

Cross-lingual Short-text Matching of Question Pairs

Kai-Chou Yang
National Cheng Kung University



Task Overview

Problem Definition

The goal of this challenge is to build a cross-lingual short-text matching model. The source language is English and the target language is Spanish. Participants could train their models by applying advanced techniques to classify whether question pairs are the same meaning or not.

| | spn_1 | eng_1 | spn_2 | eng_2 | label |
|---|---|---|---|---|-------|
| 0 | ?No he podido pagar con mi tarjeta, que debo h... | I have not been able to pay with my card, what... | No puedo pagar mi pedido con mi tarjeta. | I can not pay for my order with my card. | 1 |
| 1 | ?Por qué aparece "no pagado" cuando el pago ya... | Why does it appear "not paid" when the payment... | He pagado por transferencia bancaria, pero el ... | I paid by bank transfer, but the payment has n... | 0 |
| 2 | ?Cuándo recibiré mi reembolso si pago con tarj... | When will I receive my refund if I pay by cred... | ¿Cuándo recibiré el reembolso si cancelo mi pe... | When will I receive the refund if I cancel my ... | 0 |
| 3 | ?Qué pasará después de abrir una disputa? | What will happen after opening a dispute? | ¿Qué pasará después de haber enviado mi solici... | What will happen after I have sent my Warranty... | 0 |
| 4 | El producto que he recibido no corresponde con... | The product I received does not correspond wit... | He recibido un producto que no funciona. | I received a product that does not work. | 0 |

Overview of this dataset. We are asked to distinguish the intents between sentence 1 and sentence 2. Label being 1 indicates that spn_1 and spn_2 have the same meaning.

Feature Engineering

Feature Engineering

- Distance features
 - sentence vectorization
 - bag-of-words model
 - bag-of-words weighted by TF/IDF
 - weighted average of word embedding based on TF/IDF
 - Topic features
 - Text features

Feature Engineering

- Distance features
- Topic features
 - the same intents, the same prefixes
 - a. **I want to** create an argument
 - b. **I want to** talk to a person
 - c. **I want to** ask whether...
 - captured by LDA and LSI
- Text features

Feature Engineering

- Distance features
- Topic features
- Text features

Q1: What time is it in China?

Q2: What is processing time?

Feature Engineering

- Distance features
- Topic features
- Text features

Q1: What time is it in China? $\text{length}(\text{Q1}) = 6$

Q2: What is processing time? $\text{length}(\text{Q2}) = 4$

$\text{length_diff} = 2$

Feature Engineering

- Distance features
- Topic features
- Text features

Q1: What time is it in China? $\text{unique_words}(Q1) = 6, \text{not_stopwords}(Q1) = 2$

Q2: What is processing time? $\text{unique_words}(Q1) = 6, \text{not_stopwords}(Q1) = 2$

Feature Engineering

- Distance features
- Topic features
- Text features
- Get 85% accuracy with gradient boosting trees
 - outperforms some encoder based models like BiLSTM

Models Design

Decomposable attention

- Given 2 sentences (a_1, \dots, a_n) , (b_1, \dots, b_m)

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | | | |
| a_2 | | | |
| a_3 | | | |
| a_4 | | | |

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | | |
| a_2 | | | |
| a_3 | | | |
| a_4 | | | |

$$\text{Score}(a_1, b_1) = a_1 * b_1 = 0.7$$

Decomposable attention

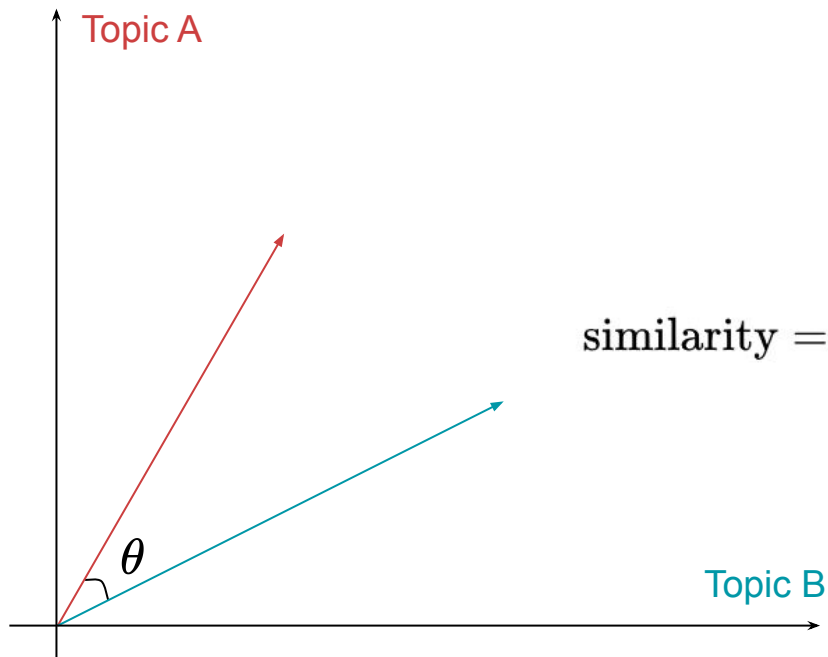
- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | | |
| a_2 | | | |
| a_3 | | | |
| a_4 | | | |

$$\text{Score}(a_1, b_1) = \underline{a_1} * b_1 = 0.7$$



Decomposable attention



$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | | |
| a_2 | 0.13 | | |
| a_3 | | | |
| a_4 | | | |

$$\text{Score}(a_1, b_1) = a_1 * b_1 = 0.7$$

$$\text{Score}(a_2, b_1) = a_2 * b_1 = 0.13$$

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | | |
| a_2 | 0.13 | | |
| a_3 | 0.02 | | |
| a_4 | | | |

