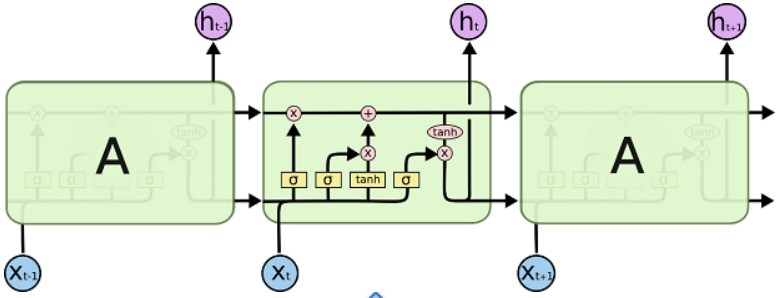
COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

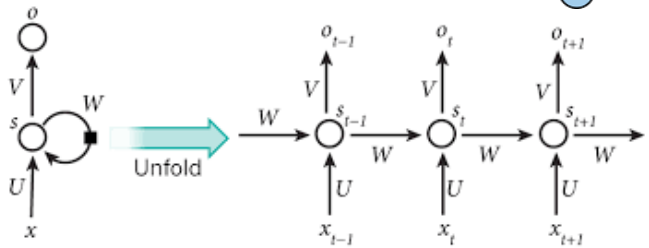
Explaining RNN Predictions for Sentiment Classification

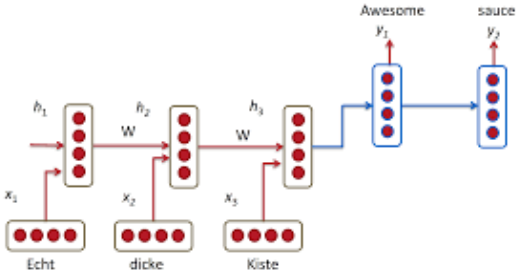
Ninghao Liu and Qingquan Song

2018-11-29

Why Recurrent Neural Network?

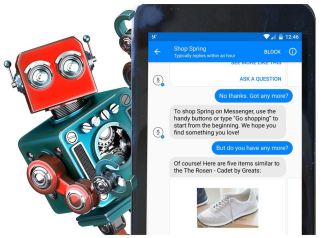






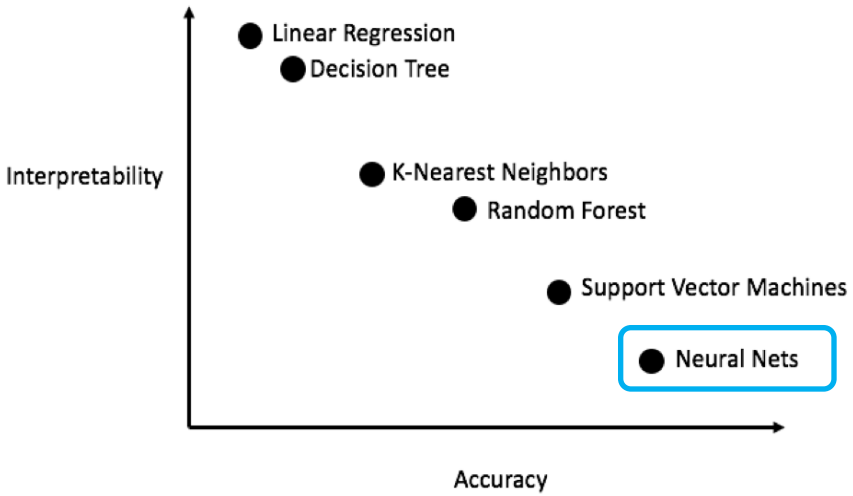
Architectures

Applications



RNNs have made lots of progress in NLP domain.

Why Interpretation?



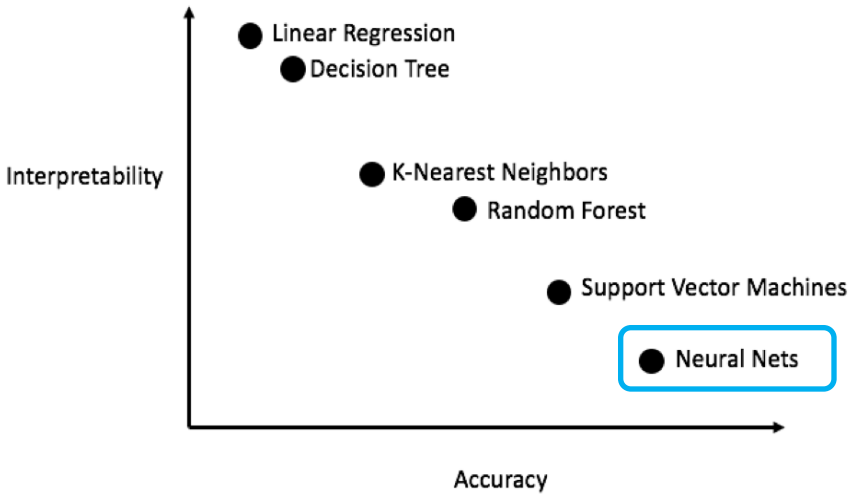
NNs are regarded as *black box*

Self-driving cars

Medical diagnosis

"Interpreting Machine Learning Models". Medium.com, 2017.

Why Interpretation?



NNs are regarded as *black box*

Self-driving cars

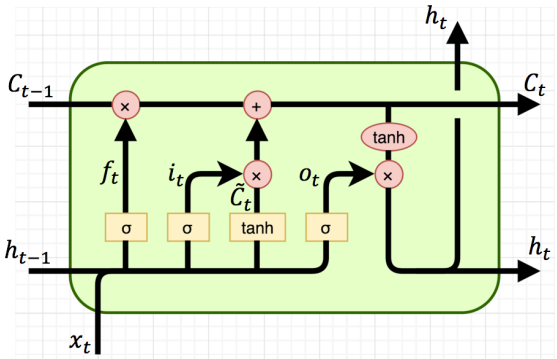
Decisions made by RNNs sometimes could be critical, interpretation can increase trust.

Medical diagnosis

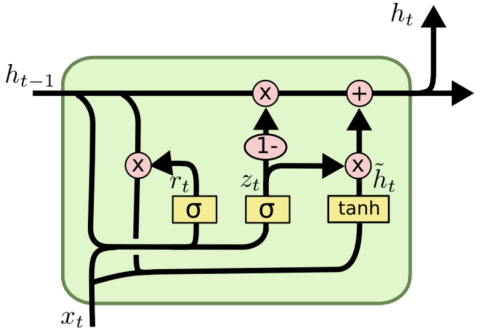
"Interpreting Machine Learning Models". Medium.com, 2017.

Target Models and Settings

Target Models:
Long-Short Time Memory
& Gated Recurrent Unit



(a) Long Short-Term Memory



(b) Gated Recurrent Unit

Dataset: Stanford
Sentiment Treebank

The corpus contains 11,855 sentences extracted from movie reviews. [**Train:** 6920; **Dev.:** 872; **Test:** 1821]

Prediction Accuracy

Metrics \ Models	LSTM	GRU
Precision	0.7985	0.8053
Recall	0.7974	0.8045
F1-Score	0.7972	0.8044
AUC	0.7974	0.8045

- Hidden Unit: 150
- Learning Rate: 1e-3
- Training Epoch: 20
- Loss: Neg. Log-Loss

Developed Interpretation Methods

Naive Explanation (NaiveExp):

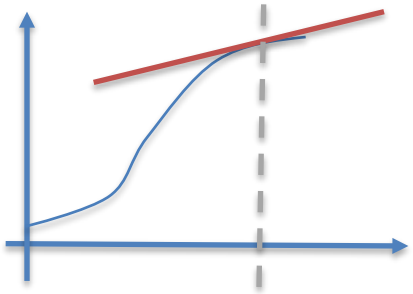
The response score to the *t*-th input word:

$$s(x_t) = W \cdot \Delta h = W \cdot (h_t - h_{t-1})$$

Motivation: Utilize the prediction change on time stamp *t*.
Drawback: Over-simplify the internal forgetting mechanism of RNN.

Vanilla Gradient (VaGrad):

$$s(x_t) = \| \frac{\partial y_c}{\partial \mathbf{x}} (\mathbf{x}_t) \|_2$$

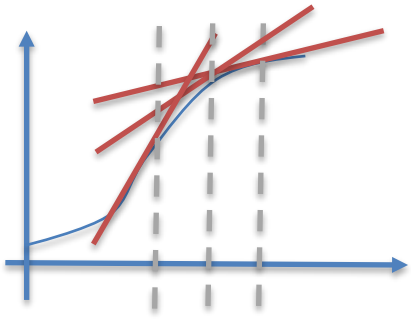


Motivation: Post-hoc explanation using gradient, i.e., local prediction change w.r.t. the input.
Drawback: (1) Easily affected by noise; (2) Does not distinguish between positive and negative contribution.

