

# Neural Networks for Text Semantic Similarity Modeling

Zhe Hu





# Outline

---

**1. Introduction**

**2. Models**

**3. Experiments**

**4. Analysis**

---



# 1. Introduction

---

Given a sentence pair, we want to know where they have the same meaning or not?

| Sentence 1  | Label | Sentence 2   |
|---|-------|--|
| Samsung halts production of its Galaxy Note 7 as battery problems linger. | True  | #Samsung temporarily suspended production of its Galaxy #Note7 devices following reports |
| CO2 levels mark 'new era' in the world's changing climate.                | True  | CO2 levels haven't been this high for 3 to 5 million years.                              |
| The 7 biggest changes Obamacare made , and those that may disappear.      | False | What a repeal of Obamacare would look like , in plain English.                           |
| Fraugster , a startup that uses AI to detect payment fraud , raises \$5M. | False | AI is on the rise and in this case being applied to something worthwhile payment fraud.  |

This task is useful in many NLP systems, like dialog system, question answering, etc.

*Examples are taken from LanguageNet page: <https://languagenet.github.io/>*

---



# 1. Introduction

---

- Naïve Way:

If two sentences have many same words, they may have the same meaning.

However, in many cases it is not accurate.

1. What kind of fruit is this ?      What kind of fruit do you eat ? -- not similar

2. What are some DIY ways to remove scratches from car ?   How do I get scratches out of a vehicle ?

-- similar

- More Advanced Method:

Let's try NNs!





## 2. Models

---

- Sentence Encoder Models

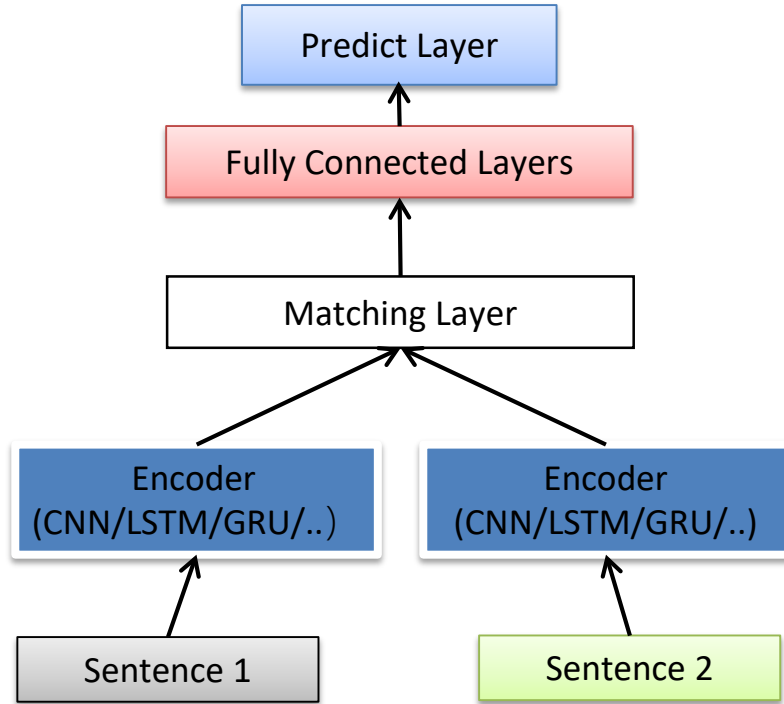
Use neural nets to encode each sentence into vector representation, and compare their semantic relationship.