$$\text{Score}(a_1, b_1) = a_1 * b_1 = 0.7$$

$$\text{Score}(a_2, b_1) = a_2 * b_1 = 0.13$$

$$\text{Score}(a_3, b_1) = a_3 * b_1 = 0.02$$

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | | |
| a_2 | 0.13 | | |
| a_3 | 0.02 | | |
| a_4 | 0.15 | | |

$$\text{Score}(a_1, b_1) = a_1 * b_1 = 0.7$$

$$\text{Score}(a_2, b_1) = a_2 * b_1 = 0.13$$

$$\text{Score}(a_3, b_1) = a_3 * b_1 = 0.02$$

$$\text{Score}(a_4, b_1) = a_4 * b_1 = 0.15$$

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | 0.07 | 0.64 |
| a_2 | 0.13 | 0.46 | 0.06 |
| a_3 | 0.02 | 0.39 | 0.10 |
| a_4 | 0.15 | 0.08 | 0.20 |



aligned sequence(b_1', b_2', b_3')

$$b_1' = 0.70 * a_1 + 0.13 * a_2 + 0.02 * a_3 + 0.15 * a_4$$

$$b_2' = 0.07 * a_1 + 0.46 * a_2 + 0.39 * a_3 + 0.08 * a_4$$

$$b_3' = 0.64 * a_1 + 0.06 * a_2 + 0.10 * a_3 + 0.20 * a_4$$

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | 0.07 | 0.64 |
| a_2 | 0.13 | 0.46 | 0.06 |
| a_3 | 0.02 | 0.39 | 0.10 |
| a_4 | 0.15 | 0.08 | 0.20 |



aligned sequence (b_1', b_2', b_3')

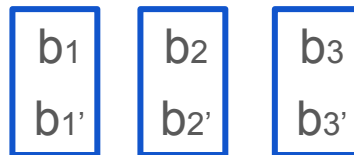
$$b_1' = 0.70 * a_1 + 0.13 * a_2 + 0.02 * a_3 + 0.15 * a_4$$

$$b_2' = 0.07 * a_1 + 0.46 * a_2 + 0.39 * a_3 + 0.08 * a_4$$

$$b_3' = 0.64 * a_1 + 0.06 * a_2 + 0.10 * a_3 + 0.20 * a_4$$

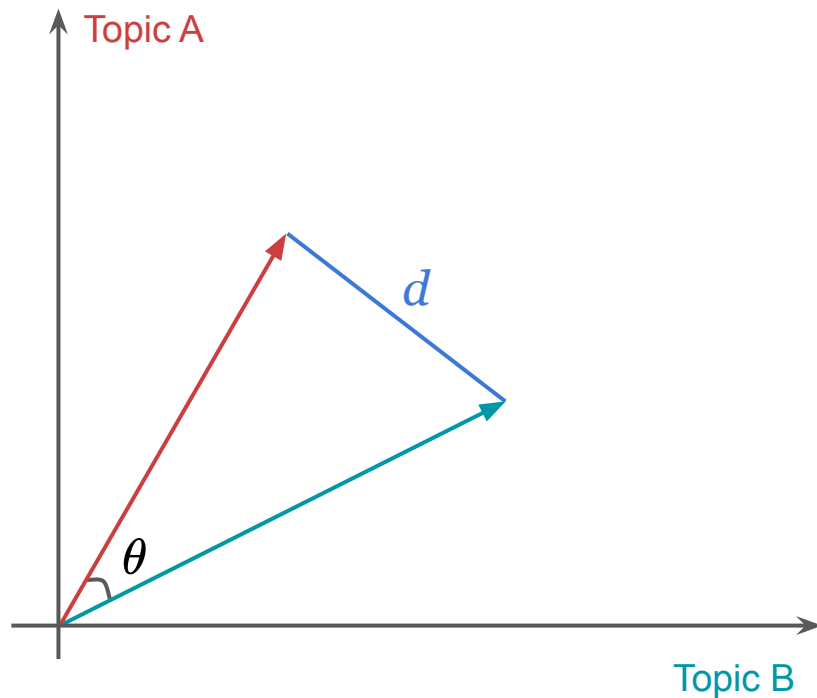
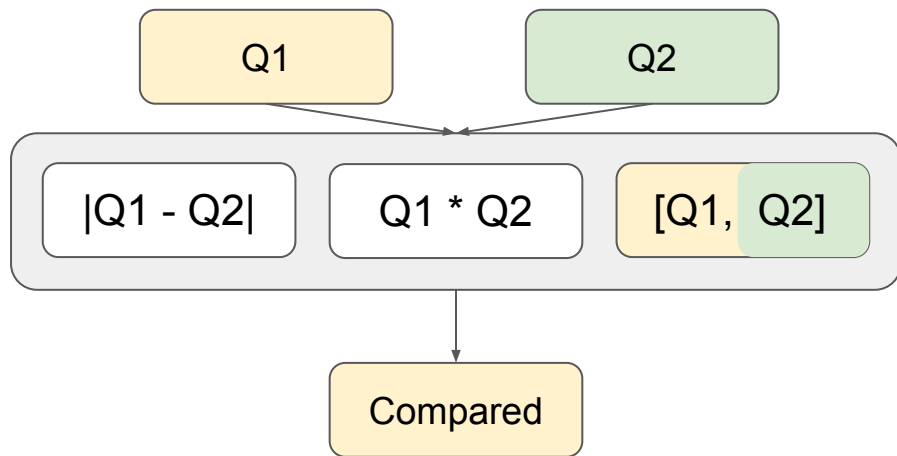


Compare the original
sequence and aligned
sequence



Decomposable attention

- Heuristic Matching



Decomposable attention

- Ideas: **align** and **compare** the words in sent1 with the similar words in sent2.
 - Helps us focus on the difference
 - What is <X> ?
 - What is EMS
 - Can you tell me what is area number
 - Do you know what is going on
 - How can I get <X>?
 - How can I get an order
 - How can I get my refund
 - How can I get questions answered