Developed Interpretation Methods

Integrated Gradient (InteGrad):

$$s(x_t) = \| \frac{(\mathbf{x}_t - \mathbf{x}'_t)}{K} \odot \sum_{k=1}^{K} \frac{\partial y_c}{\partial \mathbf{x}_t} (\mathbf{x}'_t + \frac{k}{K}(\mathbf{x}_t - \mathbf{x}'_t)) \|_2$$



Denoising

Gradient Times Embedding (EmbGrad):

$$s(x_t) = \mathbf{x}_t \cdot \frac{\partial y_c}{\partial \mathbf{x}}(\mathbf{x}_t)$$

Directional (Polarized)

Integrated Gradient Times Embedding (EmbInteGrad):

$$s(x_t) = \mathbf{x}_t \cdot InteGrad(x_t)$$

$$InteGrad(x_t) = \frac{(\mathbf{x}_t - \mathbf{x}'_t)}{K} \odot \sum_{k=1}^{K} \frac{\partial y_c}{\partial \mathbf{x}_t} (\mathbf{x}'_t + \frac{k}{K}(\mathbf{x}_t - \mathbf{x}'_t))$$

Visualization of Interpretation Results

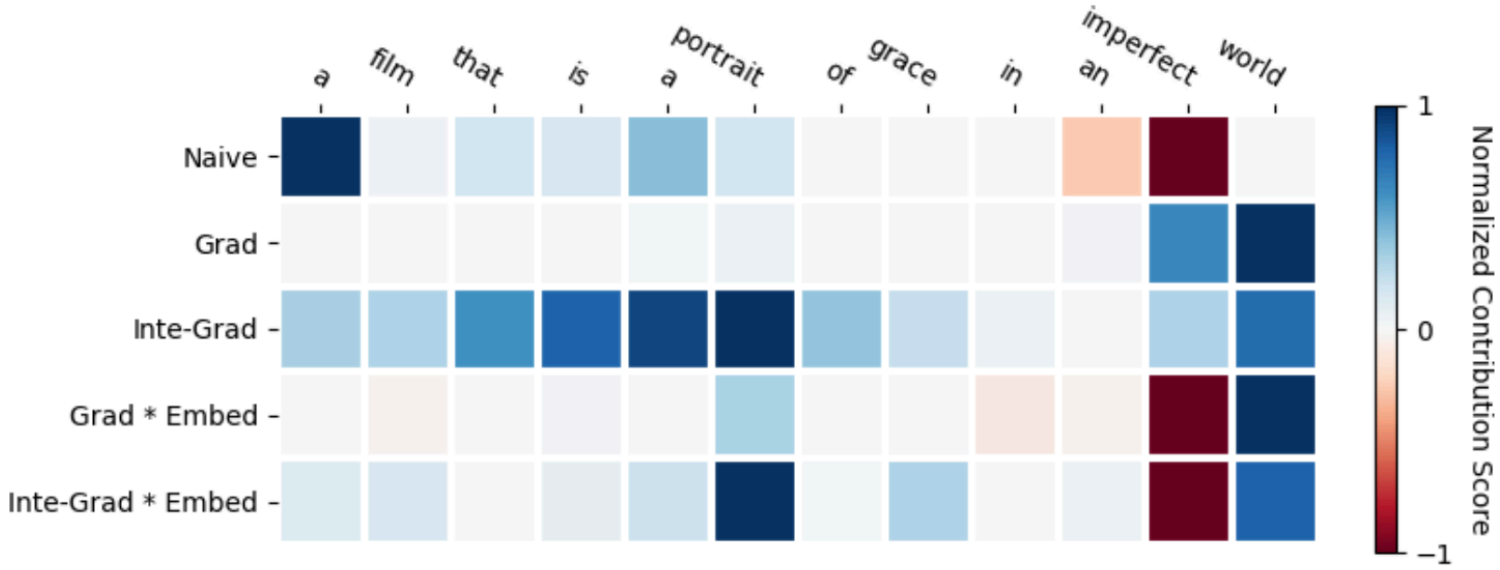


Figure 2: GRU: Word-level attribution heatmap for five methods. The sentence explored possesses positive sentiment and is corrected predicted by the GRU. Blue and red color denote positive and negative contribution of a word to the prediction, respectively.

Quantitative Evaluation of Interpretation

Main idea: Adversarial Attacking

Perturb the embedding vectors of important words, and measure the change of prediction.

Methods \ Models	LSTM	GRU
NaiveExp	0.00781	0.00111
VaGrad	0.00090	0.00374
InteGrad	-0.00112	0.00325
EmbGrad	0.04731	0.04097
EmbInteGrad	0.04200	0.03708

Conclusion

- Some interpretation techniques for other models (e.g., CNN) could be utilized
- However, the uniqueness of NLP and RNN may prevent the directly adoption of some interpretation methods
- The design of evaluation metrics for interpretation is still a very challenging task
- More exploration on the intermediate representation space may be one direction

EmoContext: Contextual Emotion Detection in Text Conversation

Shaolong Chen
Manasa
Jiayi Shen
2018.11.27

Introduction

- ❏ Our task is mainly about detecting different emotions from Text.
- ❏ Includes happy, sad and angry.

id	turn1	turn2	turn3	label
156	You are funny	LOL I know that. :p	😊	happy
187	Yeah exactly	Like you said, like brother like sister ;)	Not in the least	others

Introduction

Applications :

- ❏ Product and Service reviews.
- ❏ Result prediction.
- ❏ Decision making.

Data

The dataset is provided by CodaLab Competition.

The training data and test data.

- Trainning Data:
  1. 15K data for emotion classes.
  2. 15K data for other classes.
- Test data: 5K without label.

Models

- We will use NaiveBayes as the baseline of our model.

- Our main methods experiment with LSTMS in several ways.
- 1)Usig different word Embeddings
- 2)Different LSTM structures

Evaluation

Our evaluation, is same as the one followed by the SemEval task evaluation policy. Here, we are considering micro-averaged F_1 score for three emotion classes namely, happy, sad and angry for the predictions made on test set that happen in real world. The **precision** and **recall** are calculated as follows:

$$P_\mu = \frac{\Sigma TP_i}{\Sigma(TP_i + FP_i)} \quad \forall i \in Happy, Sad, Angry$$

$$R_\mu = \frac{\Sigma TP_i}{\Sigma(TP_i + FN_i)} \quad \forall i \in Happy, Sad, Angry$$

Where TP_i, FP_i, FN_i are true positives, false positives and false negatives of a given class i.

Naive Bayes

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} \tag{3.1}$$

where,

$P(H|E)$- posterior probability of the hypothesis.

$P(H)$- prior probability of hypothesis.

$P(E)$- prior probability of evidence.

$P(E|H)$- conditional probability of evidence of given hypothesis.

Or in a simpler form:

$$Posterior = \frac{(Prior) \times (Likelihood)}{Evidence} \tag{3.2}$$

Results and drawbacks

	Precision	Recall	F1
Happy	0.719	0.759	0.738
Sad	0.825	0.914	0.867
Angry	0.826	0.93	0.874
Average	N/A	N/A	0.826

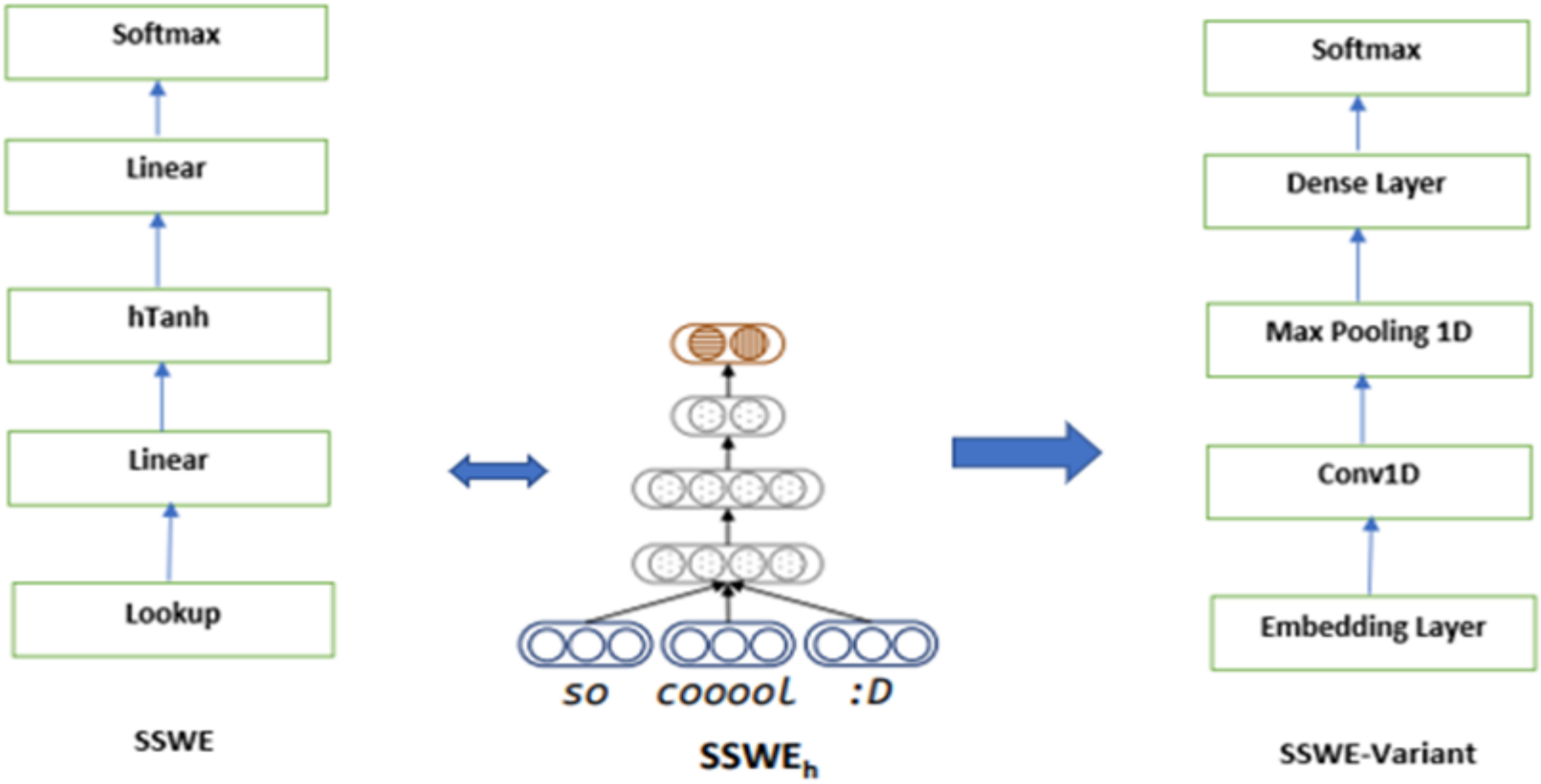
The F1-Score obtained when submitted to coda-lab is 0.41, which is pretty low.

