Hate Words Detection With Contribution Measure

Team 23: 최우진,김동근,권용빈,이재희

Goal

1. Binary classification of given text (hate or not-hate)

2. Hate word detection in a speech based on quantitative measure of contribution

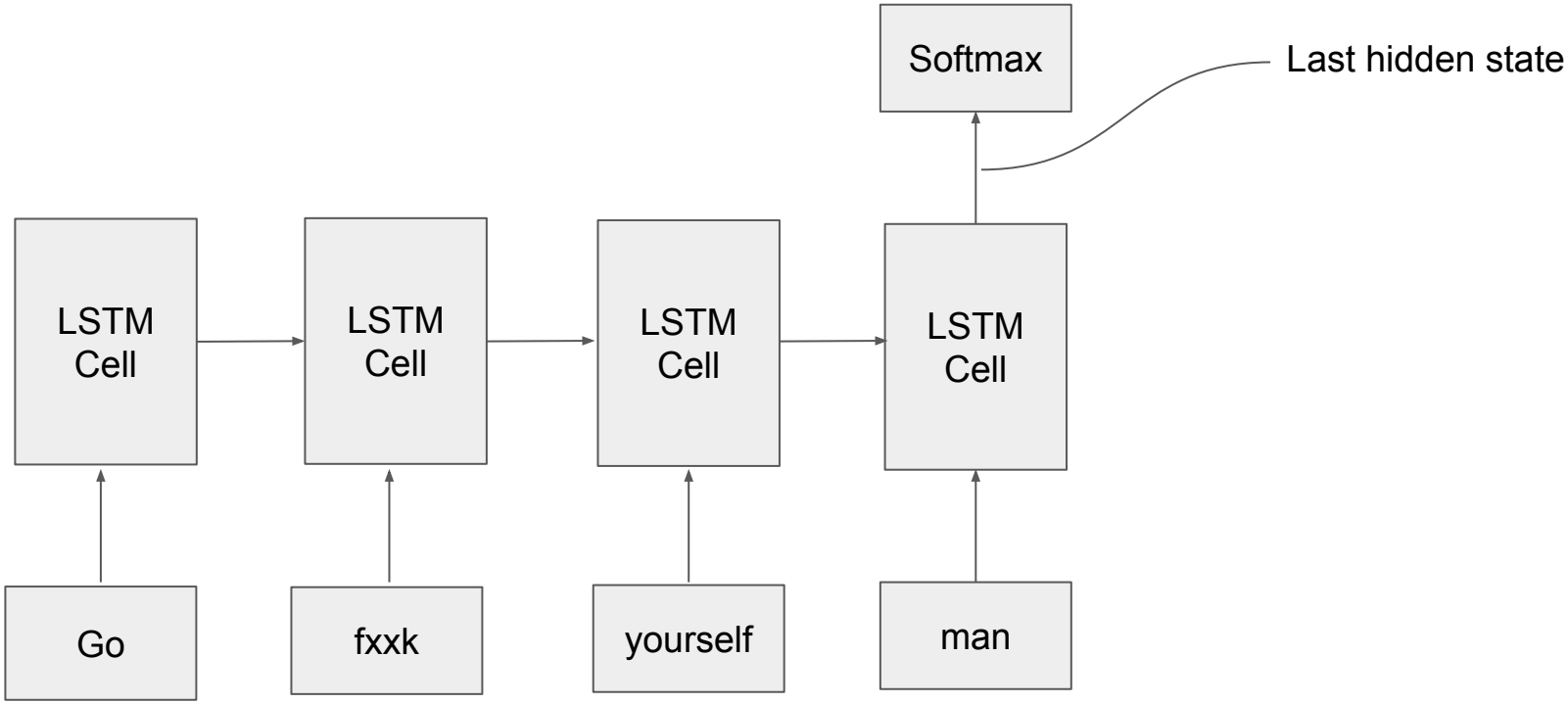
Inductive Bias

Quantitative measure of each words contribution!