Decomposable attention

- Given 2 sentences $(a_1, \dots, a_n), (b_1, \dots, b_m)$

| | b_1 | b_2 | b_3 |
|-------|-------|-------|-------|
| a_1 | 0.70 | 0.07 | 0.64 |
| a_2 | 0.13 | 0.46 | 0.06 |
| a_3 | 0.02 | 0.39 | 0.10 |
| a_4 | 0.15 | 0.08 | 0.20 |



aligned sequence (b_1', b_2', b_3')

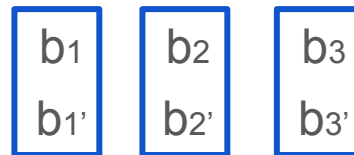
$$b_1' = 0.70 * a_1 + 0.13 * a_2 + 0.02 * a_3 + 0.15 * a_4$$

$$b_2' = 0.07 * a_1 + 0.46 * a_2 + 0.39 * a_3 + 0.08 * a_4$$

$$b_3' = 0.64 * a_1 + 0.06 * a_2 + 0.10 * a_3 + 0.20 * a_4$$



Compare the original
sequence and aligned
sequence

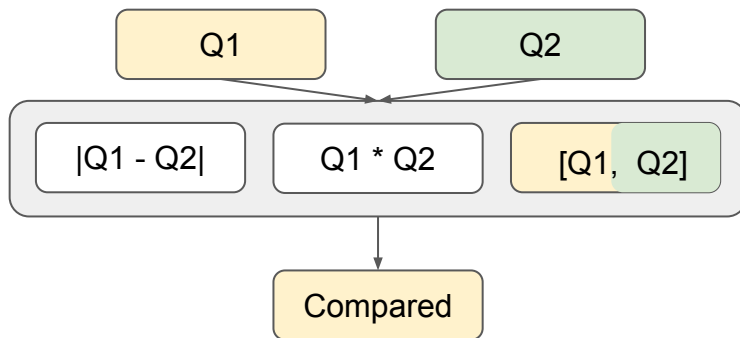


Decomposable attention

| | word1 | word2 | word3 | word4 | word5 |
|------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Sentence 2 | I | love | you | , | Jack |

Decomposable attention

| | word1 | word2 | word3 | word4 | word5 |
|------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Sentence 2 | I | love | you | , | Jack |



Decomposable attention

| | word1 | word2 | word3 | word4 | word5 |
|--------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Sentence 2 | I | love | you | , | Jack |
| Align 2 to 1 | | | | | |

Decomposable attention

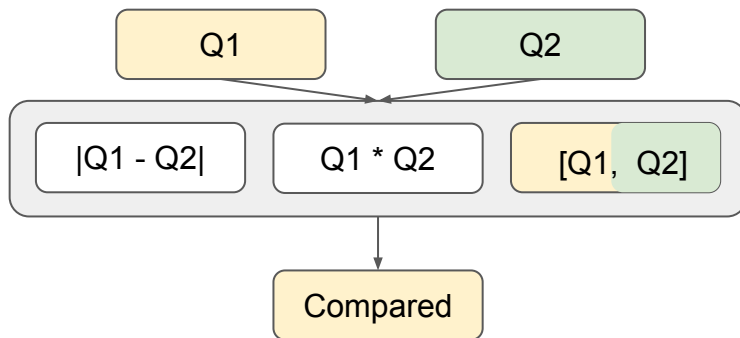
| | word1 | word2 | word3 | word4 | word5 |
|--------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Sentence 2 | I | love | you | , | Jack |
| Align 2 to 1 | Jack | | | | |

Decomposable attention

| | word1 | word2 | word3 | word4 | word5 |
|--------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Sentence 2 | I | love | you | , | Jack |
| Align 2 to 1 | Jack | , | I | love | you |

Decomposable attention

| | word1 | word2 | word3 | word4 | word5 |
|--------------|-------|-------|-------|-------|-------|
| Sentence 1 | Jack | , | I | love | you |
| Align 2 to 1 | Jack | , | I | love | you |