- Sentence Interaction Models

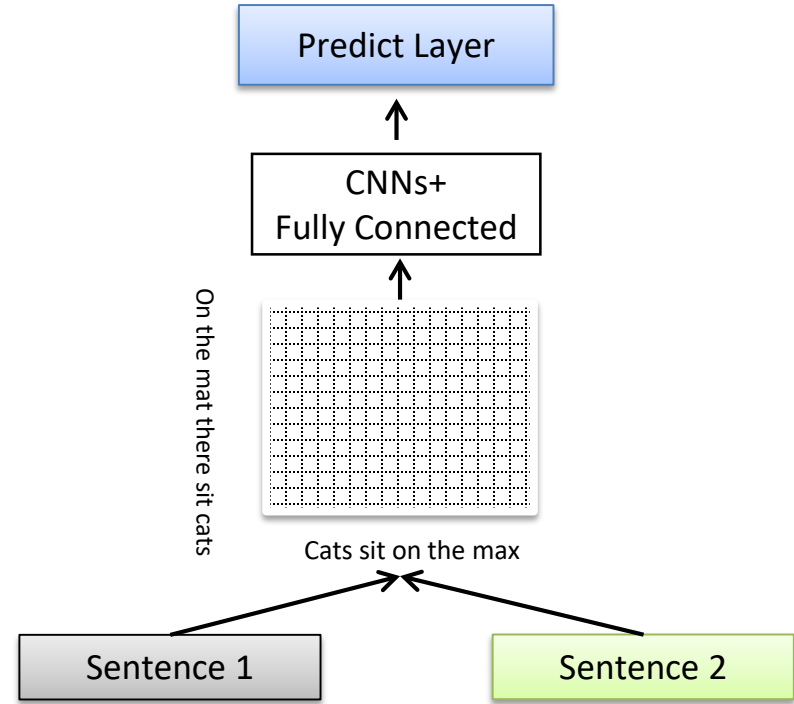
Calculate word-pair interactions to build the sentence interaction matrix, and use image recognition method to do classification.



## Sentence Encoder Structure



## Sentence Interaction Structure



# Some state-of-art models for sentence pair modeling:

| Model                              | structure                                     |
|------------------------------------|---|
| <b>InferSent</b> (Conneau, et al.) | BiLSTM encoder + row-max pooling              |
| <b>ESIM</b> (Chen et al.)          | BiLSTM encoder + soft alignment attention     |
| <b>MatchPyramid</b> (Pang, et al.) | Interaction-based model with multi-CNN layers |
| .....                              | .....   |

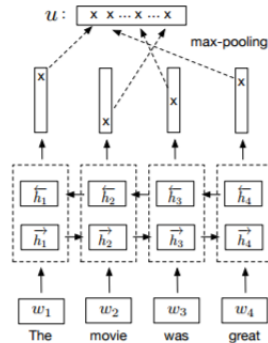


Figure 1. InferSent

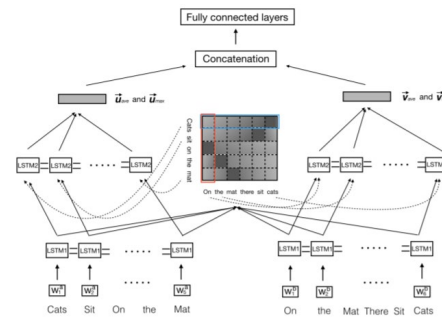


Figure 2. ESIM

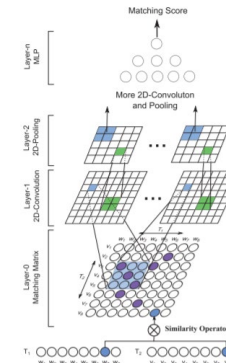


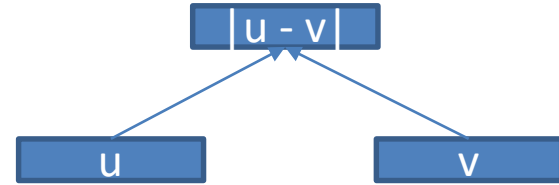
Figure 3. MatchPyramid

Note: figure1 and figure3 are from the original papers, and figure2 is from paper (Lan, et al.)

