Toxic Comment Classification 최종발표

(https://github.com/Timmy-Oh/kaggle_toxic_comment)

(김지현, 이상열, 윤상필, 박희준, 오영택)

Overview

In this competition, you're challenged to build **a multi-headed model** that's capable of **detecting different types of of toxicity** like threats, obscenity, insults, and identity- based hate better than Perspective's current models. You'll be using **a dataset of comments from Wikipedia's talk page edits**.

Evaluation:

- the mean column- wise ROC AUC

Data

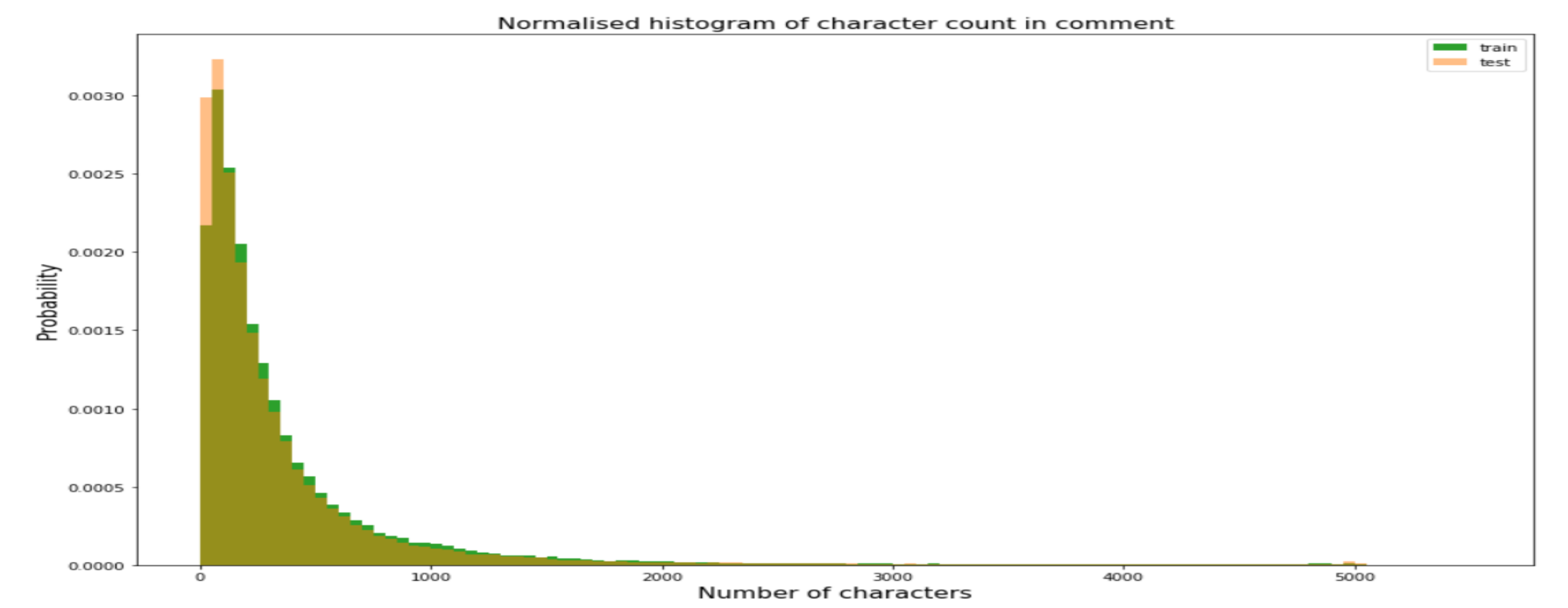
- **id :** the unique Id of comment with hash format
- **comment_text :** The actual text contents
- **toxic, severe_toxic, obscene, threat, insult, identity_hate :** the labels of comments

- Total number of comments in **the training data** : 159571
- Total number of comments in **the test data** : 153164