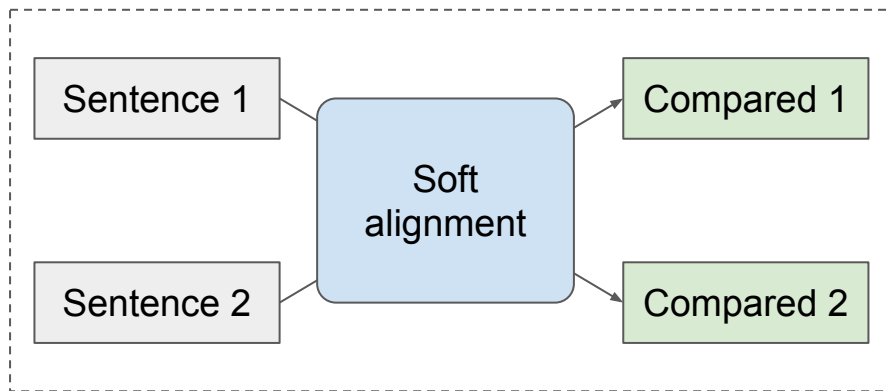


Limitations of Attention

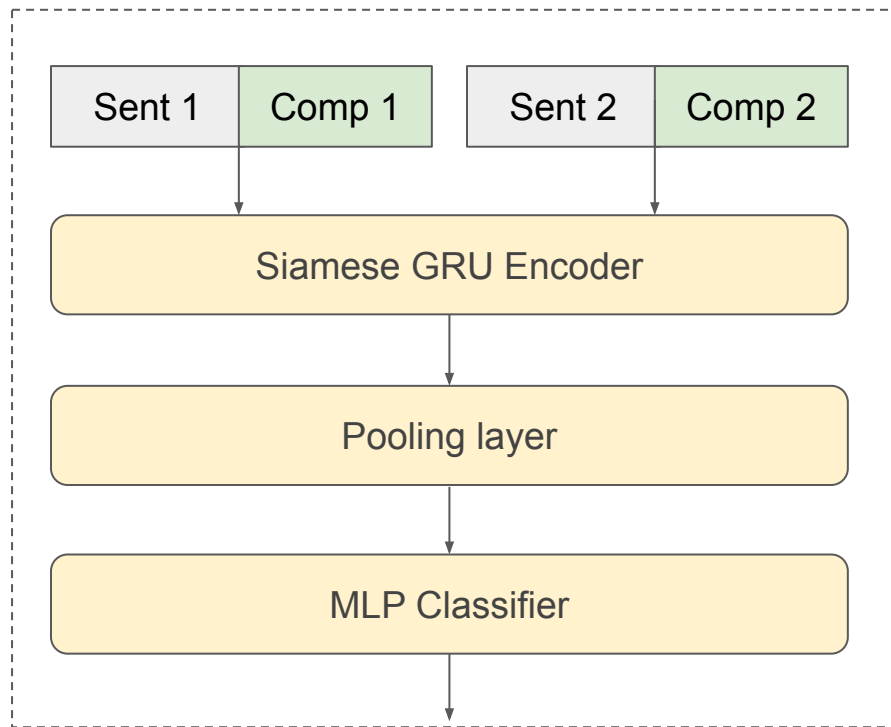
1. Drops relative position information
 - lose dependencies between words
2. Cold start relations
 - source languages are different
 - i. training set are translated from english
 - ii. testing set are spanish, originally
 - the performance of relation learning is restricted to the translator

Compare-Propagate Recurrent Neural Network

- Follow the idea of [CAFE](#)
 - propagate the compared results with the original word embeddings



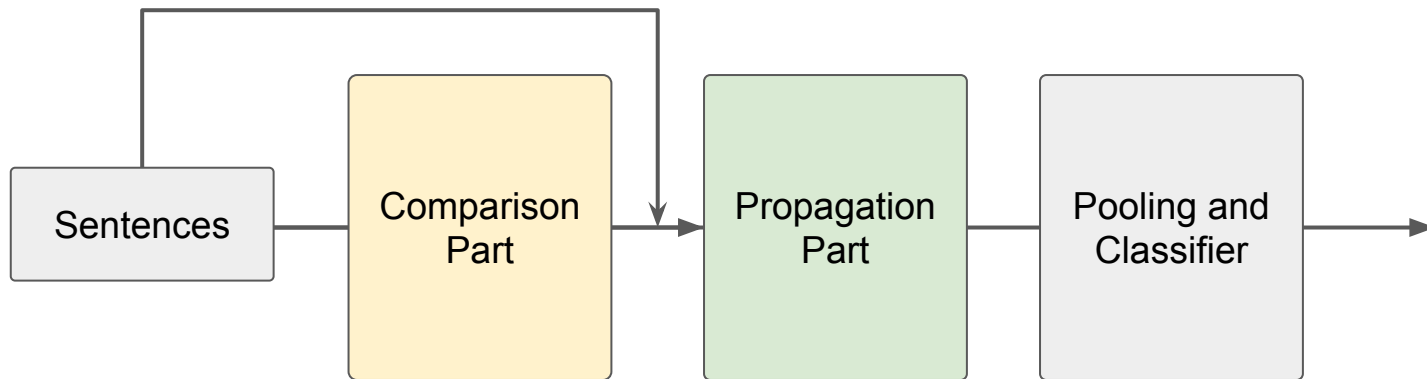
Comparison Part



Propagation Part

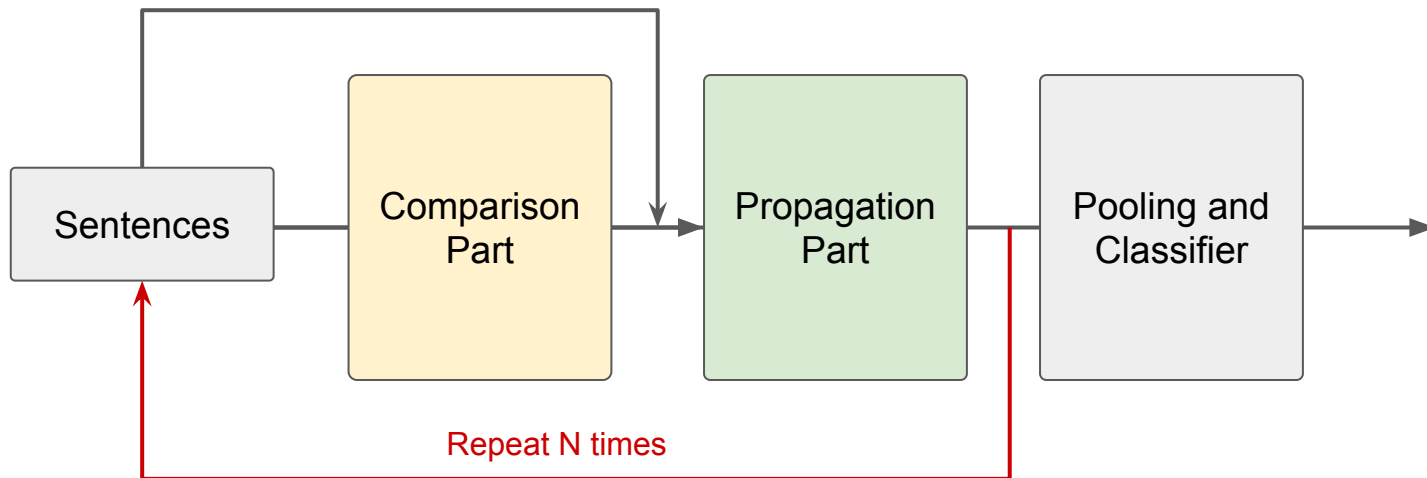
Compare-Propagate Recurrent Neural Network