LSTM-Models

- The models are simple LSTM networks with different kinds of Input Embeddings.
- Two Input Embeddings:
- 1) GloVe and
- 2) A variant of Sentiment Specific Word Embeddings (SSWE).
- Glove is an unsupervised learning algorithm for obtaining vector representation for words. We have used, pretrained word vectors of 100 dimensions.

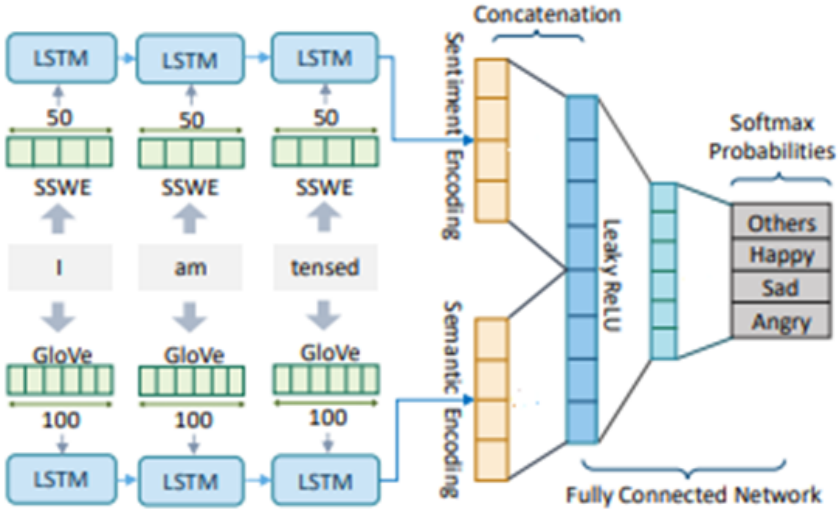
Sentiment Specific Word Embeddings (SSWE)

- Mainly based on the unsupervised C&W model.
- Slide across sentence with a window of n, rather than whole sentence.
- We then predict the sentiment polarity using a neural network and learn the embeddings associated.


Softmax
Linear
hTanh
Linear
Lookup
SSWE
so cooool :D
SSWE$_h$
Softmax
Dense Layer
Max Pooling 1D
Conv1D
Embedding Layer
SSWE-Variant

LSTM+ GloVe +SSWE-variant embeddings

•GloVe gives us semantic information and SSWE-variant can give us the sentiment information, we constructed an LSTM network with two LSTM layers



•Test set has a real life distribution, which is about 4% each of 'angry', 'sad', 'happy' class and the rest is 'others' class which will significantly change the F1 scores. To try to reduce this impact, we have oversampled the others class .

Model	LSTM-hidden Units	Embeddings dimensions	Micro-F1-Score
LSTM+Glove+SSWE-var	128	SSWE-50 dimensions GloVe-100 dimensions	0.711058
LSTM +SSWE-var	128	SSWE-50 dimensions	0.675174
LSTM+GloVe	128	GloVe-100 dimensions	0.68248
LSTM+Glove+SSWE-var With no oversampling of Others class	128	SSWE-50 dimensions GloVe-100 dimensions	0.659857
LSTM+Glove+SSWE-var	248	SSWE-50 dimensions GloVe-100 dimensions	0.691193

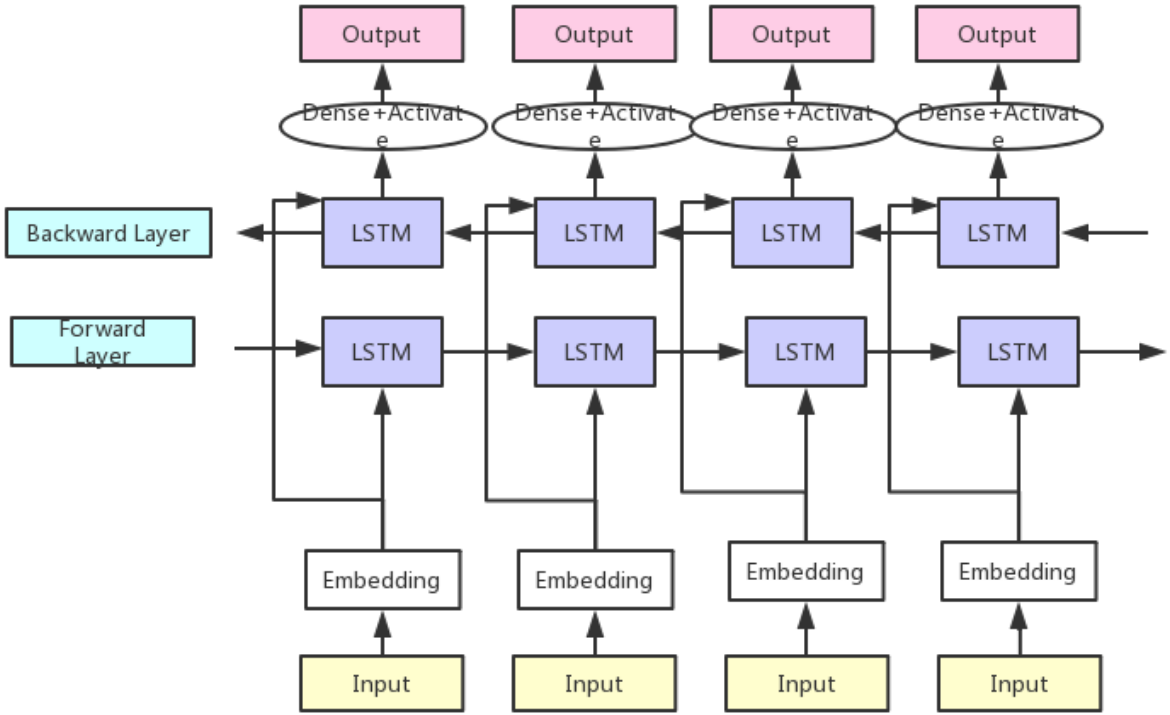
- The accuracies obtained for the below models are: 0.8449, 0.8446, 0.8694 respectively

	Happy Precision	Recall	F1	Sad Precision	Recall	F1	Angry Precision	Recall	F1	Avg F1
LSTM-Glove	0.855	0.750	0.799	0.838	0.850	0.844	0.845	0.819	0.832	0.8281
LSTM-SSWE	0.797	0.782	0.790	0.834	0.859	0.846	0.845	0.790	0.816	0.8203
LSTM-SSWE-GLOVE	0.889	0.776	0.829	0.886	0.862	0.846	0.865	0.867	0.836	0.852

Conclusions

- The LSTM-SSWE(Variant)-GloVe performs better than the other two in both test sets mentioned above but with a small margin. The SSWE (Variant) if trained on a bigger set of data with all emotion labels, might have performed much better
- not considered the dialogue/conversation structure

Bidirectional LSTM



The bidirectional LSTM has almost the same performance than LSTM ,which may be because that the forward information and backward information of dialogue data may not vary too much.

Epoch = 10

Validation Result

	Happy			Sad			Angry			Avg F1
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	
Validation Set	0.912	0.797	0.851	0.926	0.835	0.878	0.929	0.843	0.884	0.8726

Test Result

Here test set is provided by the CodaLab Competition, which only calculates the F1 score.