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

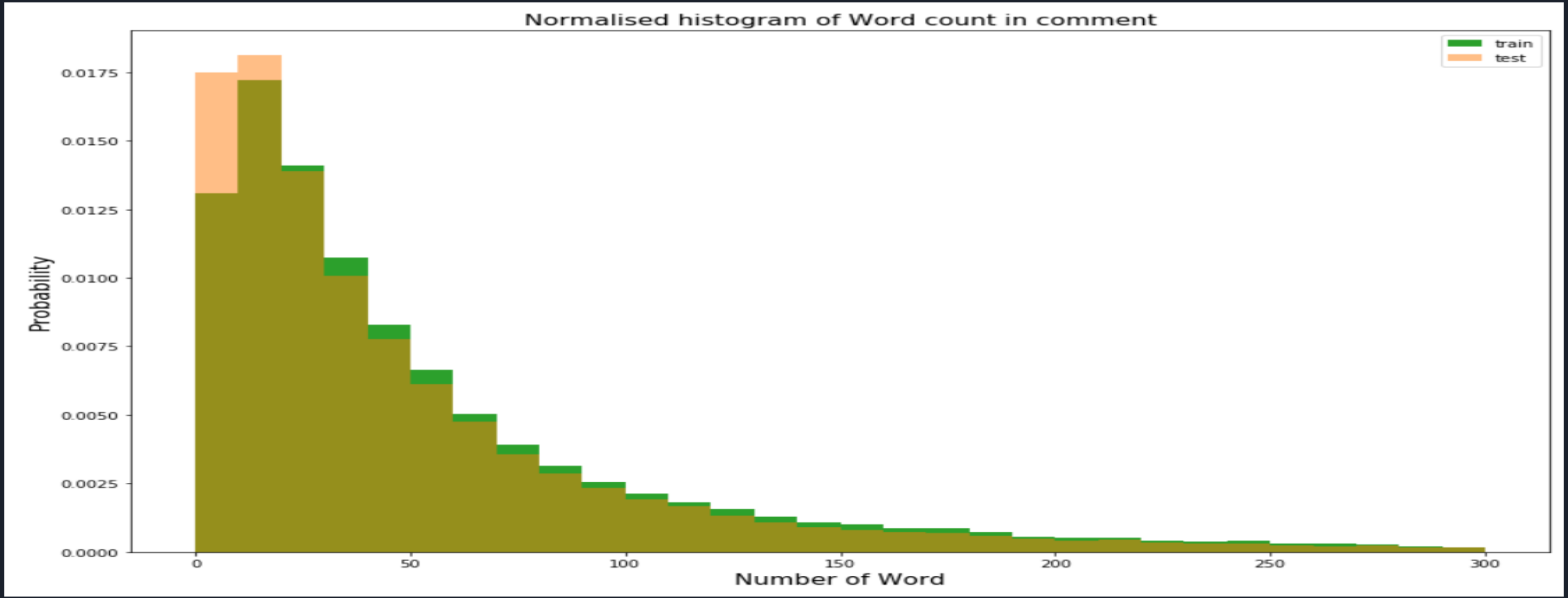
Character Based Sentence Length Dist.

mean-train 394.07 std-train 590.72 mean-test 364.88 std-test 592.49 max-train 5000.00 max-test 5000.00



Word Based Sentence Length Dist.

mean-train 67.27 std-train 99.23 mean-test 61.61 std-test 98.96 max-train 1411.00 max-test 2321.00



Preparation

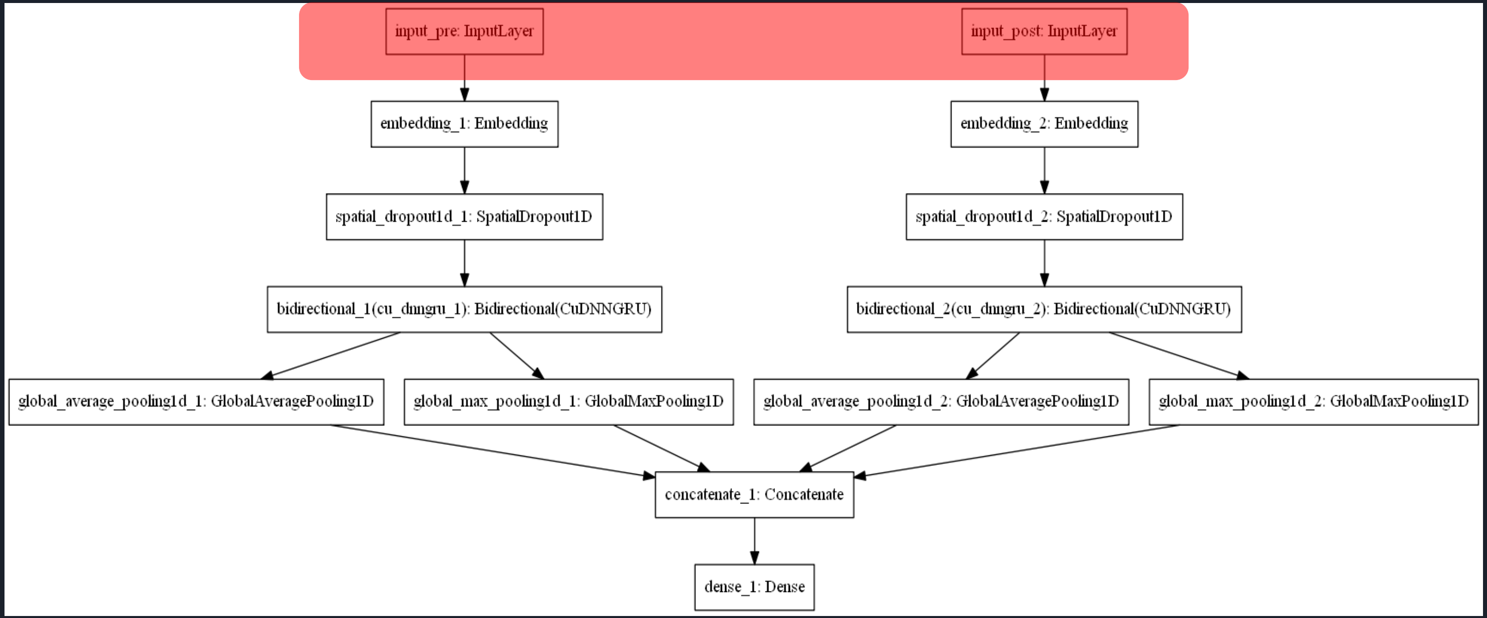
Requirements:

- Anaconda
- tensorflow- gpu == 1.6.0
- keras == 2.1.5

Pretrained Word Embeddings:

- FastText: crawl- 300d- 2M
- Glove: glove.840B.300d

Baseline


input_pre: InputLayer
input_post: InputLayer
embedding_1: Embedding
embedding_2: Embedding
spatial_dropout1d_1: SpatialDropout1D
spatial_dropout1d_2: SpatialDropout1D
bidirectional_1(cu_dnngru_1): Bidirectional(CuDNNGRU)
bidirectional_2(cu_dnngru_2): Bidirectional(CuDNNGRU)
global_average_pooling1d_1: GlobalAveragePooling1D
global_max_pooling1d_1: GlobalMaxPooling1D
global_average_pooling1d_2: GlobalAveragePooling1D
global_max_pooling1d_2: GlobalMaxPooling1D
concatenate_1: Concatenate
dense_1: Dense

Data Preprocessing For Input layer

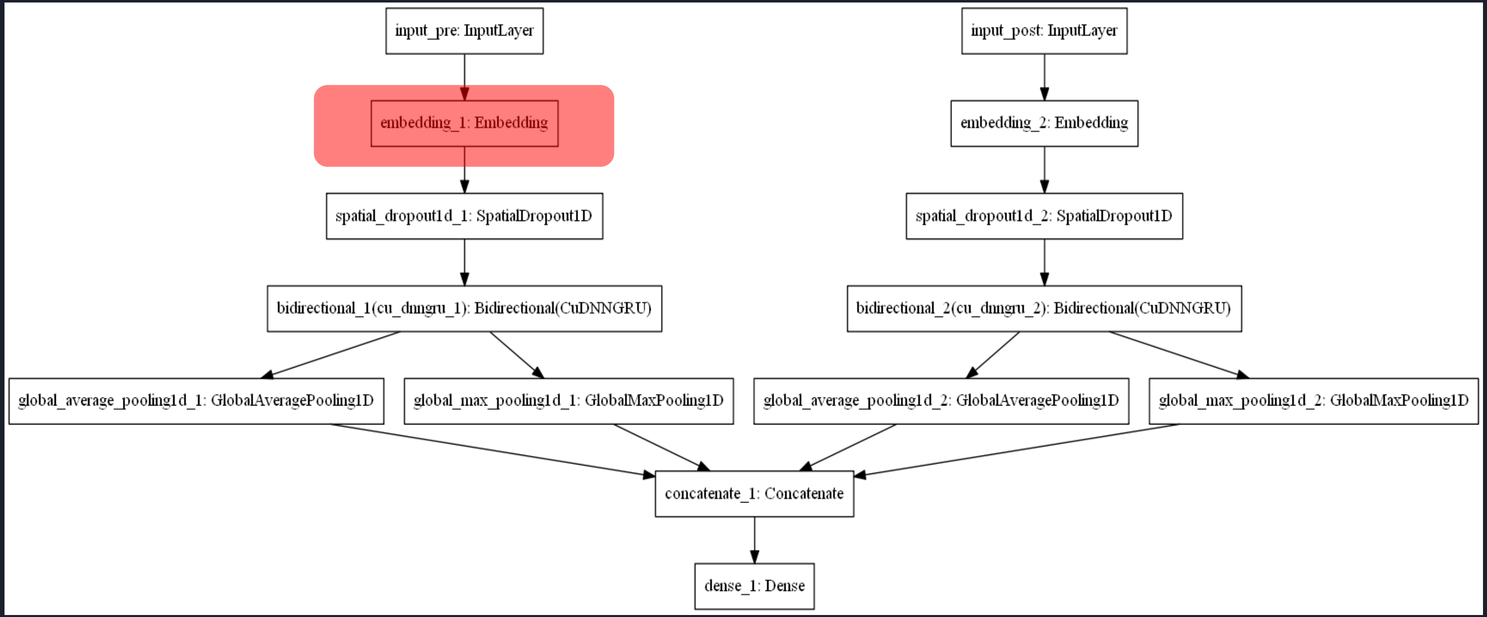
Text to word sequence:

- ✓ 스페이스 기준의 단어 단위 Tokenizing
- ✓ 형태소 분석 x
- ✓ Filter는 기본 Punctuation
- ✓ 소문자 처리
- ✓ OOV 처리 X
- ✓ 상위 80000개 단어 사용

Sequence padding:

- ✓ Sequence maxlen 180
- ✓ Truncating pre
- ✓ Padding pre & post (두가지 padding 방법을 사용해 2개의 Input Set을 생성)

