Text Classification via Multiple Models

Yining Hong, 515021910453
evelinehong@sjtu.edu.cn
Research Center of Intelligent Internet of Things, Shanghai Jiao Tong University

Problem: Text Classification

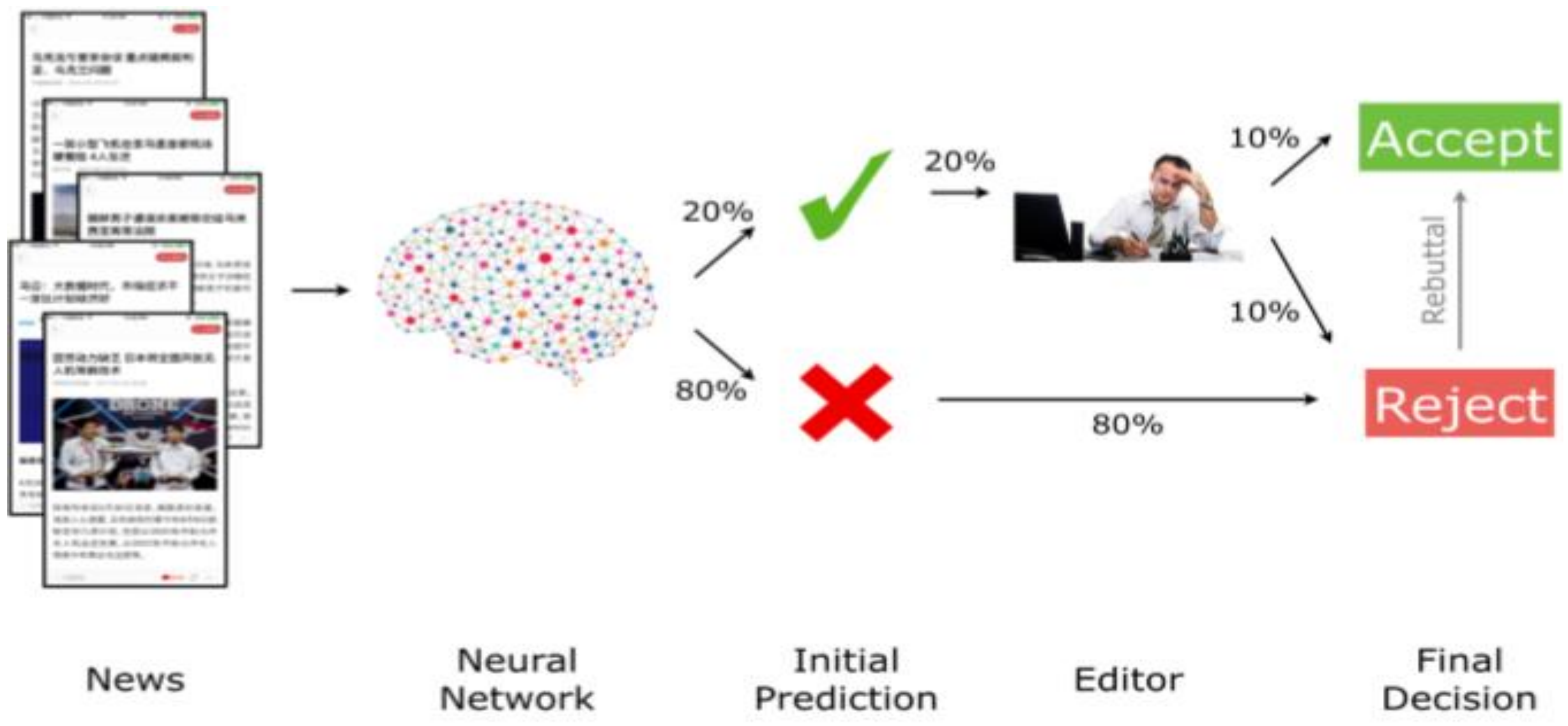
Motivation

- Classify texts by hand is time-consuming Different
- Previous works usually focus on texts in English