	Accuracy	Precision	Recall	F1
Test Set	-	-	-	0.695652

Epoch = 30

Validation Result

	Happy			Sad			Angry			Avg F1
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	
Validation Set	0.988	0.843	0.909	0.921	0.846	0.882	0.961	0.859	0.907	0.899

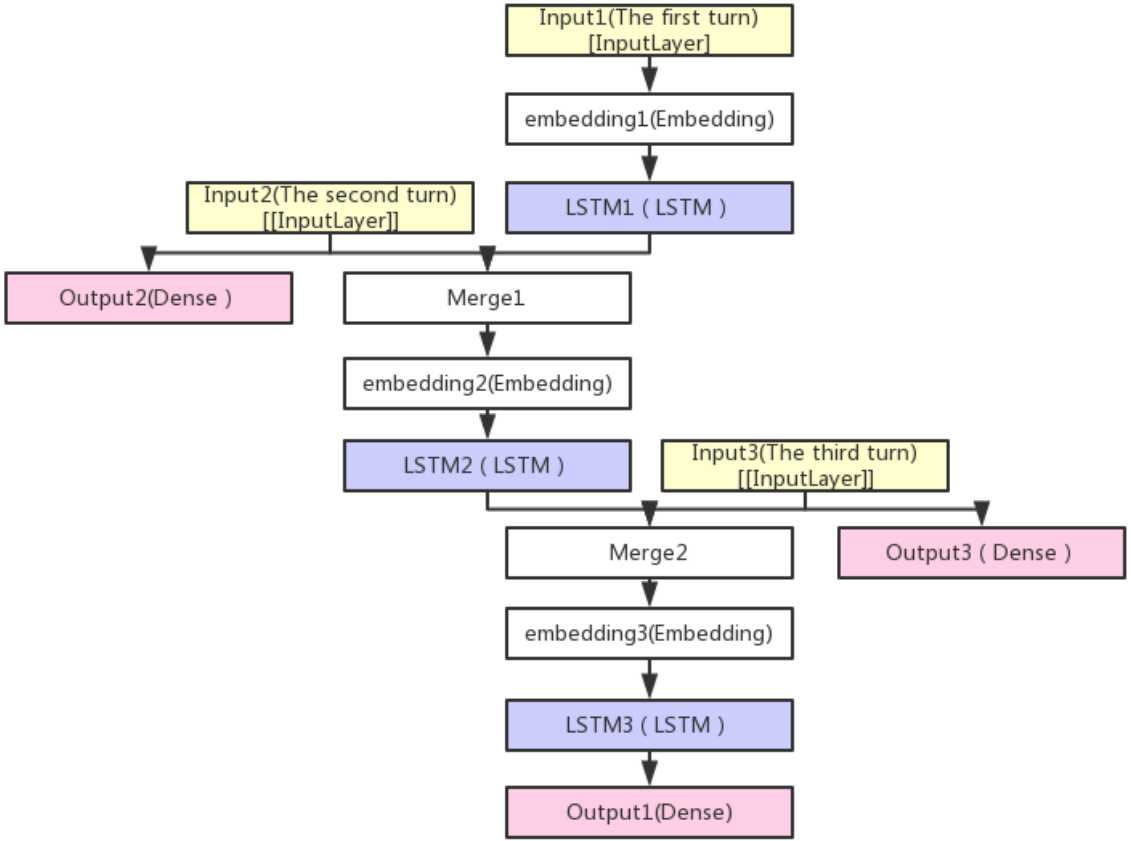
Test Result

Here test set is provided by the CodaLab Competition, which only calculates the F1 score.

	Accuracy	Precision	Recall	F1
Test Set	-	-	-	0.633267

To what degree should train our model?

Multi Input Multi Output (MIMO) LSTM



Validation Result

	Happy			Sad			Angry			Avg F1
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	
Validation Set	0.834	0.615	0.708	0.897	0.658	0.759	0.838	0.677	0.749	0.741

Test Result

	Accuracy	Precision	Recall	F1
Test Set	-	-	-	0.674074

Partial Data + LSTM

Use only the third turn data + LSTM

Validation Result

	Happy			Sad			Angry			Avg F1
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	
Validation Set	0.848	0.661	0.743	0.900	0.733	0.808	0.836	0.707	0.766	0.774

Test Result

	Accuracy	Precision	Recall	F1
Test Set	-	-	-	0.669056

Analysis and Thoughts

1. Make the model lighter.
2. Increase the usage of two side outputs. This also calls for the improvement of the structure.
3. A good way may be training the word vectors inside the given corpus.

Conclusions

1)Naive Bayes has performed bad on the test data giving only accuracy of 0.41 while the other models performed comparatively good.
2) The MIMO model, even though considered the conversation model, it still lacks behind in its F1-Score. It has large number of parameters compared to any other model.
3)The LSTM+GloVe+SSWE(Variant) model gave an F1-Score of 0.71 and is highest achieved out of all these models
4)We were able to make the models perform with results F1-Score of 0.69 approximately, but to improve this further to 0.70 and more was pretty difficult.

Q&A

ENGINEERING
TEXAS A&M UNIVERSITY

Toxic Comment Classification (CSCE 638)

By Abhishek Das, Niti Jain and Varsha Khanna

Guided by Prof. Ruihong Huang

Table of Contents

ENGINEERING
TEXAS A&M UNIVERSITY

Introduction

- Social media platforms are increasingly known for issues of abuse and online harassment.
- Determining if a comment is toxic is difficult and a time consuming process.
- If public forums could automate the process of identifying online harassment, we would save both time and encourage better online conversations.

Problem Statement

- Classify the Wikipedia's talk page comments into the following groups : toxic, obscene, severe toxic,insult, threat and identity hate.

- It is a multilabel classification problem i.e. the data could belong to more than one label simultaneously.

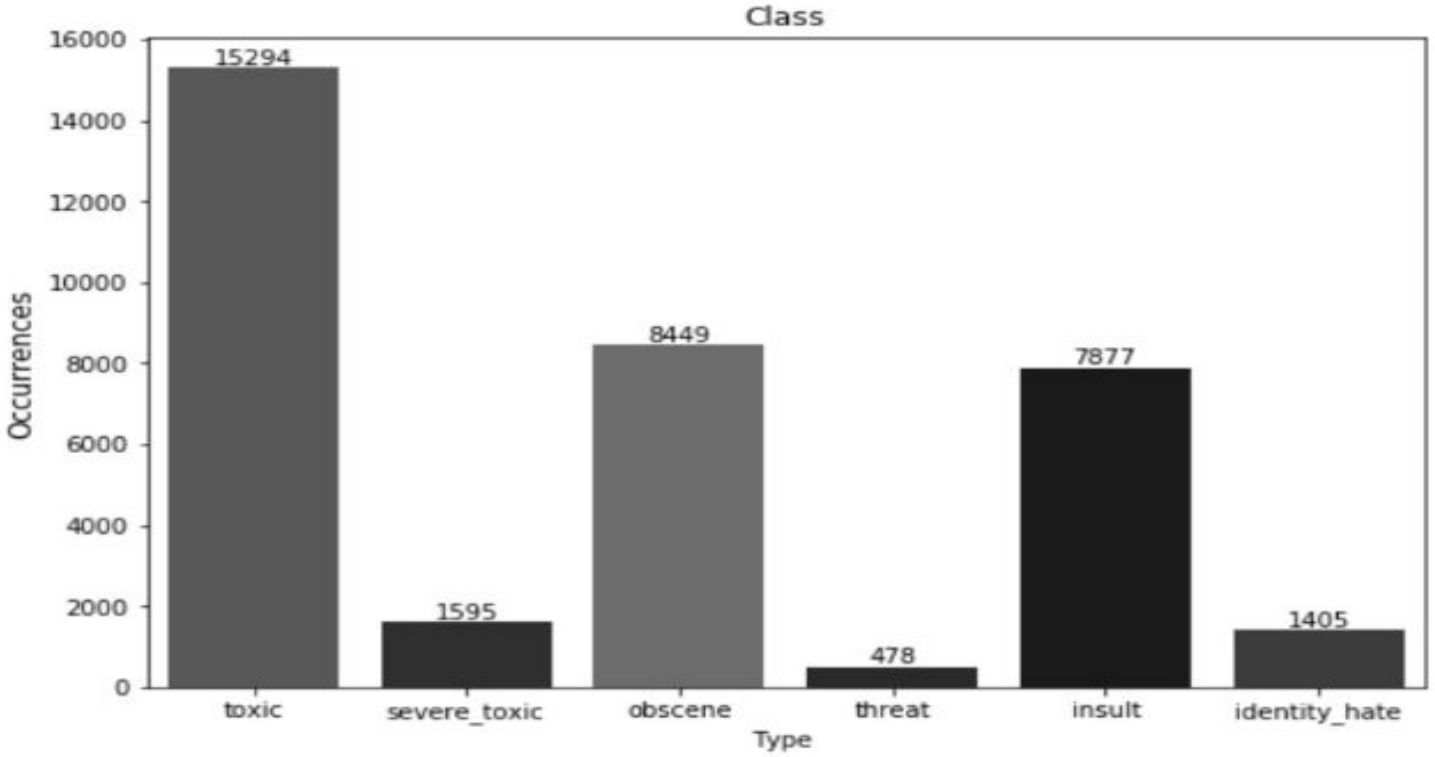
Related Work

- **Sentiment Analysis for online comments-** One of the initial work done by Yin, Dawei for abusive comment research with support vector machines.