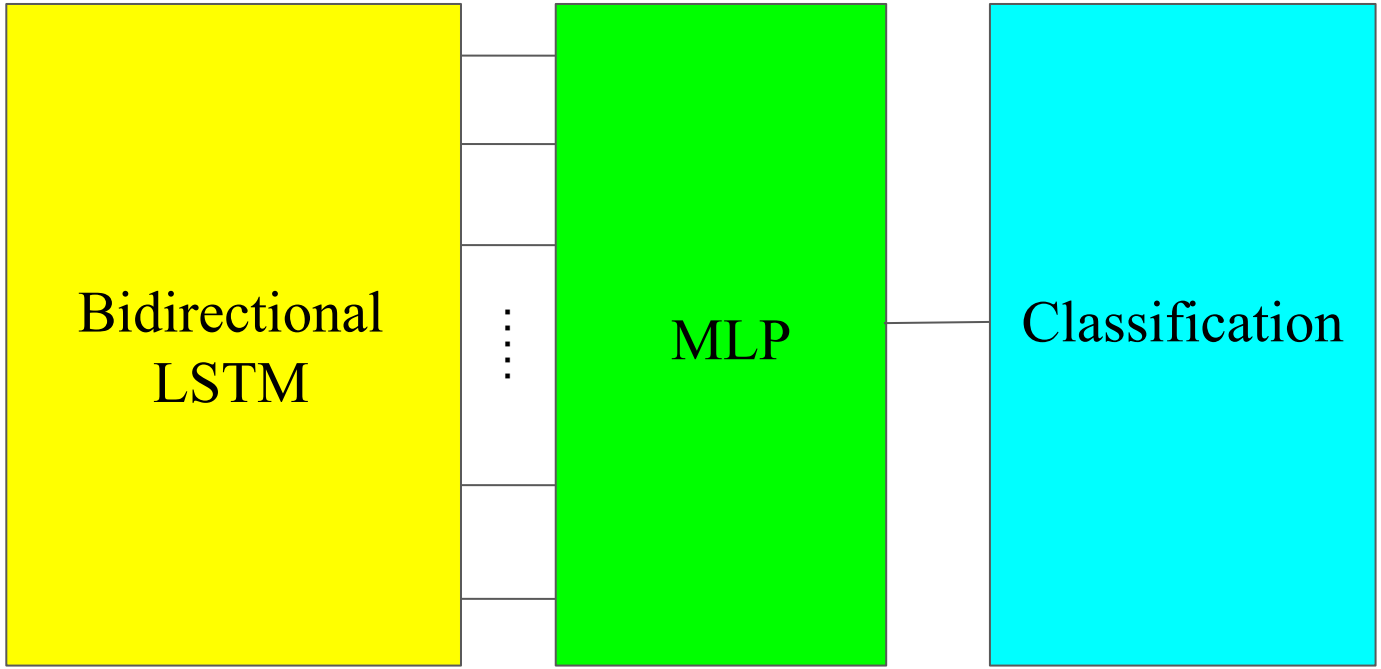
Go fxxk yourself man

-> Which word contributed the most?