Achievement

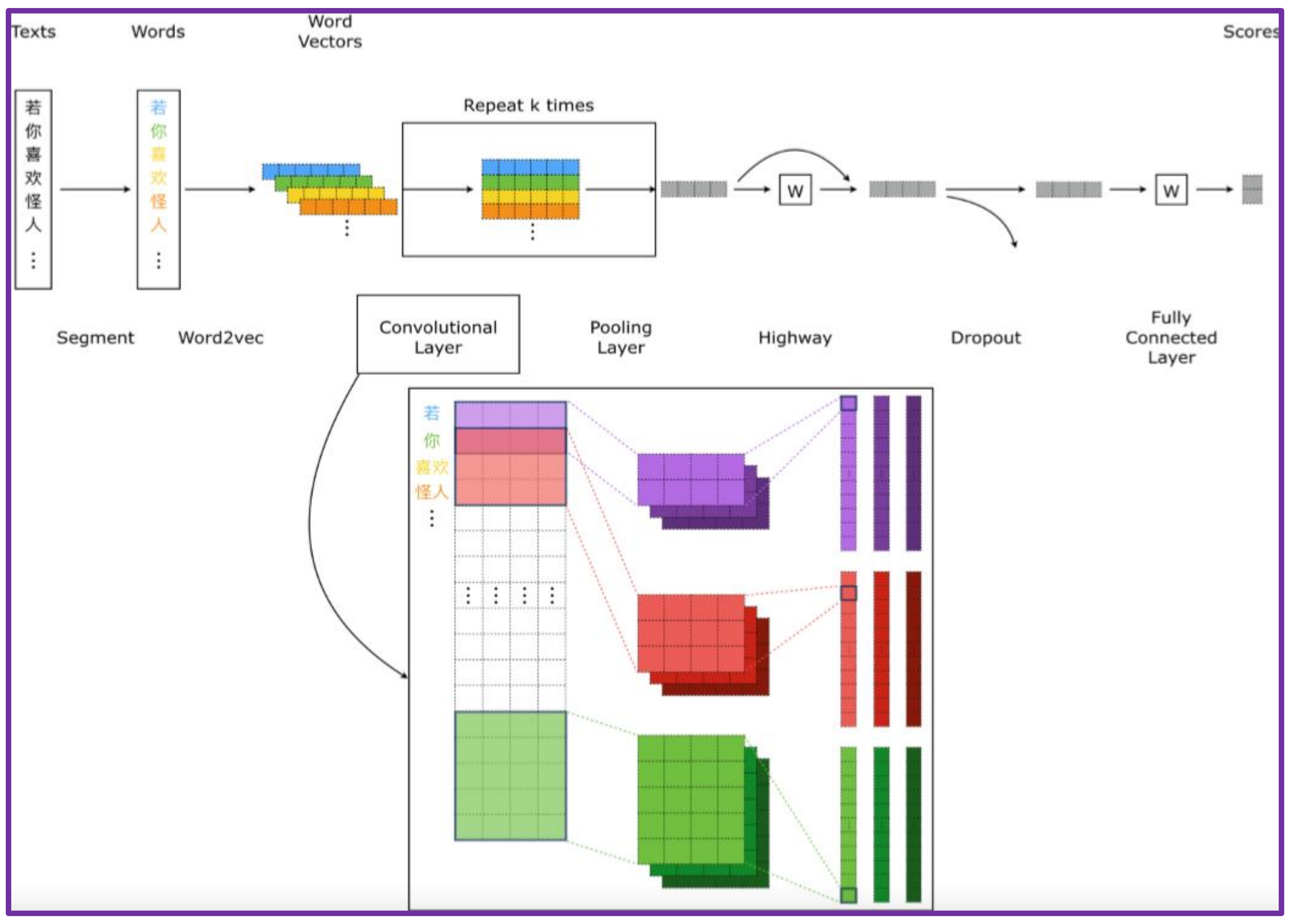
- Works well on Chinese with the help of segmenter
- Achieves the best result on public leaderboard, second best result in private leaderboard
- Utilizes underlying hierarchy information