- **Conversation AI team of Alphabet** - allow binary classification only (does not allow users to know which types of toxicity they can be categorized into).

Dataset

- Contains 159,571 instances and 6 classes



Preprocessing

- **Remove all the non-english comments** - removed 4290 comments (0.02%)
- **Convert our input to lowercase**
- **Remove all the irrelevant characters -** removed !"#$%&()*+,\t-./:;\n<=>?@[\]^_`{|}~
- **Padding**
- **Tokenizer**

Word Embeddings

- One hot encoding, a very sparse representation can't identify similarities between semantically close words.
- Word embeddings are just vector representation of words in a small dimensional space.
- Efficient and expressive than one hot encoding.
- Good pre-trained embeddings already available like GloVe and FastText are useful rather than manual web scraping.

Models

- **Convolutional Neural Networks**:- Convolution is one of the main building blocks of a CNN.

- Convnets have two properties:
  1. The patterns they learn are translation invariant
  2. They can learn spatial hierarchies of patterns

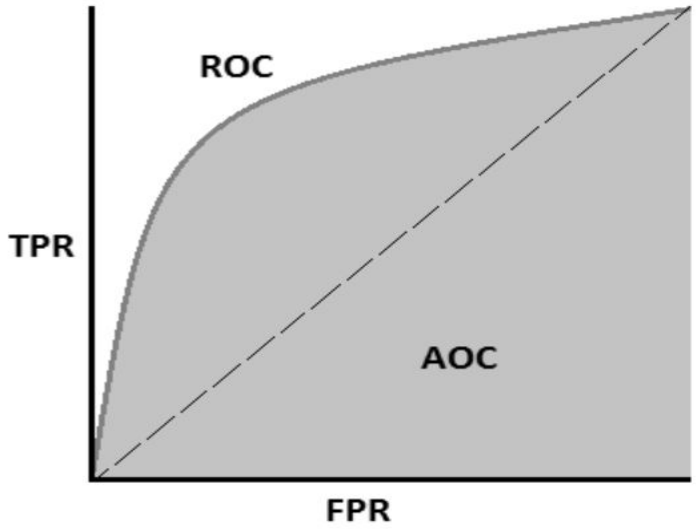
- CNNs has two major parts:
  1. The Feature extraction part
  2. The Classification part

Models(continued)

- **Recurrent Neural Network** : iterates through sequence and maintains previous state information.
- Problem- vanishing gradient problem
- **Long Short-Term Memory**(LSTM) is an algorithm implemented to counteract this
- Adds a way to carry information across many timesteps and helps in saving information for later.

Evaluation Metrics

- Best mean column-wise **ROC Area under the curve**.
- The more the value of AUC, the better the model is at predicting.



Result

ENGINEERING
TEXAS A&M UNIVERSITY

- FastText embeddings perform better than GloVe embeddings when it comes to handling toxic data.

- Convolutional neural network perform better than LSTM model for the same word embeddings.

Model	Word Embeddings	Area under the curve
LSTM	GloVe	0.9575
LSTM	FastText	0.9701
CNN	GloVe	0.9624
CNN	FastText	0.9751

Conclusion and Future Work

ENGINEERING
TEXAS A&M UNIVERSITY

- Best performance using CNN along with FastText word embeddings.
- Fails to understand the context of a comment. Example: “I am a gay black man” has 87 percent toxicity.
- Feedback channel can be provided to correct such behavior.
- An important research topic for future work is investigating improved semantic embedding.

References

[1]M. Duggan, Online harassment, 2014. http://www.pewinternet.org/2014/10/22/online-harassment/.

[2]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challeng

[3] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jgou, T. Mikolov, FastText.zip: Compressing text classification models

[4] Jerey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[5] Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of
machine learning algorithms. Pattern Recognition.

[6] Yin, Dawei, et al. Detection of harassment on web 2.0. Proceedings of the Content
Analysis in the WEB 2 (2009): 1-7.

[7] Perspective, available at https://www.perspectiveapi.com/

ENGINEERING

TEXAS A&M UNIVERSITY

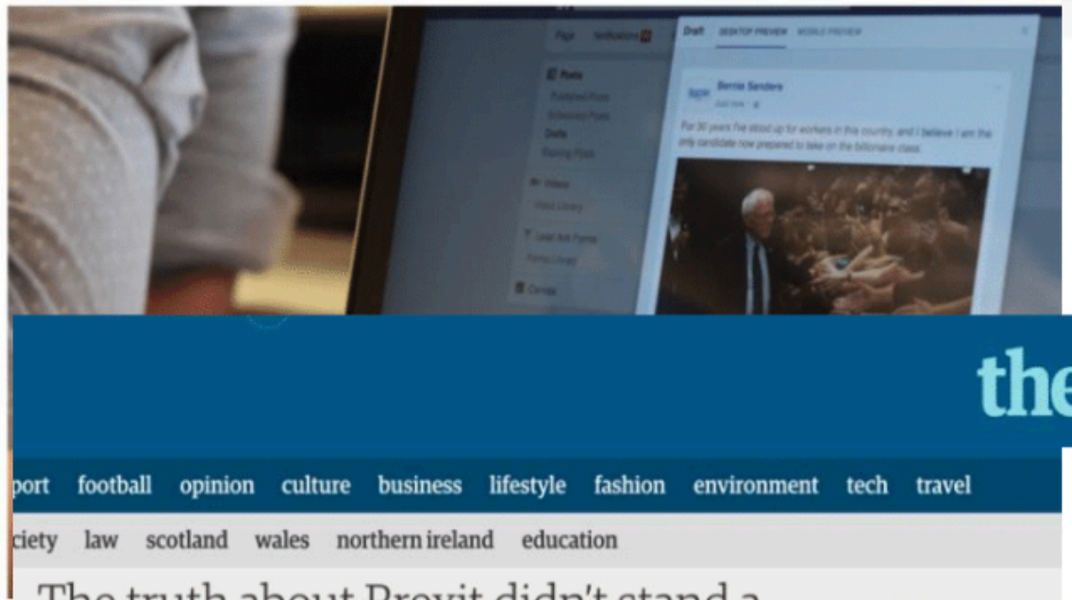
THANK YOU!

ARE YOU IN AN ECHO CHAMBER?

JANVI, ANDREW, NIKITA

WHAT ARE ECHO CHAMBERS, DO THEY REALLY EXIST, AND HOW DO THEY IMPACT SOCIETY?

The biggest threat to democracy? Your social media feed



WIRED

Opinion: Social Media Leads Us to Become Victims of Our Own Biases

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE

MOSTAFA M. EL-BERMAWY BUSINESS 11.18.16 5:45 AM

YOUR FILTER BUBBLE IS DESTROYING DEMOCRACY

theguardian

website of the year

port football opinion culture business lifestyle fashion environment tech travel

ciety law scotland wales northern ireland education

The truth about Brexit didn't stand a chance in the online bubble

Emily Bell

SCIENCE OF US

Search

The 'Filter Bubble' Explains Why Trump Won and You Didn't See It Coming

By **Drake Baer**

November 9, 2016
1:04 p.m.

Share

Tweet

OVERVIEW

- Goal - Identify if a social media user is in an echo chamber?
- Approach - socio and political topics, controversial views
- Data - Twitter datasets
- Classification models - Benchmarks and CNN
- Social network graph - can we find an echo-chamber?