Baseline


input_pre: InputLayer
input_post: InputLayer
embedding_1: Embedding
embedding_2: Embedding
spatial_dropout1d_1: SpatialDropout1D
spatial_dropout1d_2: SpatialDropout1D
bidirectional_1(cu_dnngru_1): Bidirectional(CuDNNGRU)
bidirectional_2(cu_dnngru_2): Bidirectional(CuDNNGRU)
global_average_pooling1d_1: GlobalAveragePooling1D
global_max_pooling1d_1: GlobalMaxPooling1D
global_average_pooling1d_2: GlobalAveragePooling1D
global_max_pooling1d_2: GlobalMaxPooling1D
concatenate_1: Concatenate
dense_1: Dense

Text Embedding

Embedding Matrix를 통해 Text Sequence를 Vector Sequence로 Embedding

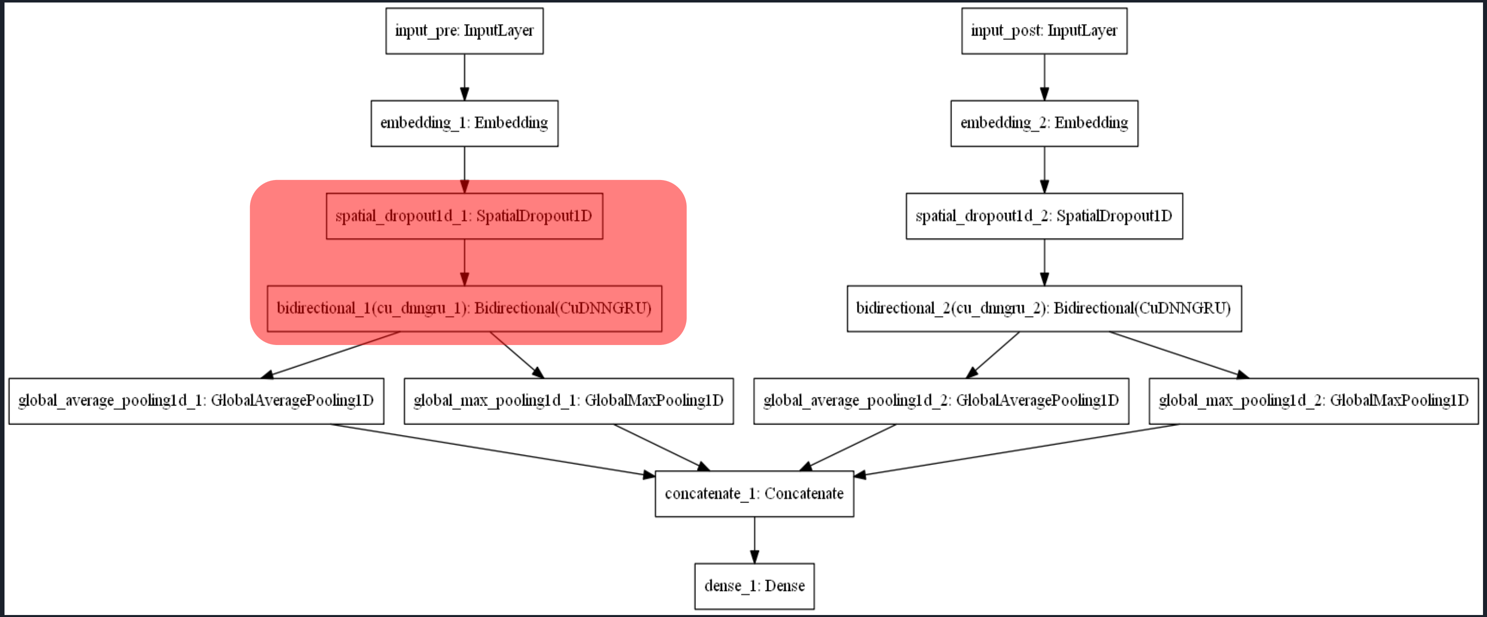
Embedding Matrix Computation:

1. Pre- trained Model을 통해 생성
   - ✓ Word2Vec
   - ✓ Glove
   - ✓ FastText
2. 해당 데이터를 통해 직접 학습하여 생성
3. ~~Online Learning~~

Out of Vocabulary 처리

- ➢ Train데이터에 없던 단어가 Test에 존재 -> 무시 (처리불가)
- ➢ Vocab에 존재하는 단어가 Embedding Matrix에 없을시 -> zero array 생성, random array 생성