Possible Application



Model Pipeline

TextCNN

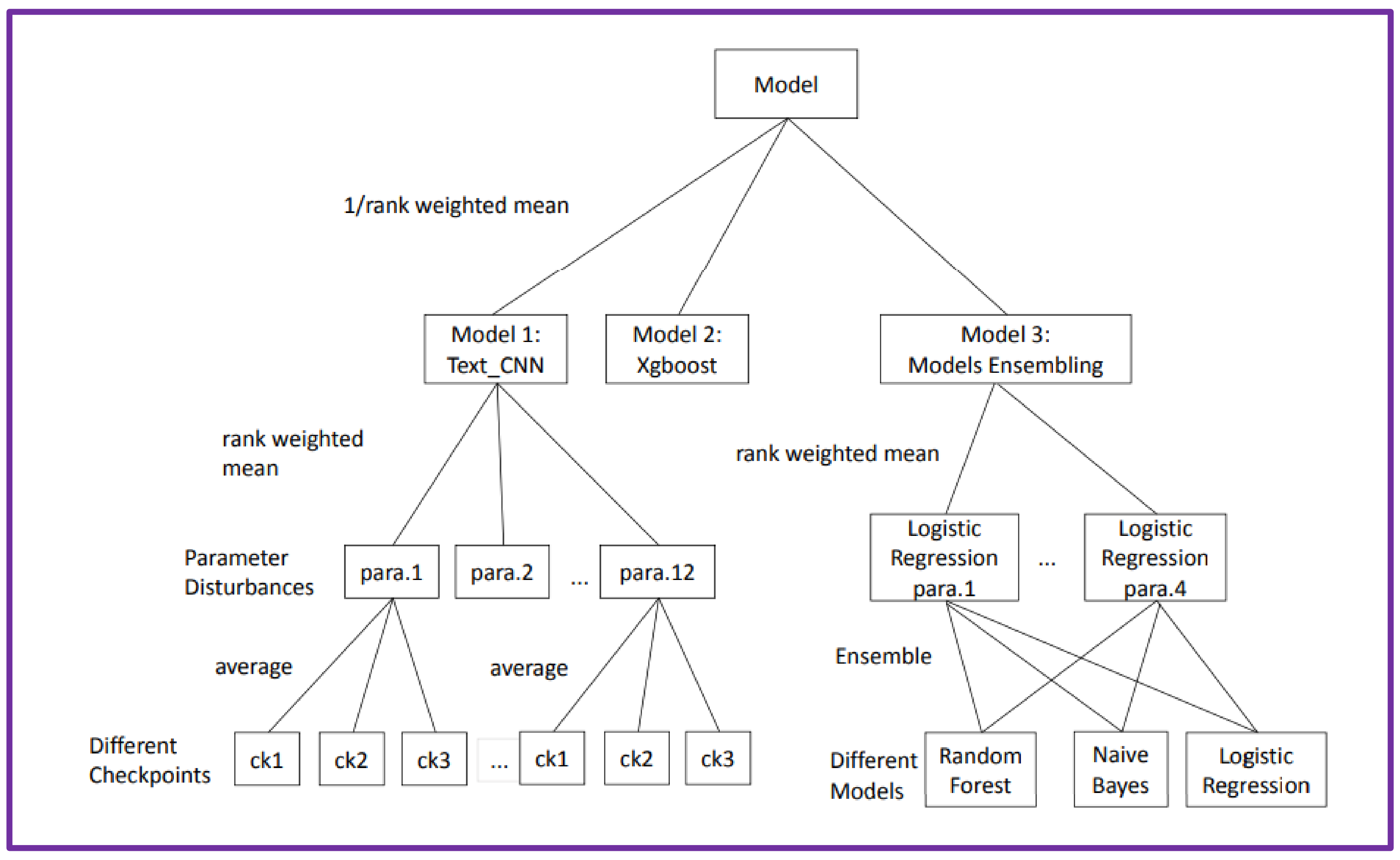


Word Segment: Jieba, for Chinese text segmentation.
Word Embedding: Word2Vec. Use Wiki model to do embedding. The embedding size is 400. Trained for 20 epochs.
Convolutional Layer: 1800 filters. Sizing from 1 to 9.
Highway: $\mathrm{t} = sigmoid(Wy+b)$, $\mathrm{z} = \mathrm{t} \cdot ReLU(Wy+b) + (1-t) \cdot \mathrm{y}$, where y is the input and z is the output.
Dropout: The dropout rate is 0.5.

Xgboost, Logistic Regression and Naïve Bayes

Corpus Segment: Jieba, for Chinese text segmentation.
Corpus to Bunch: Use Bunch structure in SkLearn library.
TFIDF: A numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
Classifiers: Parameters, etc. class weight, need changing.

Ensemble



M1: TextCNN. 40+ results averaging:

- Parameter Disturbances: learning rate, L2 regularization…
- Different Checkpoints: dependent upon validation set accuracy

M2: **Single Xgboost.**

M3: Ensembling of basic models via logistic regression.

- Results of LR, Xgboost, Naïve Bayes as new features
- Inputs new features to upper LR

M: **Top layer ensemble.** Use $\mathrm{score} = \sum_{i=1}^{n} \frac{w_i}{rank_i}$

Experiments

Setup

Metrics: AUC, area under ROC curve.