- Advantages of CPRNN
 - capture the positional information and word dependency
 - propagate the difference with word embedding simultaneously



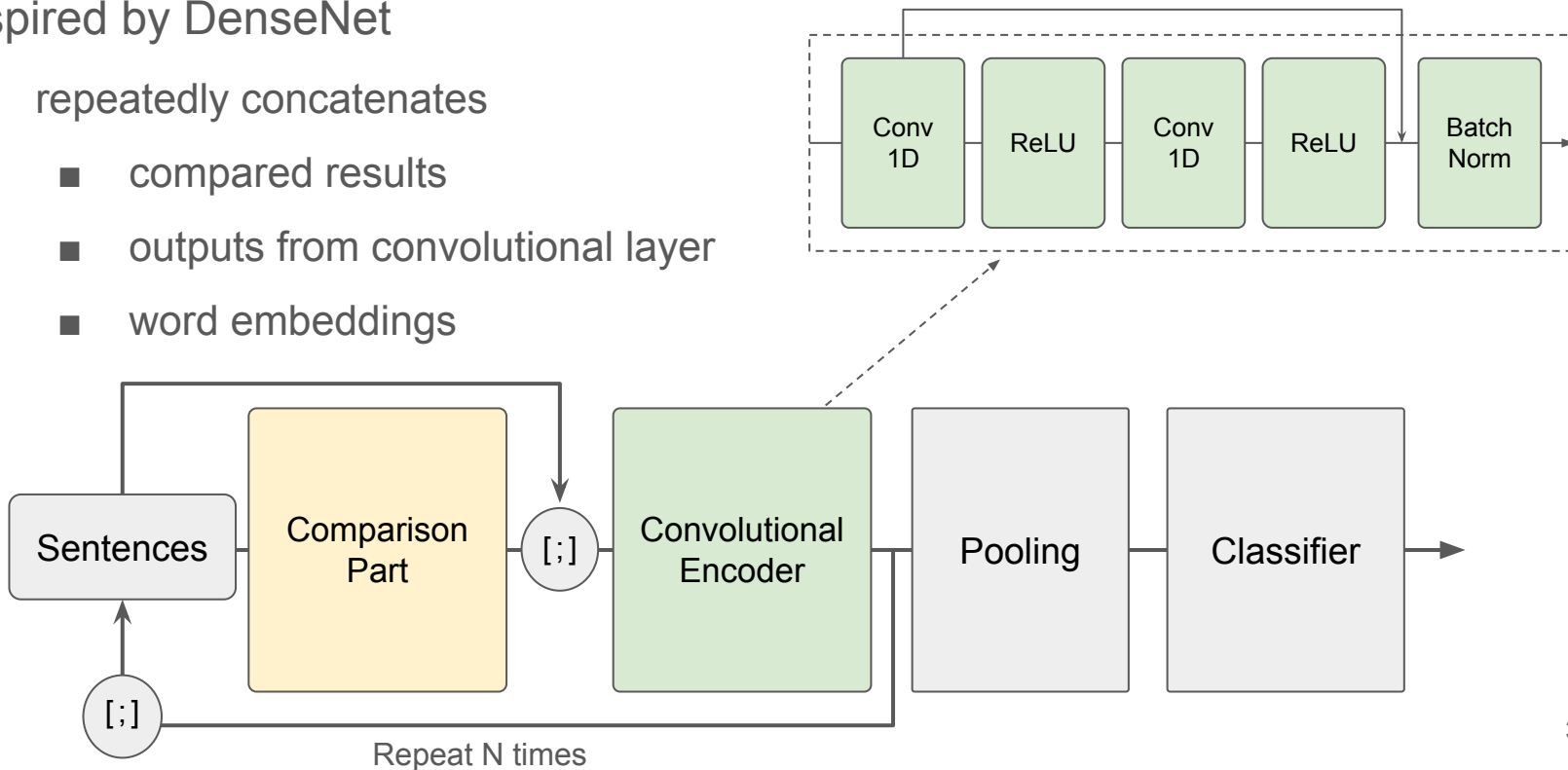
Compare-Propagate Recurrent Neural Network

- Advantages of CPRNN
 - capture the positional information and word dependency
 - propagate the difference with word embedding simultaneously



Densely Augmented Convolutional Neural Network

- Inspired by DenseNet
 - repeatedly concatenates
 - compared results
 - outputs from convolutional layer
 - word embeddings



Densely Augmented Convolutional Neural Network

- Advantages of DACNN
 - faster and more efficient than the recurrent based models

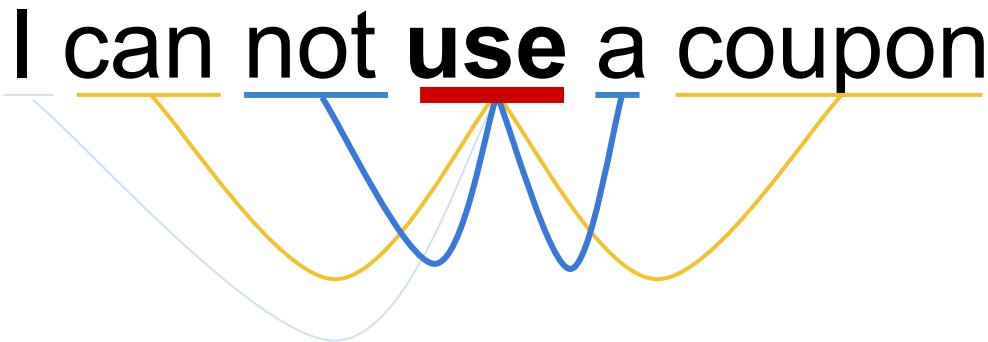
| Model | log-loss | Training time per epoch | #Parameters |
|-------|--------------|-------------------------|--------------|
| CPRNN | 0.353 | 25 secs | 1.7m |
| DACNN | 0.356 | 10 secs | 0.35m |