Baseline

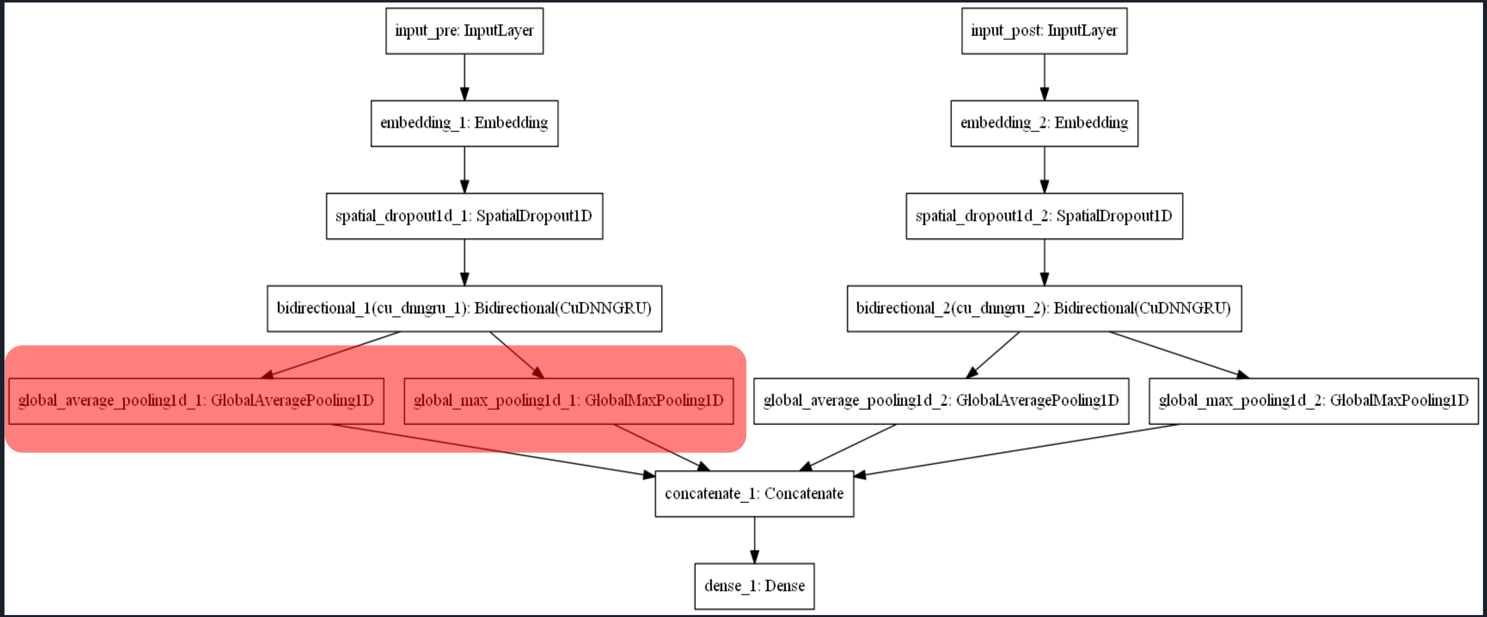

input_pre: InputLayer
input_post: InputLayer
embedding_1: Embedding
embedding_2: Embedding
spatial_dropout1d_1: SpatialDropout1D
spatial_dropout1d_2: SpatialDropout1D
bidirectional_1(cu_dnngru_1): Bidirectional(CuDNNGRU)
bidirectional_2(cu_dnngru_2): Bidirectional(CuDNNGRU)
global_average_pooling1d_1: GlobalAveragePooling1D
global_max_pooling1d_1: GlobalMaxPooling1D
global_average_pooling1d_2: GlobalAveragePooling1D
global_max_pooling1d_2: GlobalMaxPooling1D
concatenate_1: Concatenate
dense_1: Dense

Comment Representation

Bi-direction RNN 구조를 이용해 Comment의 Representation

- ➢ Spatial Dropout
- ➢ Bi-directional Architecture
- ➢ RNN Cell Type:
  - ✓ ~~Vanilla RNN Cell~~
  - ✓ LSTM Cell
  - ✓ GRU Cell

Baseline



input_pre: InputLayer
embedding_1: Embedding
spatial_dropout1d_1: SpatialDropout1D
bidirectional_1(cu_dnngru_1): Bidirectional(CuDNNGRU)
global_average_pooling1d_1: GlobalAveragePooling1D
global_max_pooling1d_1: GlobalMaxPooling1D
input_post: InputLayer
embedding_2: Embedding
spatial_dropout1d_2: SpatialDropout1D
bidirectional_2(cu_dnngru_2): Bidirectional(CuDNNGRU)
global_average_pooling1d_2: GlobalAveragePooling1D
global_max_pooling1d_2: GlobalMaxPooling1D
concatenate_1: Concatenate
dense_1: Dense

Pooling

Pooling Type:

GlobalMaxPooling

GlobalAvgPooling

Global K-max pooling

Other Type pooling layer:

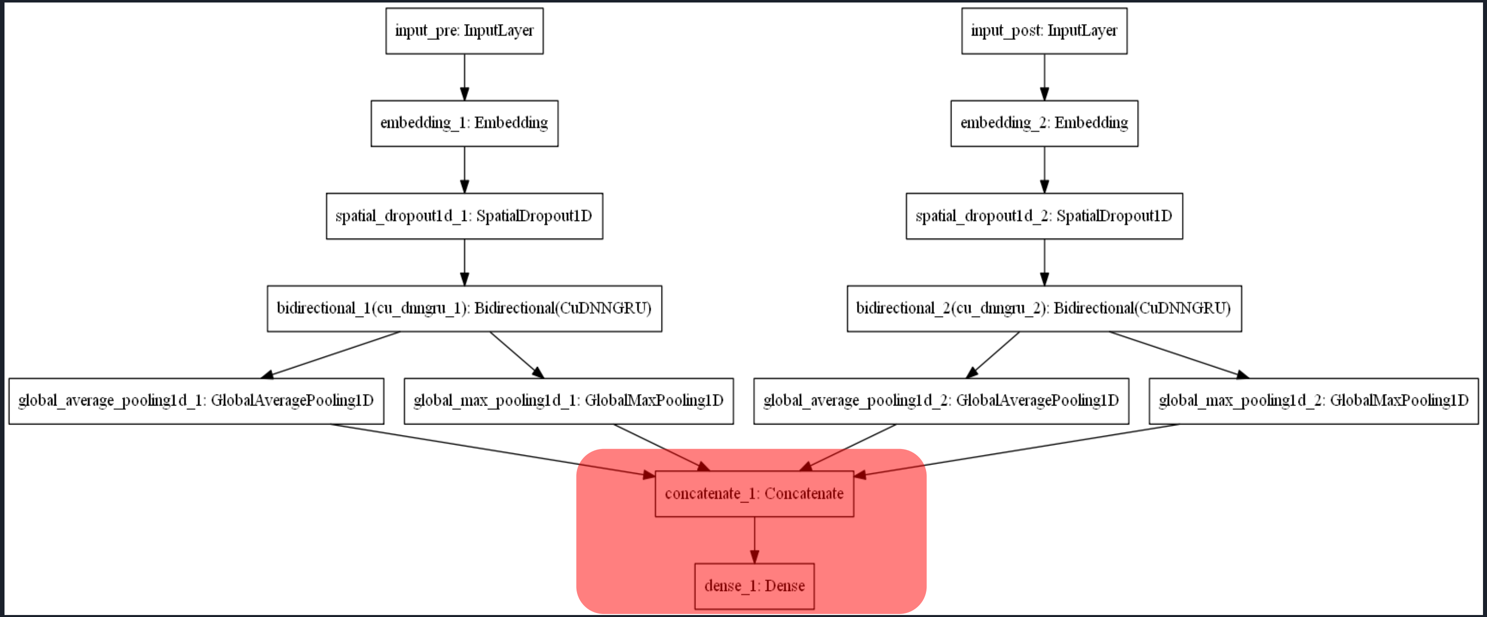
CONV-Pooling

Capsulenet

DPCNN

✓ **Double Pooling**

Baseline


input_pre: InputLayer
input_post: InputLayer
embedding_1: Embedding
embedding_2: Embedding
spatial_dropout1d_1: SpatialDropout1D
spatial_dropout1d_2: SpatialDropout1D
bidirectional_1(cu_dnngru_1): Bidirectional(CuDNNGRU)
bidirectional_2(cu_dnngru_2): Bidirectional(CuDNNGRU)
global_average_pooling1d_1: GlobalAveragePooling1D
global_max_pooling1d_1: GlobalMaxPooling1D
global_average_pooling1d_2: GlobalAveragePooling1D
global_max_pooling1d_2: GlobalMaxPooling1D
concatenate_1: Concatenate
dense_1: Dense