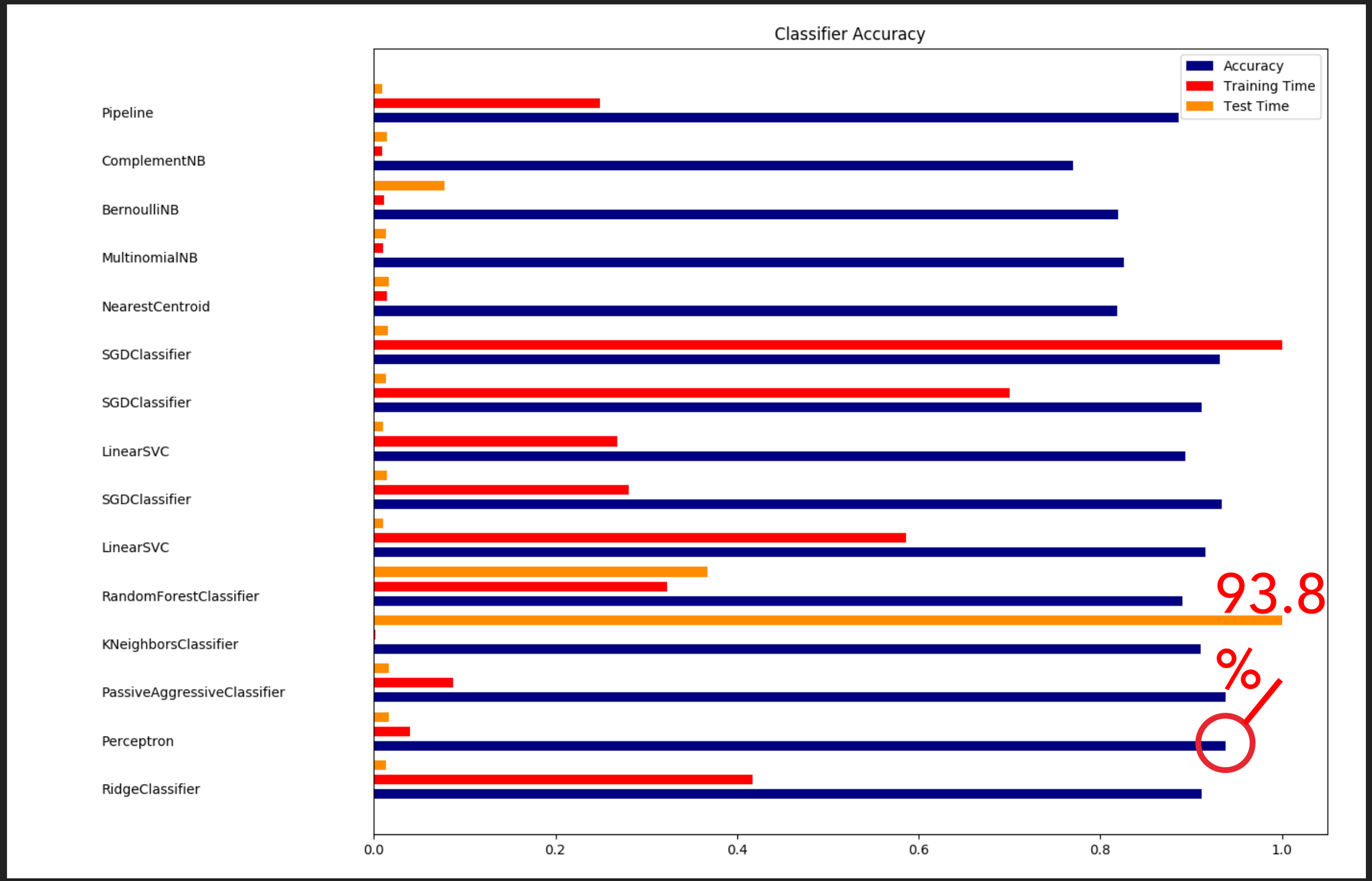
DESCRIPTION

- Twitter dataset on Obamacare and Abortion
- Contains user Twitter IDs with ground truth polarities
- Polarity in rage from -2.5 to 2.5
- Placed into 1 of 5 classes: extreme-left, moderate-left, neutral, moderate-right, and extreme-right
- Obamacare: ~8600 users in, Abortion: ~4000 users
- ~5000 Tweets per user

CLASSIFICATION OF USERS BASED ON TWEETS

BENCHMARKING TESTS (NO CROSS-VALIDATION)

- Accuracy of around 89% when tested with cross-validation
- SVM with Linear kernel, perceptron perform well



WORD EMBEDDINGS

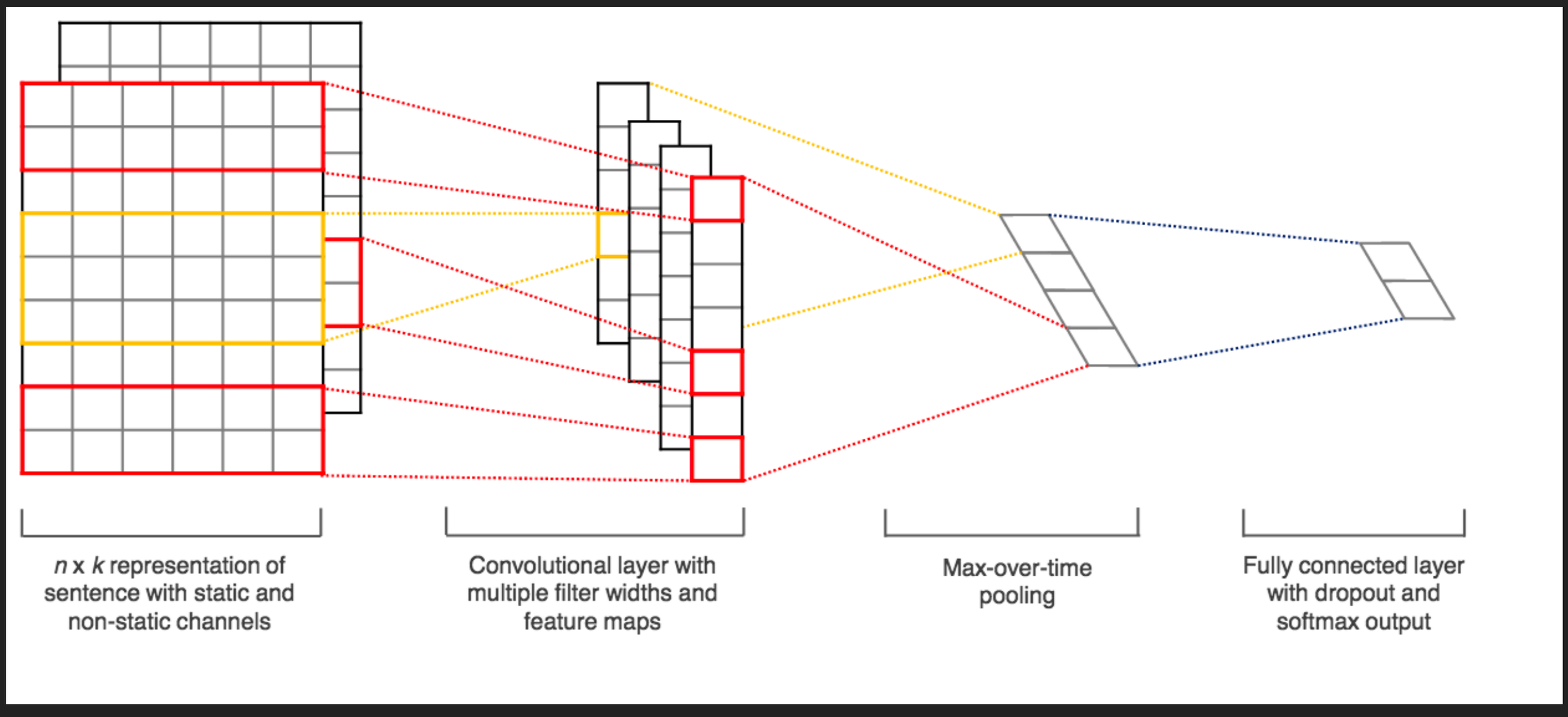
- gloVe
- word2Vec (egs)
- Trump : [('obama', 0.646), ('trumps', 0.609), ('dt', 0.6045), ('djt', 0.5994), ('drumpf', 0.5772),('hillary', 0.5479), ('romney', 0.522)]
- Stinky: [('smelly', 0.658), ('greasy', 0.603), ('wet', 0.544), ('wrinkled', 0.530), ('ewwwww', 0.5285), ('hairy', 0.514), ('salty', 0.512),('poop', 0.5030)]
- Great: stellar, terrific, wonderful…you get the gist.

USING LINEAR-SVM WITH WORD EMBEDDINGS

- MeanEmbeddingVectorizer, TfidfEmbeddingVectorizer
- Using both gloVe and word2vec
- Accuracy ~ 80.5%