Simple text classification



Our Model Structure



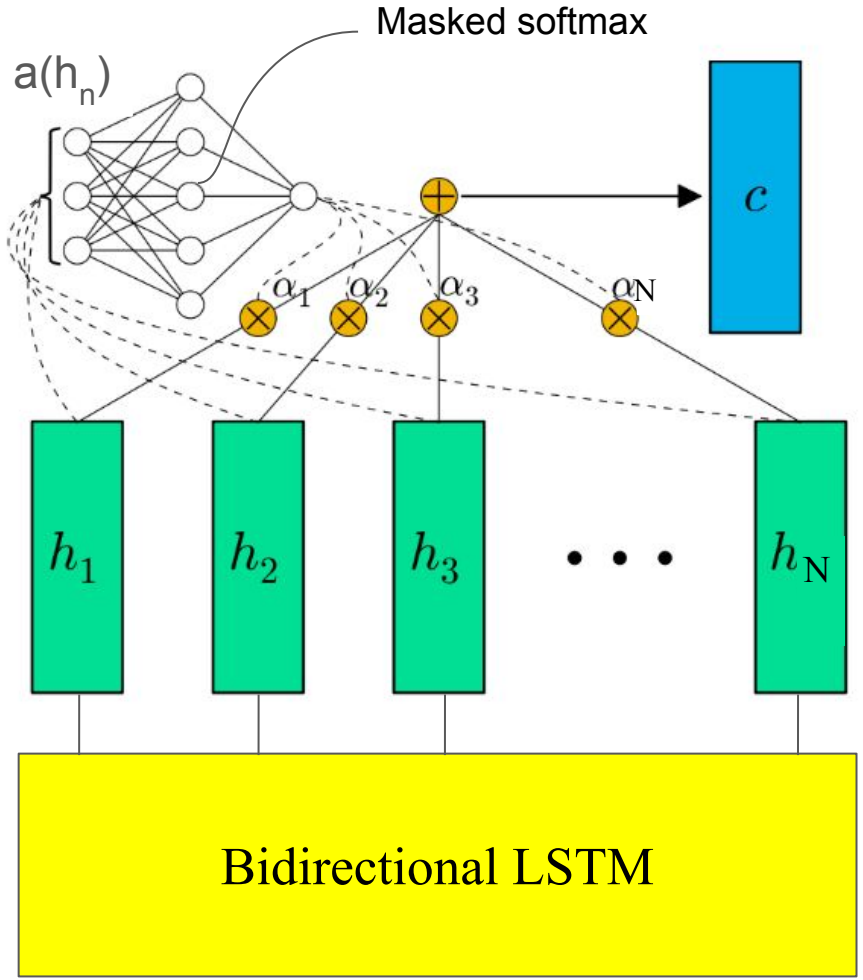
Trainable alpha function

$$\sum_{1}^{N} a_n = 1$$

a(h_n) = quantitative measure of

N^{th} word's contribution

C = weighted sum of words!



Results - Binary Classification of Hate-Speech

Training Data - Labeled dataset,

Dataset Size : Train - 10,424 / Test - 400

Pre-Trained BERT : 5 epochs / Training time - 15 (min)

Performance	**Our Model (large, nonselective data)**	**Pre-Trained BERT**
Accuracy ={(total_correct) / (total_preds)}	0.7600	0.7775

Data (large & non-selective)

Classification Accuracy : 0.7600

-> High Accuracy, poor detection, $a(h_n) \approx 1/\text{Number of words}$ (no clear distinction)

Gays shxts have no rights [0.212, 0.205, 0.204]

Black people should be fxxcked [0.216, 0.211, 0.209]

Data(non-selective)

Non-hate:
Damn, this shxx tastes so fxxking good!

Hate:
You are fxxking lame piece of shxx

Data(selective & small)

->Low Accuracy, better detection

Classification Accuracy : 0.38

$a(h_n)$ much higher than 1/Number of words (clear distinction!)

No one likes fxxking rude black chicks [0.190, 0.187, 0.183]

black ladies don't understand shxt [0.330, 0.199, 0.172]

No one likes fxxking rude whxres [0.225, 0.217, 0.215]