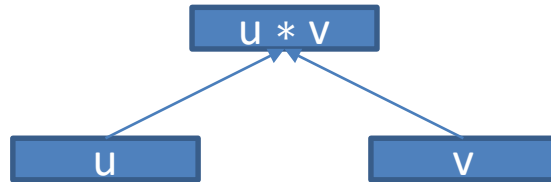
# Matching layer



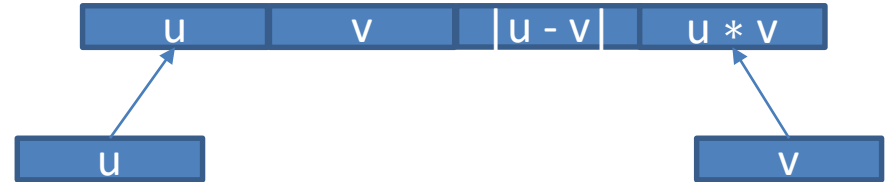
Matching 1



Matching 2



Matching 3

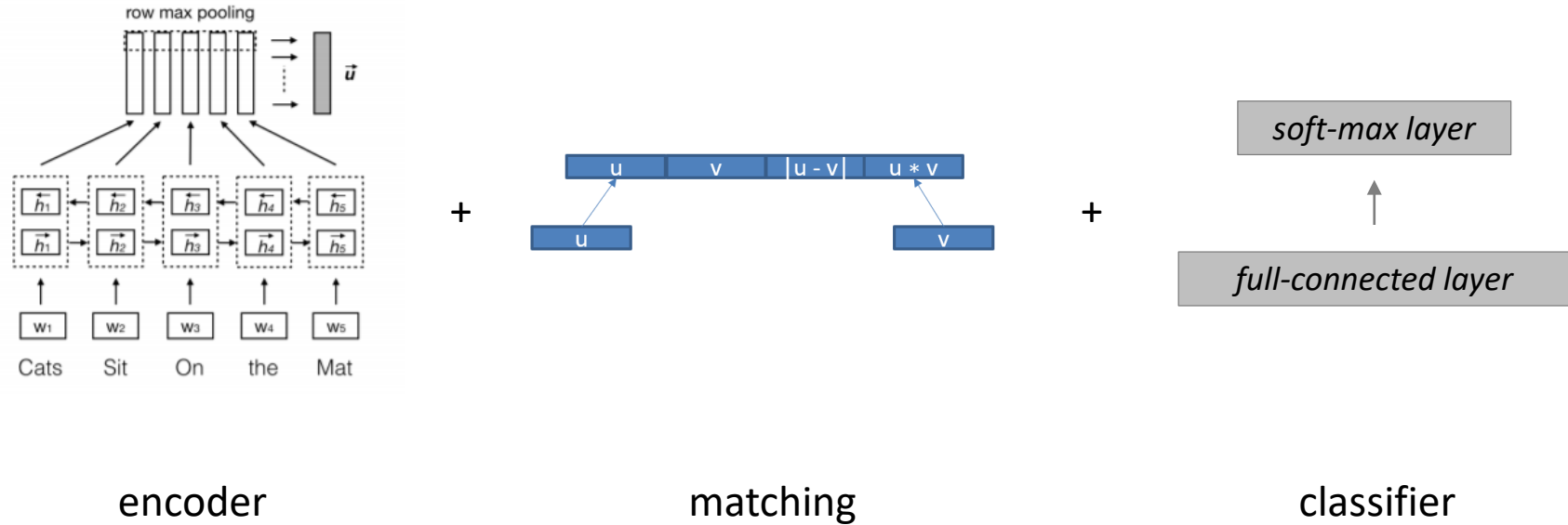


Matching 4: the best one

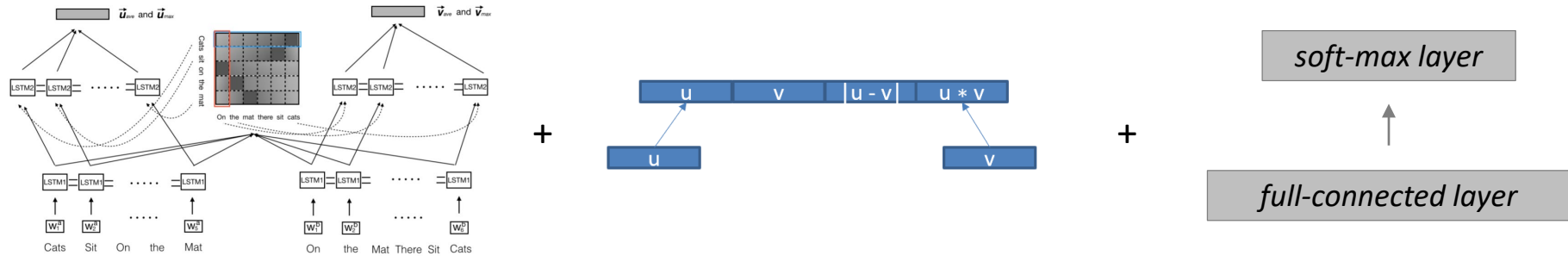




# Model 1: InferSent



# Model 2: ESIM (Enhanced LSTM for Natural Language Inference)



Encoder with soft-alignment  
attention

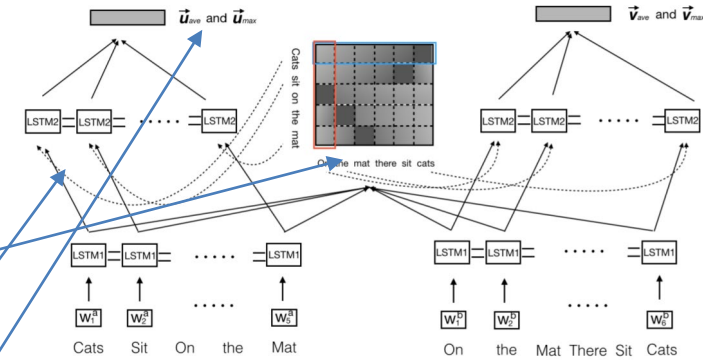
matching

classifier

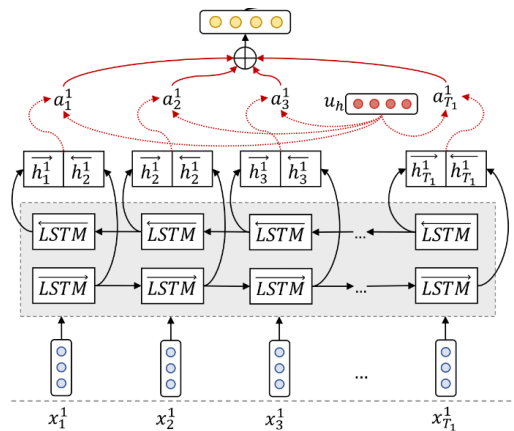
# Model 2: ESIM

soft alignment attention:

1.  $e_{ij} = \bar{a}_i^T \bar{b}_j$
2.  $\tilde{a}_i = \sum_{j=1}^{\ell_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_b} \exp(e_{ik})} \bar{b}_j, \forall i \in [1, \dots, \ell_a],$   
 $\tilde{b}_j = \sum_{i=1}^{\ell_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_a} \exp(e_{kj})} \bar{a}_i, \forall j \in [1, \dots, \ell_b],$
3.  $\mathbf{m}_a = [\bar{a}; \tilde{a}; \bar{a} - \tilde{a}; \bar{a} \odot \tilde{a}],$   
 $\mathbf{m}_b = [\bar{b}; \tilde{b}; \bar{b} - \tilde{b}; \bar{b} \odot \tilde{b}].$