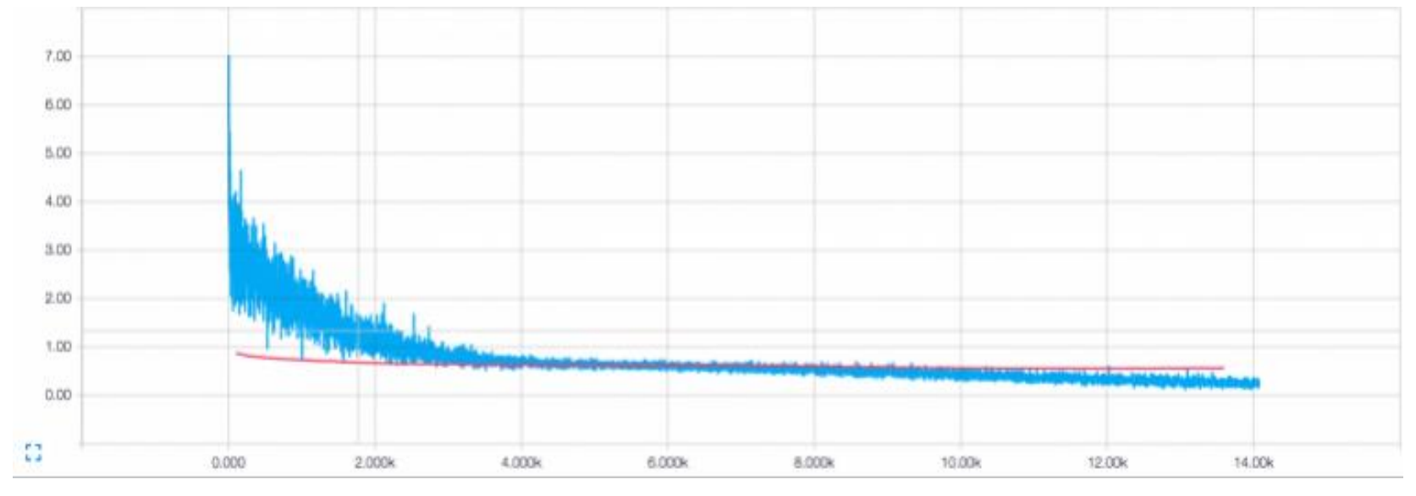
Models Settings (experiments done in different models)

- Primitive: original model, with k=1.
- Batch Normalization: after conv layer, before activation function.
- Data Augmentation: training, take a random crop from the text. During testing, send some fixed, uniformly selected crops to the network and take an average of the predicted scores as output

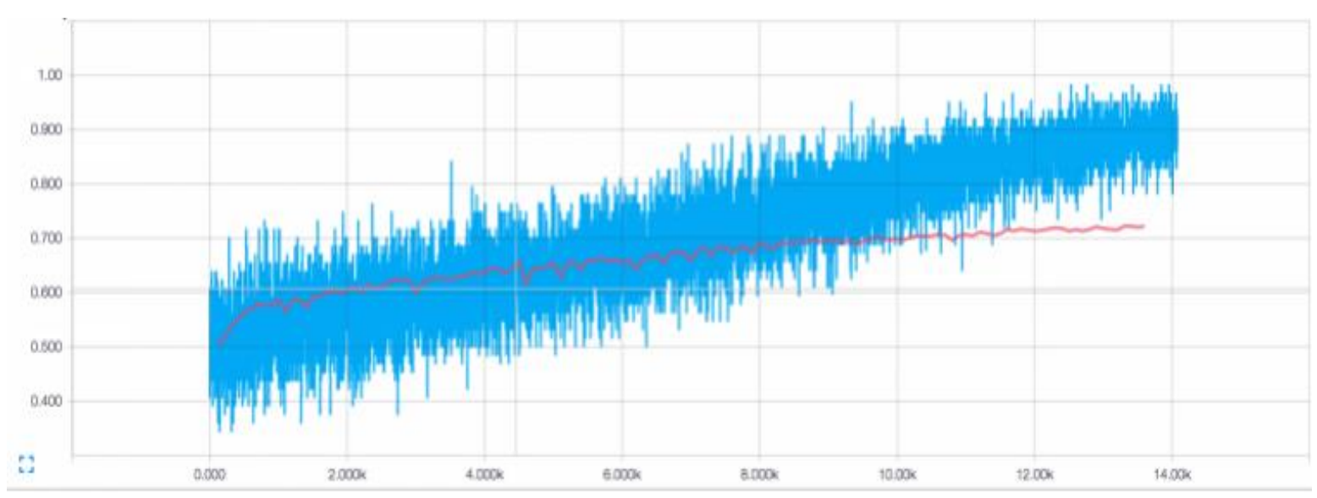
Training Configuration

- The training data is split into training set (97.5%) and validation set (2.5%).
- The batch size is 80.
- A trainable lookup table is used for word embedding so the word embedding is not static
- The initial learning rate is 0.001, and decay every 100 iterations with a decay rate of 0.97.

LOSS



Accuracy



Results

Model	Specific Model Name	Public Score	Ensemble Weight
M1	Mean of Multiple CNN	0.91605	45%
	Single CNN	0.895-0.905	
M2	Xgboost	0.91304	45%
M3	Ensembling of Basic Models	0.90516	10%
	Basic Logistic Regression	0.89506	
	Basic Random Forest	0.88674	
	Basic Naïve Bayes	0.88213	
M	Top Ensembling	0.92150	