CNN MODEL

- 3 layer model with ~92% accuracy using 2500 data points and gloVe embeddings



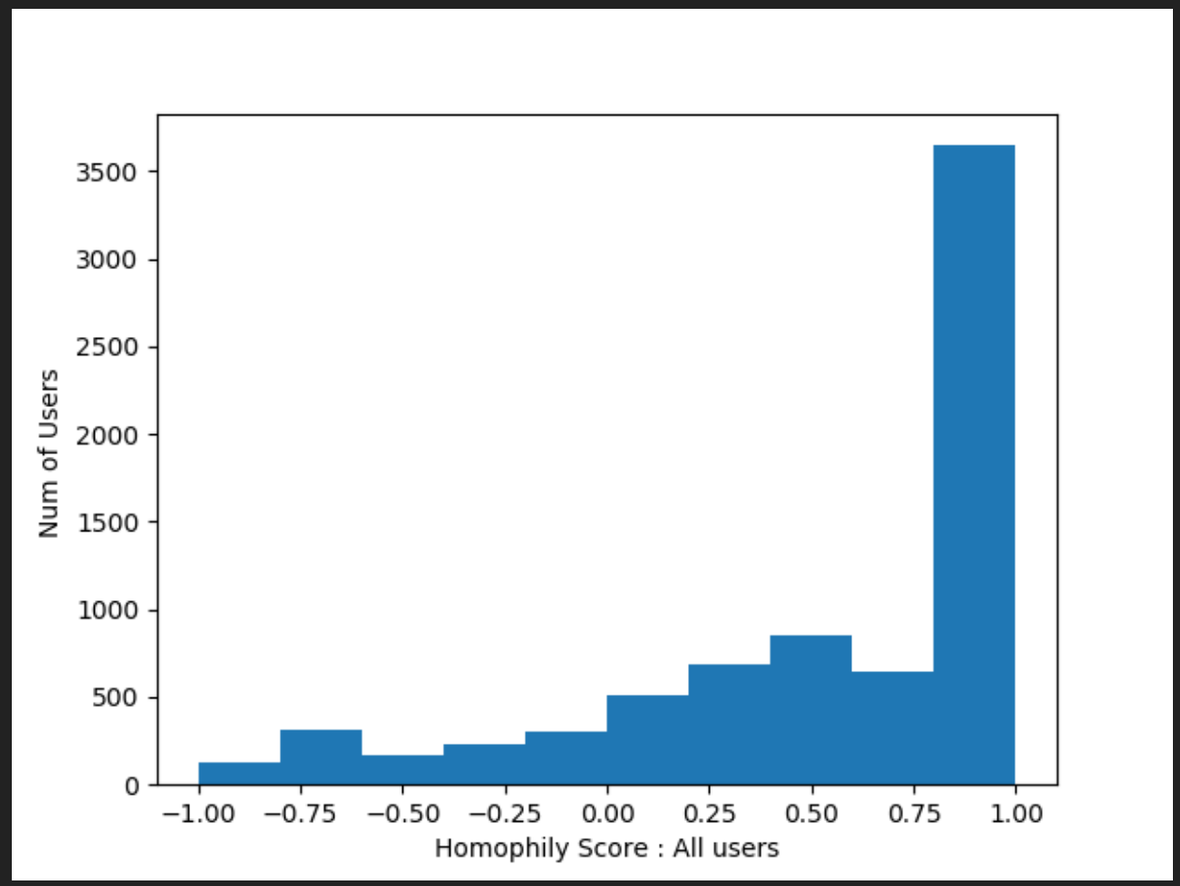
USER-NETWORK GRAPH

- “Friends” of a user
- Connected graph
- Homophily score
- Categories of users
  - Partisans
  - Bipartisans
  - Gatekeepers (guarding the echo chambers?)

GRAPH ANALYSIS

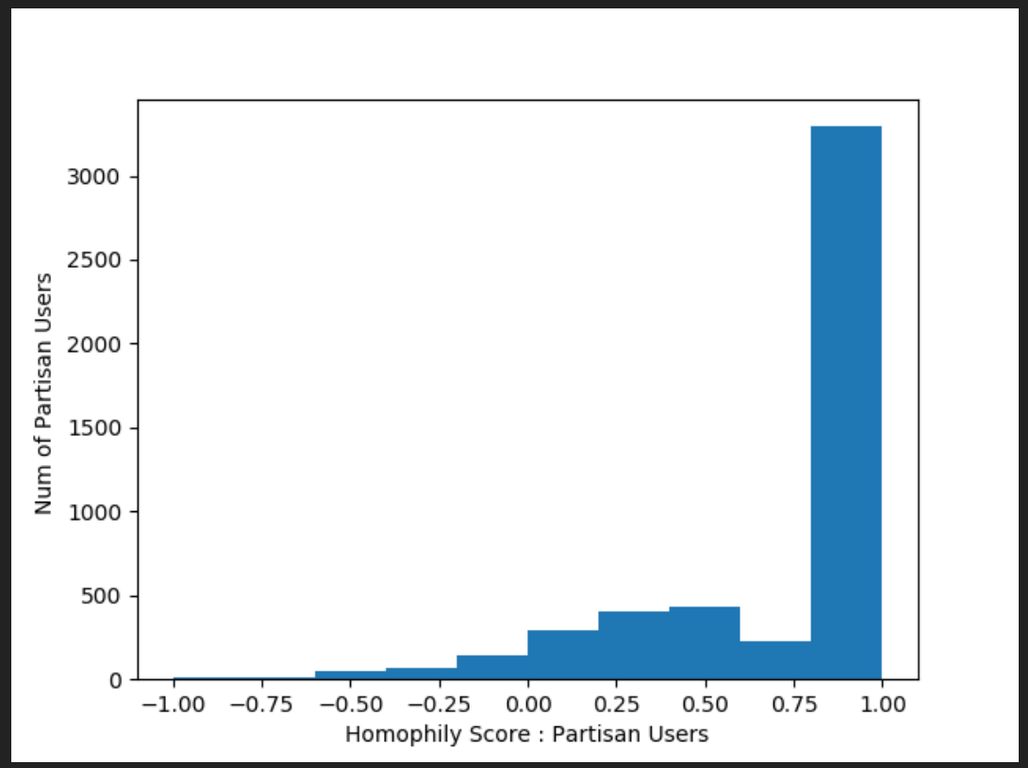
HOMOPHILY

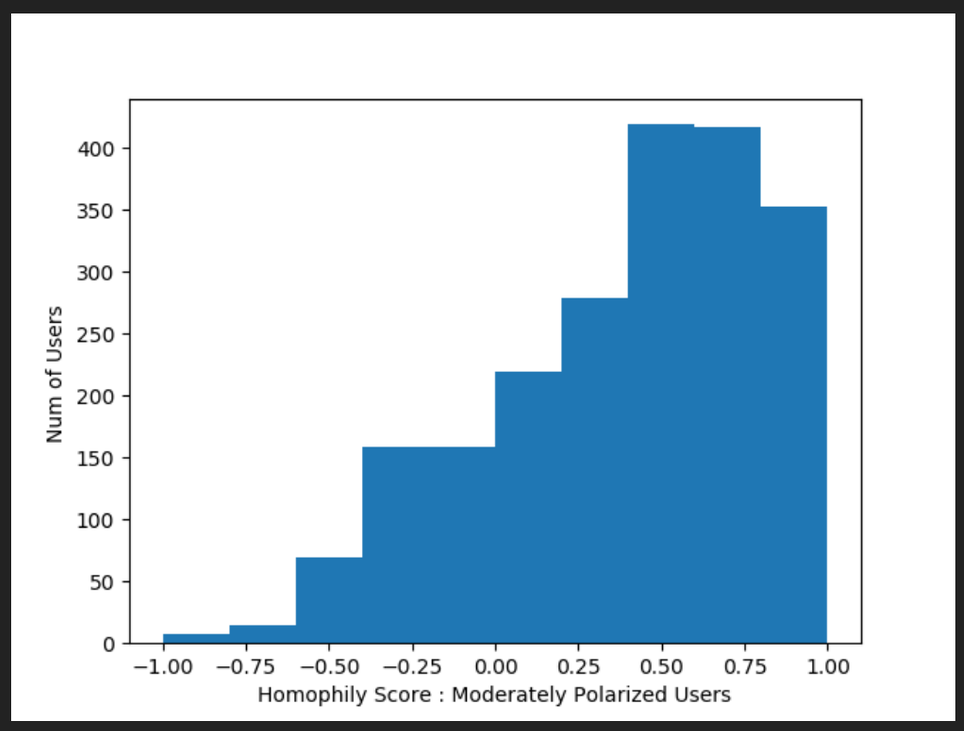
- Similarity breeds connection
- Quantified as the difference in the of proportion of friends with similar bias to friends of contrasting bias on a topic.
- Majority users tend to have more users with similar bias views.