Table: performance comparsion between CPRNN and DACNN

Densely Augmented Convolutional Neural Network

- Advantages of DACNN
 - faster and more efficient than the recurrent based models
 - capturing regional representation, which mitigates the impact of cold start words

I can not **use** a coupon



Model Variety - Inter-Models

- We implemented 3 models based on different hypotheses
 - decomposable attention
 - compare-propagate recurrent neural network
 - densely augmented convolutional neural network

| | CPRNN | DACNN | Decomposable attention |
|------------------------|-------|-------|------------------------|
| CPRNN | 1 | 0.93 | 0.9 |
| DACNN | 0.93 | 1 | 0.92 |
| Decomposable attention | 0.9 | 0.92 | 1 |

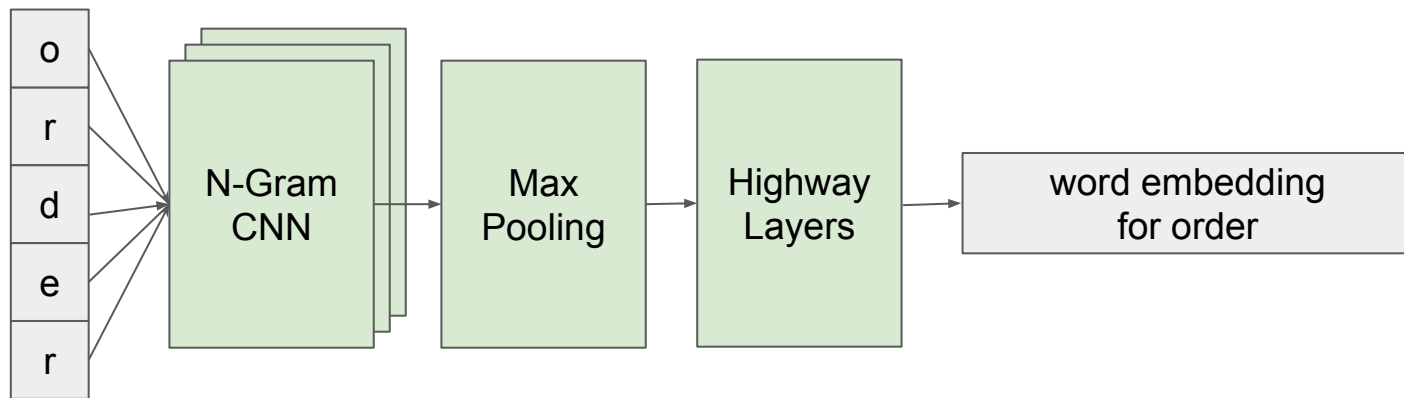
Table: the correlation matrix of prediction from each model. Since each model has a low correlation to others, we can optimize the accuracy by 2% through simply average the predictions.

Model Variety - Intra-Models

- We implemented 3 models based on different hypotheses
- Moreover, we use 3 different input to train our model
 - a. **Word level** - pretrained word embeddings
 - b. **Char level** - generated by n-gram CNN
 - c. **Word level with engineered features**

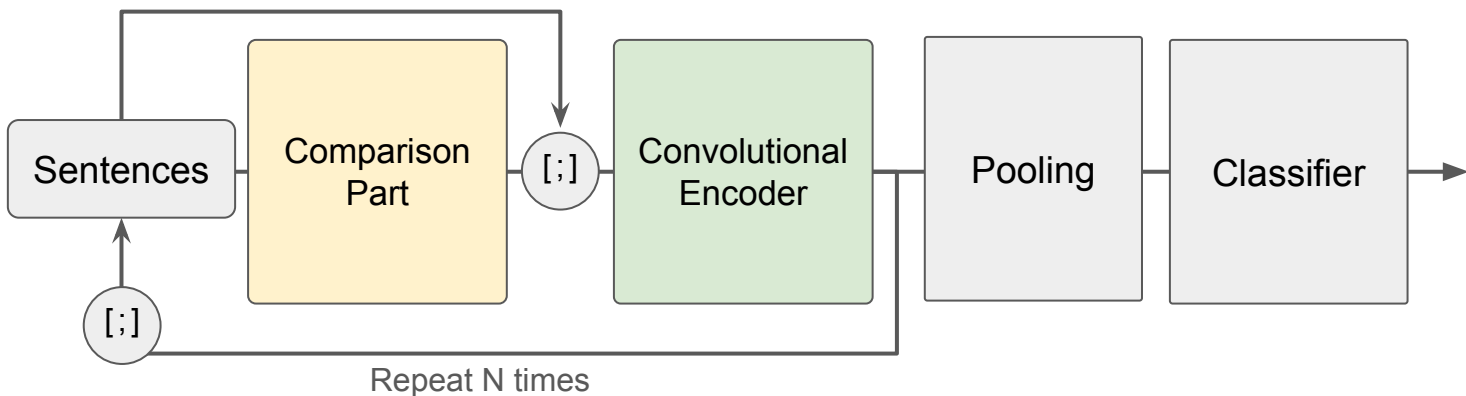
Model Variety - Intra-Models

- We implemented 3 models based on different hypotheses
- Moreover, we use 3 different input to train our model
 - a. **Word level** - pretrained word embeddings
 - b. **Char level** - generated by n-gram CNN
 - c. **Word level with engineered features**