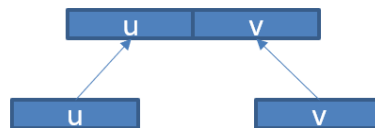


# Model 3: Siamese LSTM with self-attention(self-attentive lstm)



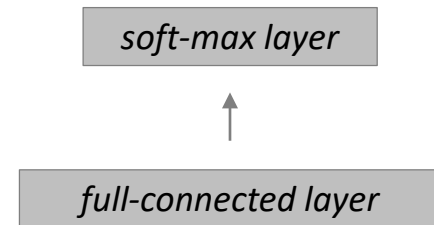
Encoder with self-attention

+



matching

+



classifier

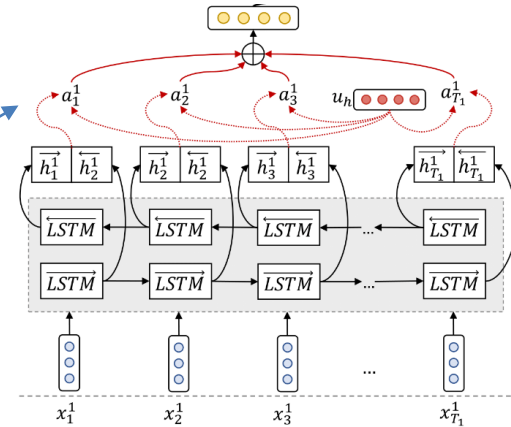
# Model 3: Siamese LSTM with self-attention

Context attention layer (Yang et al., 2016):

$$e_i = \tanh(W_h h_i + b_h), \quad e_i \in [-1, 1]$$

$$a_i = \frac{\exp(e_i^\top u_h)}{\sum_{t=1}^T \exp(e_t^\top u_h)}, \quad \sum_{i=1}^T a_i = 1$$

$$r = \sum_{i=1}^T a_i h_i, \quad r \in R^{2L}$$

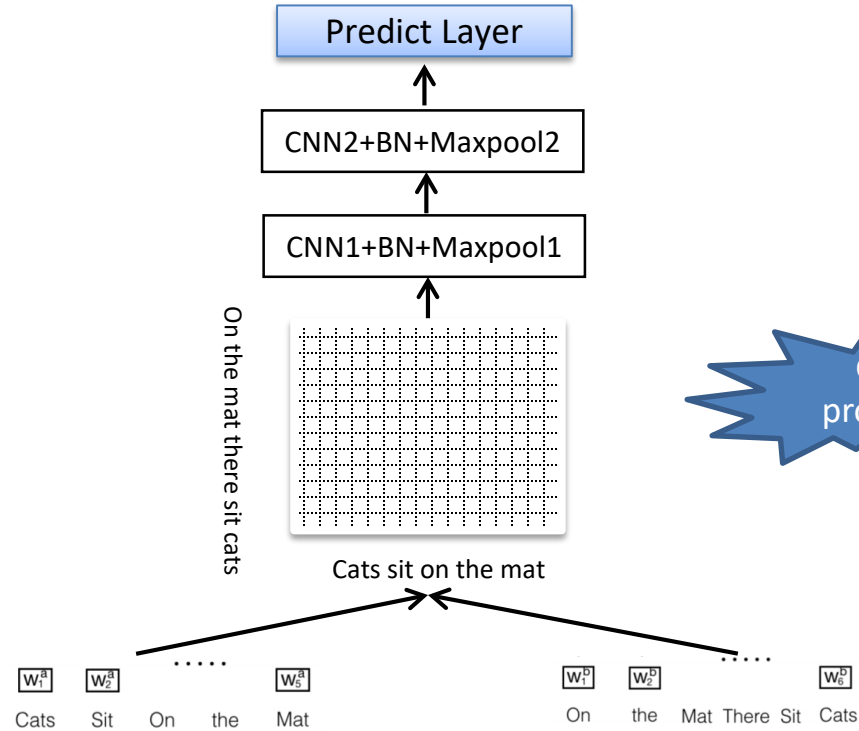


Encoder with self-attention (context-attention)

Interpretation: use fixed query  $u_h$ , that helps to identify the informative words in this sentence.