Quality of data

Large, yet non-selective data set

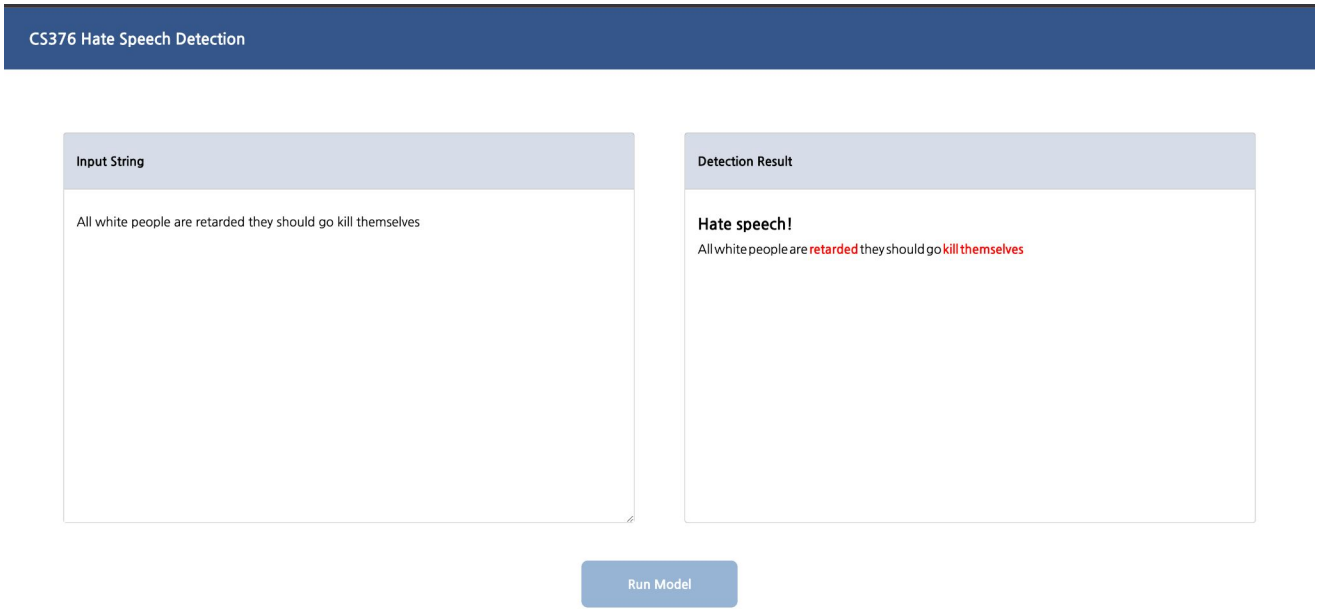
-> High Accuracy, poor detection

Small, yet selective data set

->Low Accuracy, better detection

Training data	**Large, non-selective**	**Small, selective**
Accuracy	0.7600	0.3800
Frequently detected Words (if hate speech)	`[ ('are', 19), ('women', 16), ('be', 10) ]`	`[ ('fxxking', 5), (‘shxt’, 4), ('retarded', 4) ]`

Web service



22 Spring / Team 23
20180679 : 최우진 | 20190074 : 김동근 | 20160502 : 이재희 | 20160791 : 권용빈

Further improvements

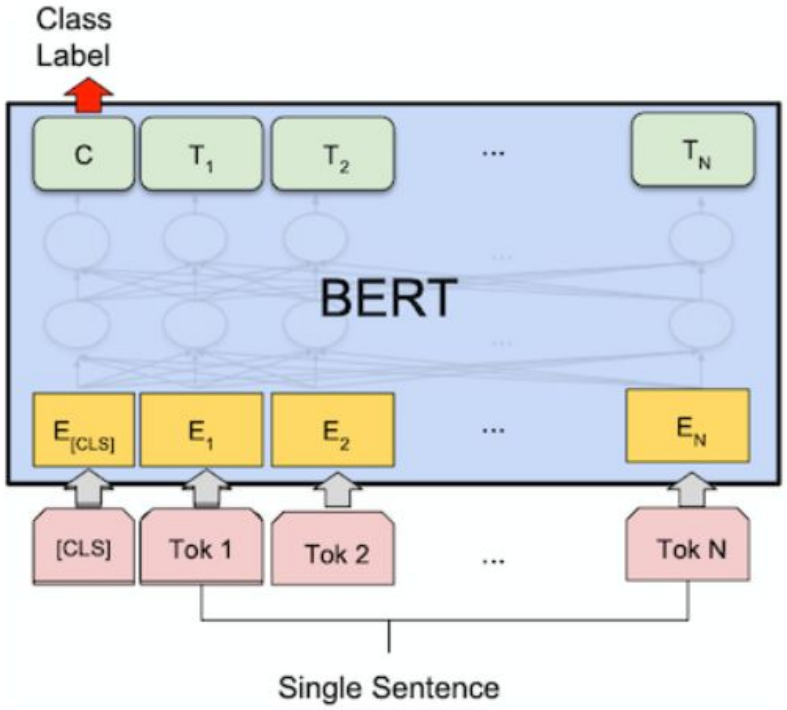
Replace LSTM to BERT!

BERT ?