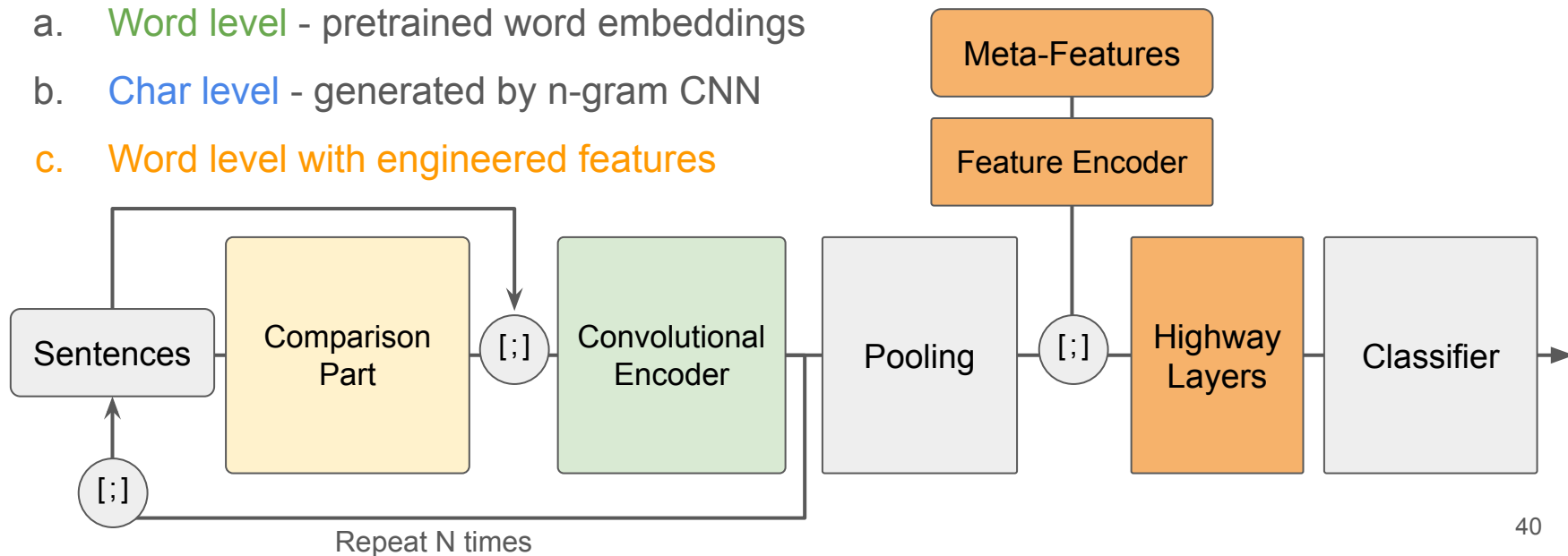
Model Variety - Intra-Models

- We implemented 3 models based on different hypotheses
- Moreover, we use 3 different input to train our model
 - a. **Word level** - pretrained word embeddings
 - b. **Char level** - generated by n-gram CNN
 - c. **Word level with engineered features**



Model Variety - Intra-Models

- We implemented 3 models based on different hypotheses
- Moreover, we use 3 different input to train our model
 - a. **Word level** - pretrained word embeddings
 - b. **Char level** - generated by n-gram CNN
 - c. **Word level with engineered features**



Model Variety - Intra-Models

- We implemented 3 models based on different hypotheses
- Moreover, we use 3 different input to train our model
 - a. **Word level** - pretrained word embeddings
 - b. **Char level** - generated by n-gram CNN
 - c. **Word level with engineered features**

| | CNN word level | CNN char level | CNN rich features |
|-------------------|----------------|----------------|-------------------|
| CNN word level | 1 | 0.89 | 0.95 |
| CNN char level | 0.89 | 1 | 0.89 |
| CNN rich features | 0.95 | 0.89 | 1 |

Table: the correlation matrix of models with different inputs.

Model Variety - Intra-Models

| Correlation | CNN word level | CNN char level | CNN rich features |
|-------------------|----------------|----------------|-------------------|
| CNN word level | 1 | 0.89 | 0.95 |
| CNN char level | 0.89 | 1 | 0.89 |
| CNN rich features | 0.95 | 0.89 | 1 |

| Correlation | RNN word level | RNN char level | RNN rich features |
|-------------------|----------------|----------------|-------------------|
| RNN word level | 1 | 0.87 | 0.96 |
| RNN char level | 0.87 | 1 | 0.87 |
| RNN rich features | 0.96 | 0.87 | 1 |

| Model | log-loss |
|-------------------|--------------|
| CNN word level | 0.356 |
| CNN char level | 0.403 |
| CNN rich features | 0.371 |
| Baggings | 0.343 |
| RNN word level | 0.353 |
| RNN char level | 0.40 |
| RNN rich features | 0.37 |
| Baggings | 0.34 |

Regularization

Regularization: Soft Labeling

- Why soft labeling?
 - The training data is noisy
 - Rule 1:
 - $\text{Sim}(A, B) = 1, \text{Sim}(B, C) = 1 \rightarrow \text{Sim}(A, C) = 1$
 - Rule 2:
 - $\text{Sim}(A, B) = 1, \text{Sim}(B, C) = 0 \rightarrow \text{Sim}(A, C) = 0$
 - Rule 1 and Rule 2 only hold for 75% and 95% cases respectively
 - The soft label is suitable for regularization and usually more meaningful.