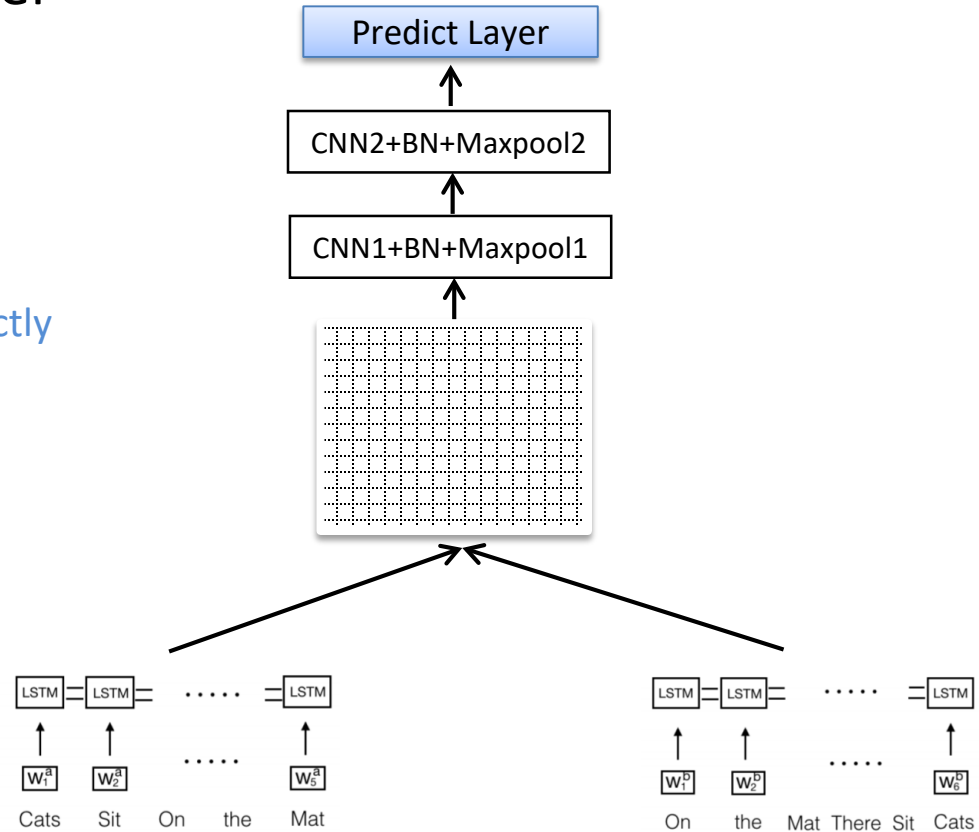
# Model 4: MatchPyramid

Using Glove to embed words into vector space.



# Model 5: Match Encoder

Instead of computing interaction directly after embedding, use encoder to aggregate some context information first.





## 3. Experiment & Results

---

### Dataset

- **Quora Question Pairs (Iyer et al., 2017):** Contains 400k question pairs collected from the Quora website, with positive and negative labels indicating whether they are paraphrase or not.

*POS What should I do to avoid sleeping in class ?*

*How do I not sleep in a boring class ?*

*NEG Do women support each other more than men do ? Do women need more compliments than men ?*

- **Twitter-URL (Lan et al., 2017):** includes 50k sentence pairs collected from tweets that share the same URL of news articles. This dataset contains both formal and informal language.

*NEG Are you open to 2020 ? I am open to doing everything I can right now.*

*From one Senator to the next .*

*POS Letter warned Wells Fargo of " widespread " fraud in 2007.*

*Letters suggest Wells Fargo scandal started earlier.*







### 3. Experiment & Results

---

#### Implementation details:

- Tensorflow & Keras
- Glove for word embedding, and char embedding is not used
- Batch size: 64
- Epoch: 30
- Optimizer: Adam