Projection Layer

Concatenate All above output

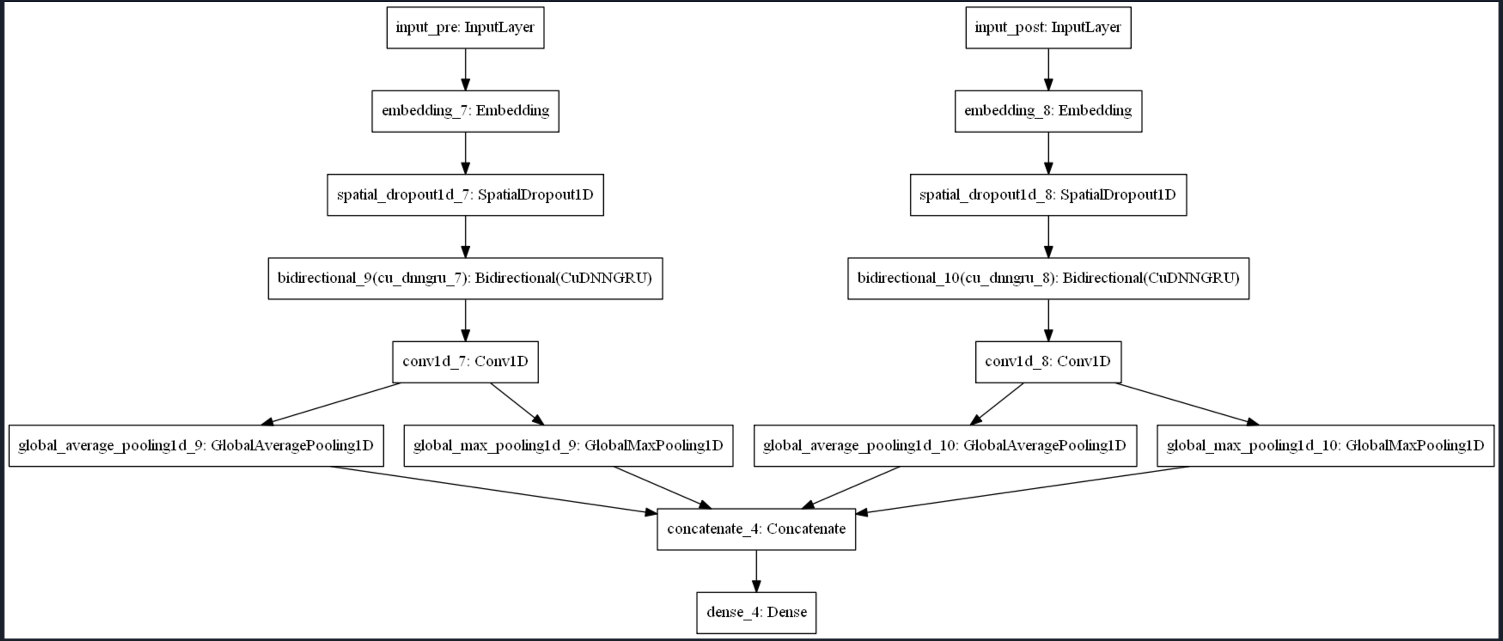
Directly Projection with Dense Layer

Other option:

- Additional Fully Connected Layer
- Direct Link from low level layer

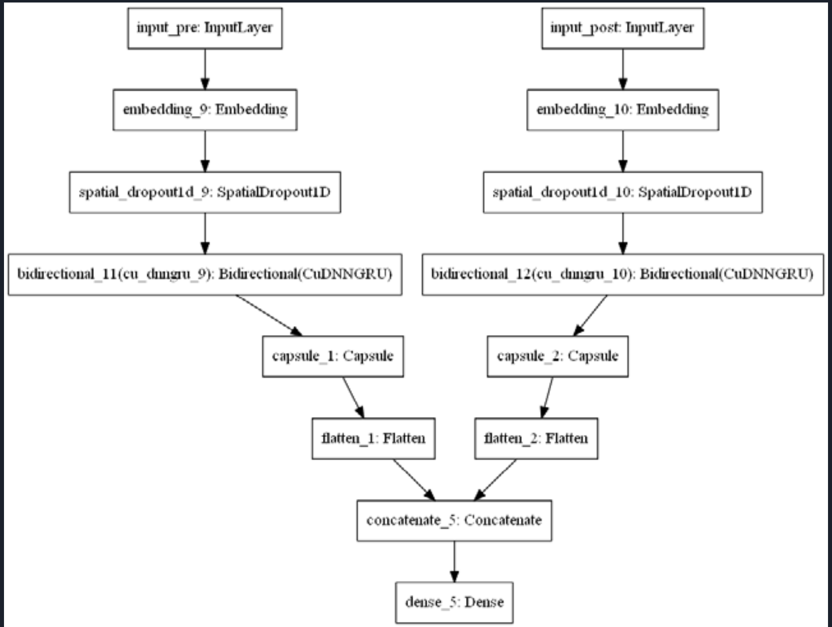
변형 Model Architecture

RNN-CNN-POOLING



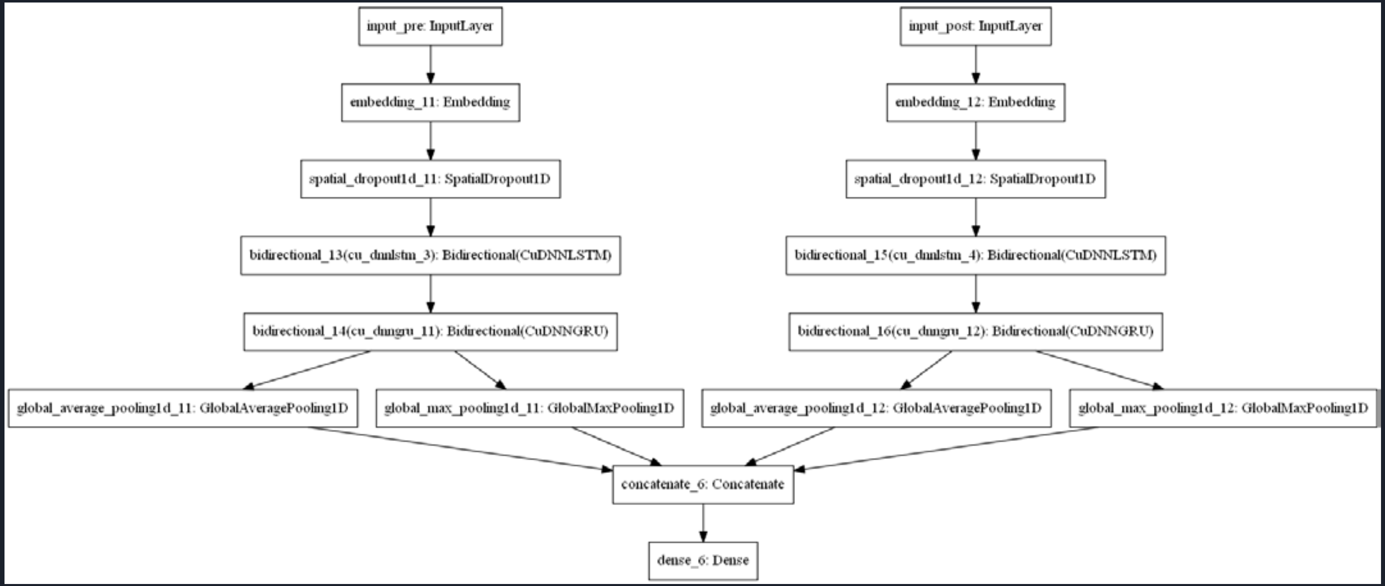
변형 Model Architecture

RNN-CapsNet



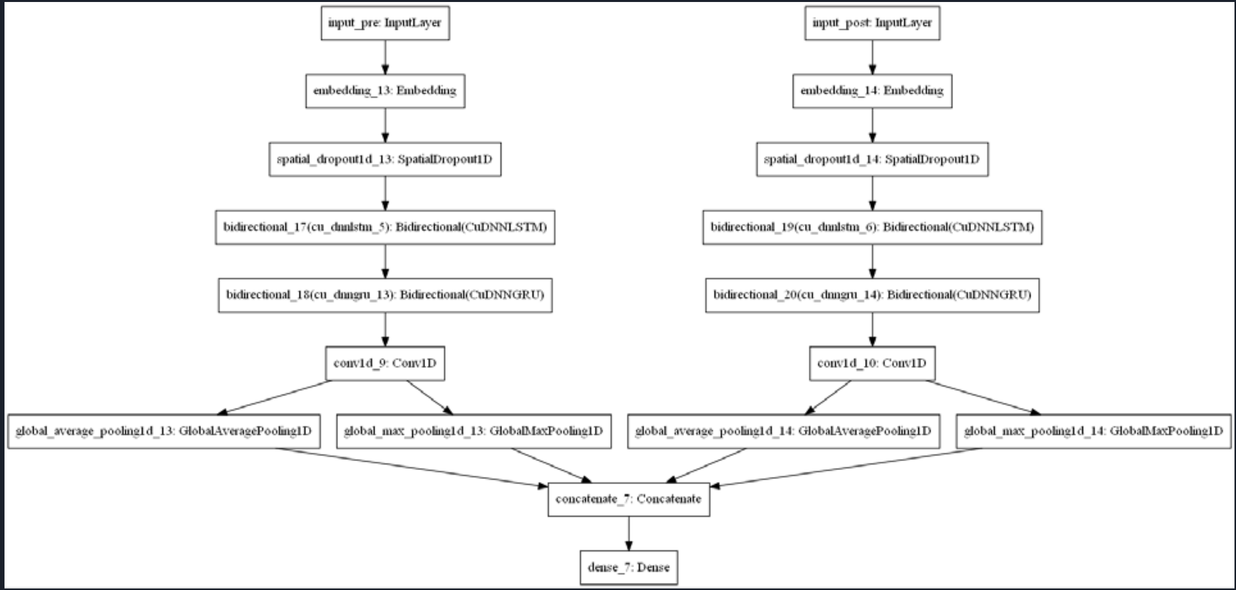
변형 Model Architecture

RNN-RNN-POOLING



변형 Model Architecture

RNN-RNN-CNN-POOLING



Result

Model	Embeddings	Public	Private
RNN	fasttext	0.9850	0.9843
RNN-CNN	fasttext	0.9846	0.9842
RNN-Capsule	fasttext	0.9847	0.9842
RNN-RNN	fasttext	0.9857	0.9847
RNN-RNN-CNN	fasttext	0.9855	0.9845
RNN	glove	0.9853	0.9842
RNN-CNN	glove	0.9854	0.9843
RNN-Capsule	glove	0.9850	0.9841
RNN-RNN	glove	0.9859	0.9851
RNN-RNN-CNN	glove	0.9857	0.9849
Ensemble	fasttext	0.9857	0.9851
Ensemble	glove	0.9860	0.9851
Ensemble	fasttext+glove	0.9862	0.9856
Ensemble	fasttext+glove+lgbm(0.9808/0.9810)	0.9870	0.9865

submission_lastday_v5_real_last.csv 0.9863 0.9871 ☑
24 days ago by YeongTaek Oh

=> 476/4451

Growth

데이터 전처리 (정규표현식, Text2Sequence, Embedding)

하이퍼 파라미터 결정에 대한 직관

Ensemble 및 Stacking의 이해

딥러닝의 학습을 위한 효율적 코드 관리 (Main, Preprocessing, Model, Training Protocol)

분석 방법 및 결과 정리 (GitHub, Kernel)

https://github.com/Timmy-Oh/kaggle_toxic_comment

컴퓨팅 리소스 (메모리, GPU)의 부족으로 인한 시간의 가치

감사합니다