Regularization: Soft Labeling

- Why soft labeling?
 - The training data is noisy
 - The soft label is suitable for regularization and usually more meaningful.

| Model | local log-loss |
|----------------------------|----------------|
| CPRNN | 0.242 |
| CPRNN with Soft label 0.98 | 0.237 |

Regularization: Embedding Dropout

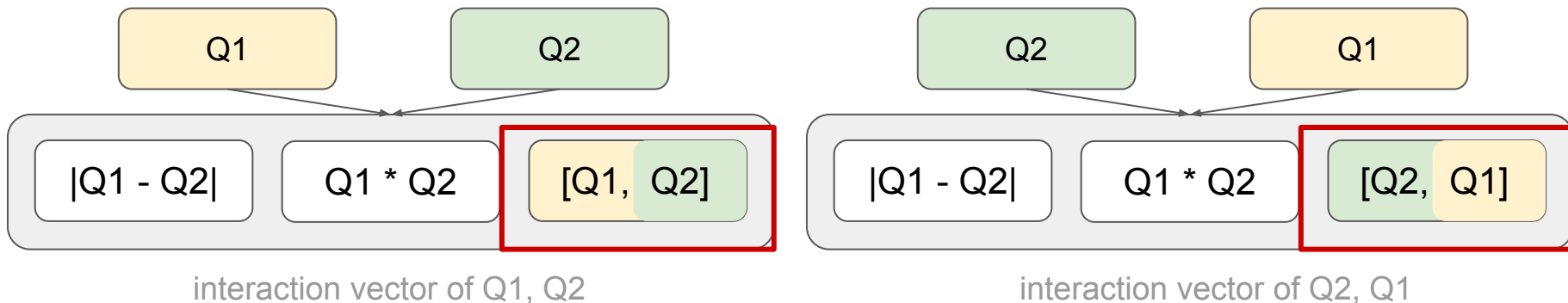
- Embedding Dropout
 - drop words in the input sentence randomly
 - an idea of data augmentation

| Model | local logloss |
|-----------------------------------|---------------|
| TextCNN without ED | 0.40 |
| TextCNN with ED | 0.31 |
| Cafe without ED | 0.27 |
| Cafe with ED | 0.24 |
| Decomposable attention without ED | 0.29 |
| Decomposable attention with ED | 0.25 |

Table : The performance comparsion of embedding dropout (ED)

Regularization: Symmetry

- Symmetric architecture helps for generalization
- Symmetry is a vital property for similarity metric
- However, the most NLI models do not obey symmetry when modeling the interactions between sentences



$\text{Sim}(Q1, Q2) = \text{Sim}(Q2, Q1)$ theoretically, while they do not have the same interaction vectors

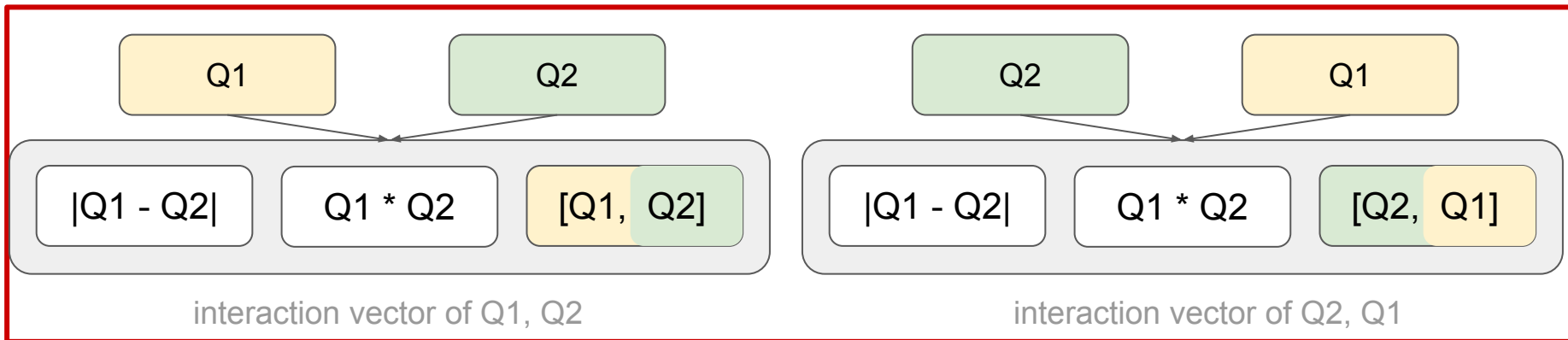
Regularization: Symmetry

- Symmetric architecture helps for generalization
- Symmetry is a vital property for similarity metric
 - Experiment on
 - train set: the original pair **<Q1, Q2>**
 - swapped train set: swap the <Q1, Q2> with **<Q2, Q1>**

| Model | log-loss for train set | log-loss for swapped train set | Difference |
|--------------------------------|------------------------|---------------------------------------|------------|
| CPRNN | 0.243 | 0.245 | 0.002 |
| DACNN | 0.267 | 0.276 | 0.009 |
| Meta Decomposable Attention | 0.212 | 0.219 | 0.007 |

Regularization: Symmetry

- Symmetric architecture helps for generalization
- Symmetry is a vital property for similarity metric
- However, the most NLI models do not obey symmetry when modeling the interactions between sentences



Average 2 vectors as the final output

Conclusions

- Design 56 high-level and interpretable features
- Implement 3 deep learning architectures
 - modify CAFE to fit this task
 - propose DACNN which is fast and lightweight
- Create the varieties intra-model
- Explore regularization for short text matching

Thank you

Reference

1. [A Compare-Propagate Architecture with Alignment Factorization for Natural Language Inference](#)
2. [A Decomposable Attention Model for Natural Language Inference](#)
3. [Character-Aware Neural Language Models](#)
4. [Convolutional Neural Networks for Sentence Classification](#)
5. [DART: Dropouts meet Multiple Additive Regression Trees](#)
6. [DeepFM: A Factorization-Machine based Neural Network for CTR Prediction](#)
7. [Densely Connected Convolutional Networks](#)
8. [Distilling the Knowledge in a Neural Network](#)
9. [Enhanced LSTM for Natural Language Inference](#)
10. [Extrapolation in NLP](#)
11. [From Word Embeddings To Document Distances](#)
12. [Learning with Noisy Labels](#)
13. [Regularizing and Optimizing LSTM Language Models](#)
14. [Regularizing Neural Networks by Penalizing Confident Output Distributions](#)
15. [R-net: Machine reading comprehension with self matching networks](#)
16. [Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm](#)