.py nr
=====================================================================================================
    |         Base          |          Bi           |         Attr          |        Attr_Bi        |
    -------------------------------------------------------------------------------------------------
    |    FN     |    All    |    FN     |    All    |    FN     |    All    |    FN     |    All    |
 nr | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  |
    -------------------------------------------------------------------------------------------------
    | 0 |18.6958| 0 |2.53863| 2 |19.1630| 1 |3.30633| 0 |18.5900| 2 |2.36805| 0 |17.9265| 0 |2.99302|
=====================================================================================================
cjn@waterloo-journalist:~/ref_train_ver2/output/log$ python ../../code/generate_log_chart.py nt
=====================================================================================================
    |         Base          |          Bi           |         Attr          |        Attr_Bi        |
    -------------------------------------------------------------------------------------------------
    |    FN     |    All    |    FN     |    All    |    FN     |    All    |    FN     |    All    |
 nt | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  |
    -------------------------------------------------------------------------------------------------
    | 2 |37.3449| 0 |0.73604| 0 |32.3805| 0 |0.94901| 2 |28.9274| 0 |0.72285| 2 |33.3652| 0 |0.56165|
=====================================================================================================
cjn@waterloo-journalist:~/ref_train_ver2/output/log$ python ../../code/generate_log_chart.py ns
=====================================================================================================
    |         Base          |          Bi           |         Attr          |        Attr_Bi        |
    -------------------------------------------------------------------------------------------------
    |    FN     |    All    |    FN     |    All    |    FN     |    All    |    FN     |    All    |
 ns | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  | Ep  |  %  |
    -------------------------------------------------------------------------------------------------
    | 0 |19.7235| 0 |2.84877| 1 |15.7680| 2 |3.71203| 1 |19.5933| 1 |3.26121| 1 |20.5447| 1 |5.07898|
=====================================================================================================
```

Conclusion

We addressed one of the most important issues in understanding dialogs with contexts, the reference understanding. To our knowledge, this is the first attempt on solving the problem. We introduce a novel three-level neural network structure to accomplish the task of word-understanding, phrase-understanding and sentence-understanding taking the advantage of multiple types of neural networks including Convolution neural networks (CNN) and Long Short Term Memory neural networks (LSTM).

Thank you!