- An open source machine learning framework based on Transformer
- Google Provides

Better Training Data Set

- Larger, clearer Dataset

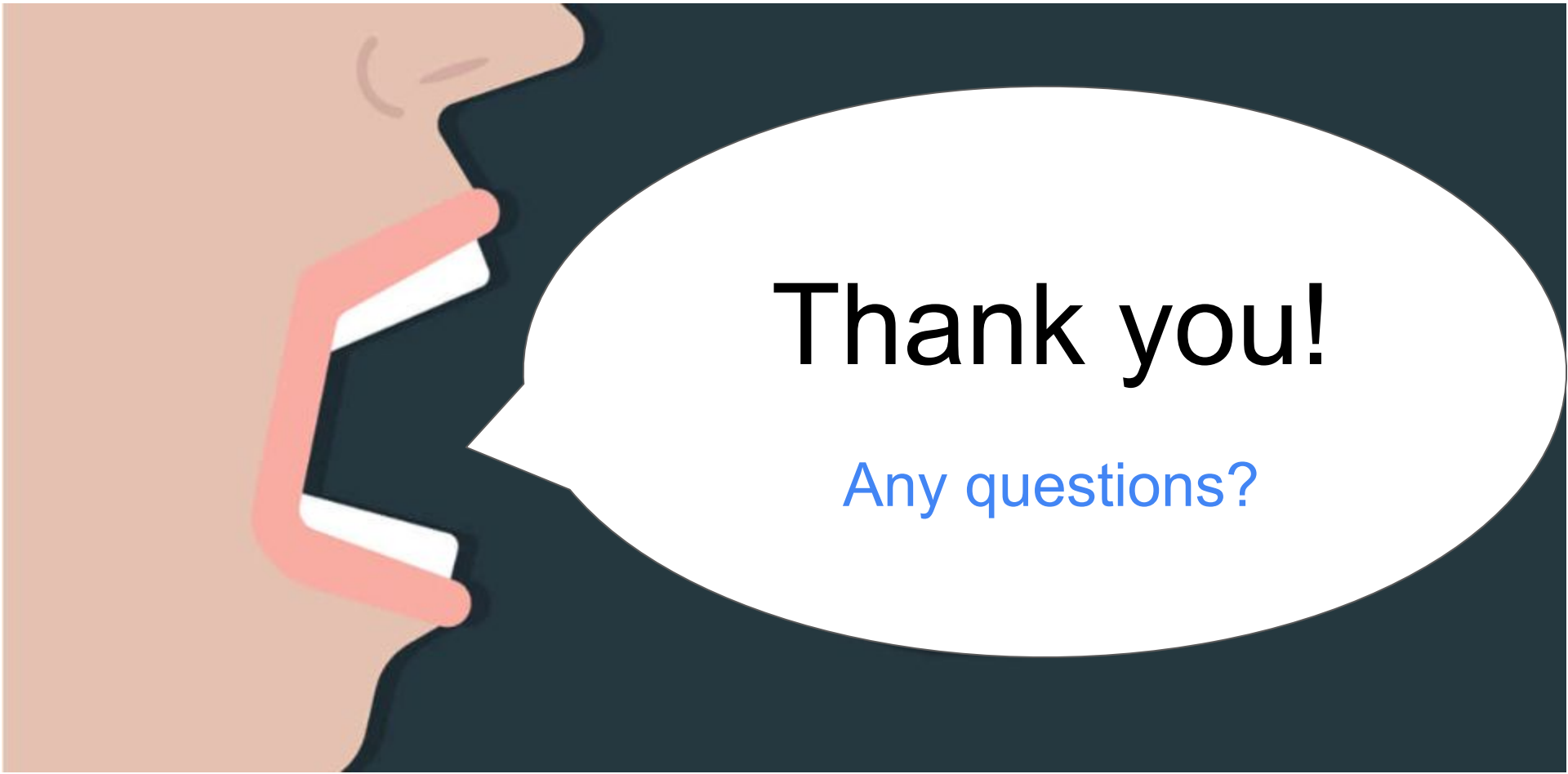


Dealing with stop words

Stop words: “I”, “are”, “the”, “to”, etc…

Possible approaches

I. Remove stop words from results
II. Train / Design architecture so that stop words yield low ~ no contribution


Thank you!
Any questions?