### 3. Experiment & Results

train:384,348  
validation: 10,000

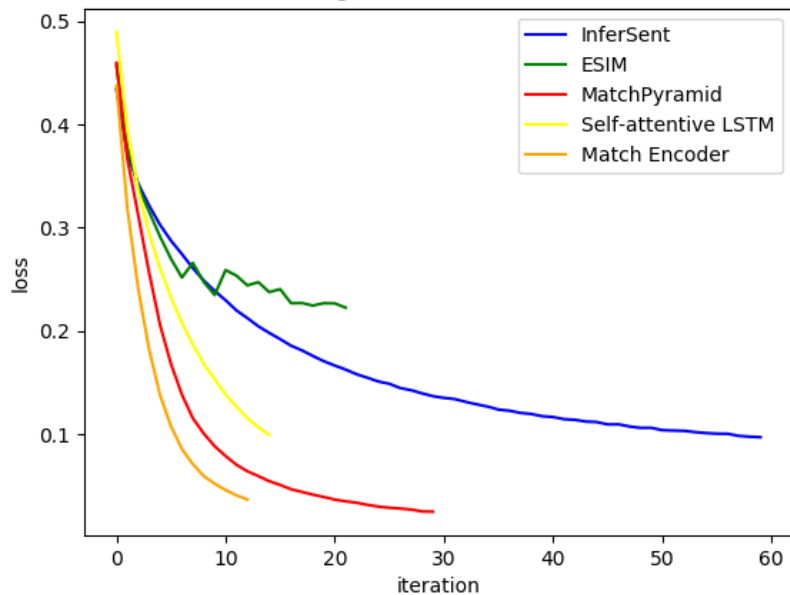
train:42,200  
validation: 9,334

| Model               | Quora (acc) | Twitter URL (f1) |
|---------------------|-------------|------------------|
| InferSent           | 86.93%      | 68.90%           |
| ESIM                | 87.06%      | 71.23%           |
| MatchPyramid        | 83.60%      | 69.87%           |
| Match Encoder       | 85.88%      | 68.89%           |
| self-attentive lstm | 82.89%      | --               |

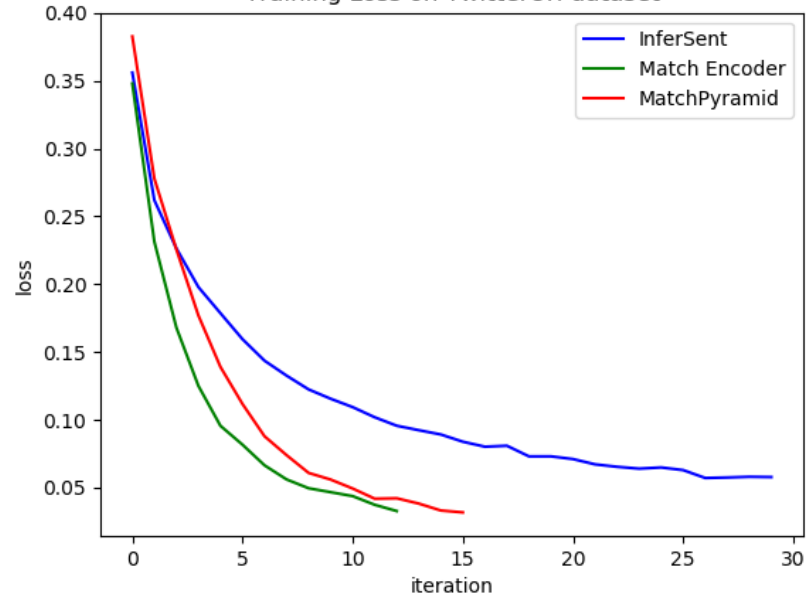
*Note: results of self-attentive lstm is using Glove.6B.100D, which gives the lower acc than Glove.840B.300D.*

### 3. Experiment & Results

Training Loss on Quora dataset



Training Loss on TwitterUrl dataset





## 4. Summary & Conclusion

---

- Paraphrase identification tasks ;
- Quora Question Pairs dataset & TwitterURL dataset;
- Different sentence-encoder models & sentence-interaction models.





## 4. Summary & Conclusion

---

### Some Conclusion:

- **MatchPyramid** does not work well on Quora dataset. We think this comes from high frequency of overlapping words (> 40%) between sentence pairs. However, **MatchEncoder** performs better because of aggregating of context information.
  - Glove.840B.300D(840B tokens, 2.2M vocab, 300d vectors) brings better results than Glove.6B.100D(6B tokens, 400K vocab, 100d) for embedding. We think word vectors trained on large corpus will definitely be more generic to be transferred on other tasks.
  - Self attention does not help a lot on sentence pair modelling task, which fails to capture the relationship between the sentence pairs in our consideration. On the contrary, interacted attention give better performance.
-



## 5. Future Work

---

- Combinations of two different types of models;
- Some other attention mechanism;
- Other sentence pair modelling tasks, like NLI.
- .....





Thank you !