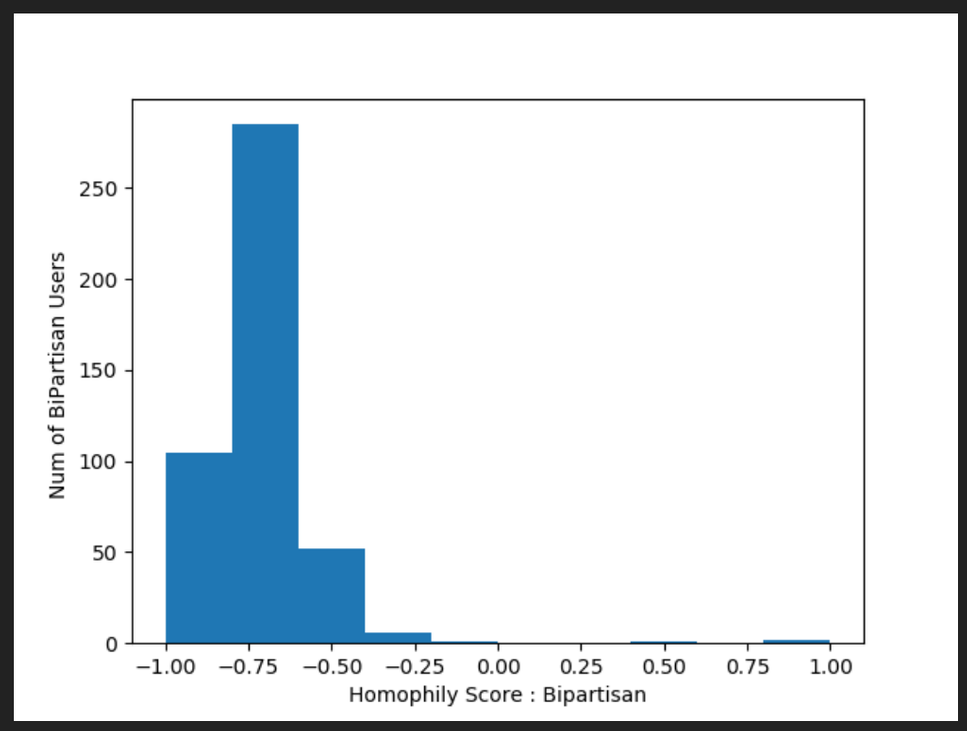


GRAPH ANALYSIS

HOMOPHILY : DIFFERENT CLASSES OF USERS

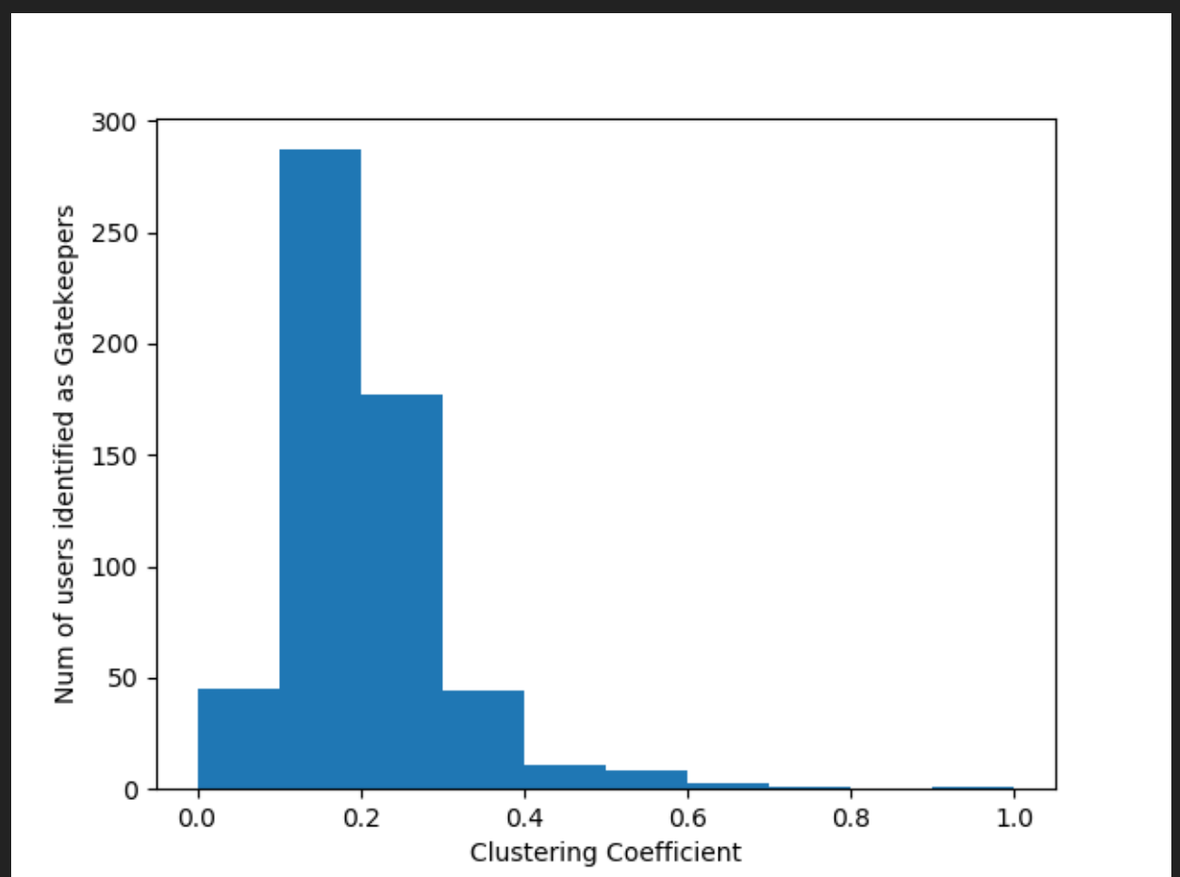


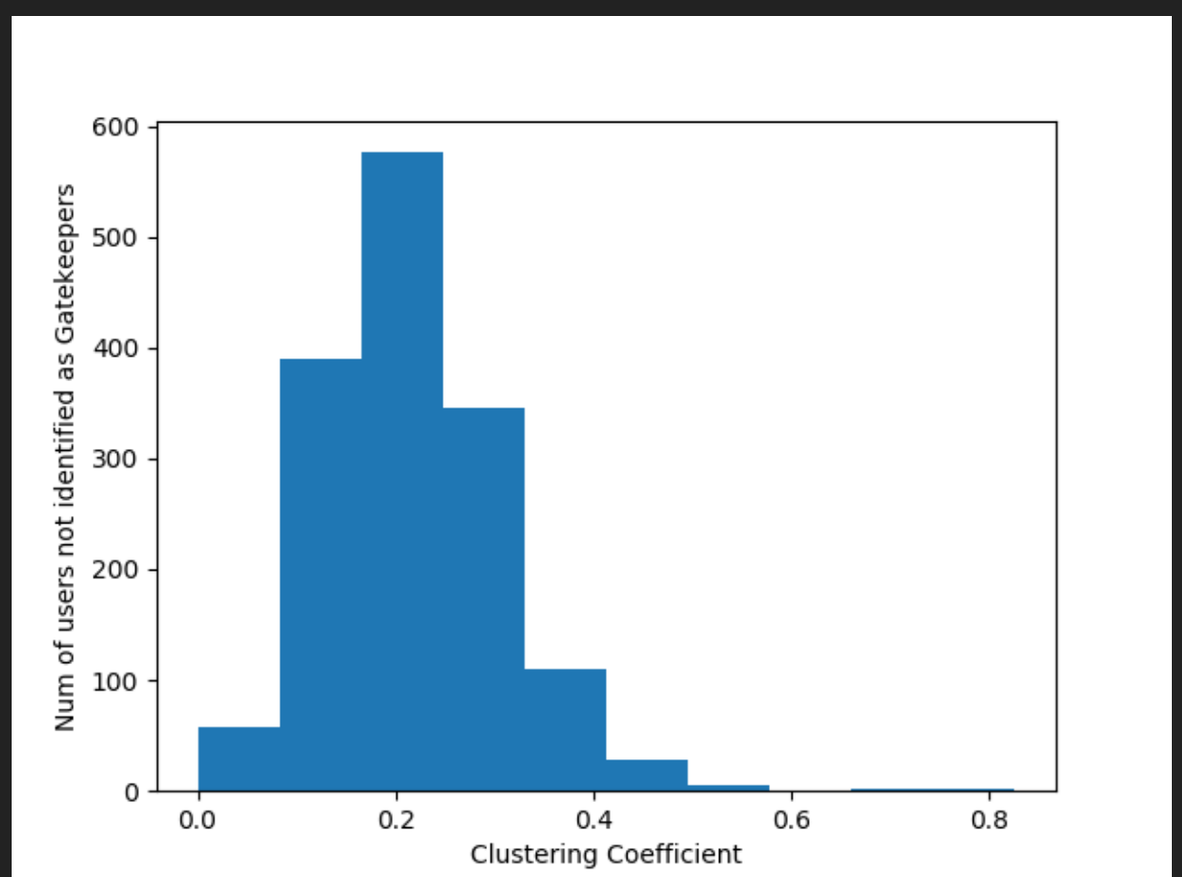




GRAPH ANALYSIS

GATEKEEPERS





- Border Spanners in the network
- Lower average clustering coefficient as compared to non gatekeepers

DO ECHO CHAMBERS EXIST? YES!

- A high percentage of users are only connected similarly biased users in the user network
- ~4,200 partisan users out of 8,600 are in a strong echo chamber! (67% of all partisan users)
- Gatekeepers - clustering coefficient

CONCLUSIONS

IMPACT

- Selective exposure to information
- Selective bias in consumption of information
- Worrisome outcomes
- Similar on other platforms, enabled by the platform (for eg. Facebook)
  - why show dissonant information?

WHAT AN ECHO CHAMBER LOOKS LIKE:

'Against' Echo Chamber: Obamacare

Partisan User: In 47 of 50 cities, ObamaCare coverage will be 'unaffordable' in 2018 by law's definition #ObamaLegacy

Friends:

#Obamacare = millions of Americans without health insurance; premiums skyrocketing; i nsurers fleeing; new taxes and hik…

Illegals erroneously get 750 million in Obamacare subsidies, Gov't screws up again! No consequences!

welcome to the future of obamacare the american version of socialized medicine

New Data Show Obamacare Insures Less Than 20 Million, Most on Medicaid

'For' Echo Chamber: Obamacare

Partisan User: Repealing obamacare alone would leave 32 million more uninsured cbo make those who would do this pay in 2018

Friends:

trump and his gop cohorts cannot come up with a health care bill that will pass so they will kill obamacare by withdrawing

one last attempt to repeal obamacare is gaining steam time to light up your senators phone lines urge them to vote no

so the idea is to buy votes for trumpcare by promising senators that their states can keep obamacare irony meet your

They begged. They bribed. They threatened. They sabotaged. Yet Obamacare still stands.

QUESTIONS?

Amazon User and Product Profiling and Recommendation based on Reviews

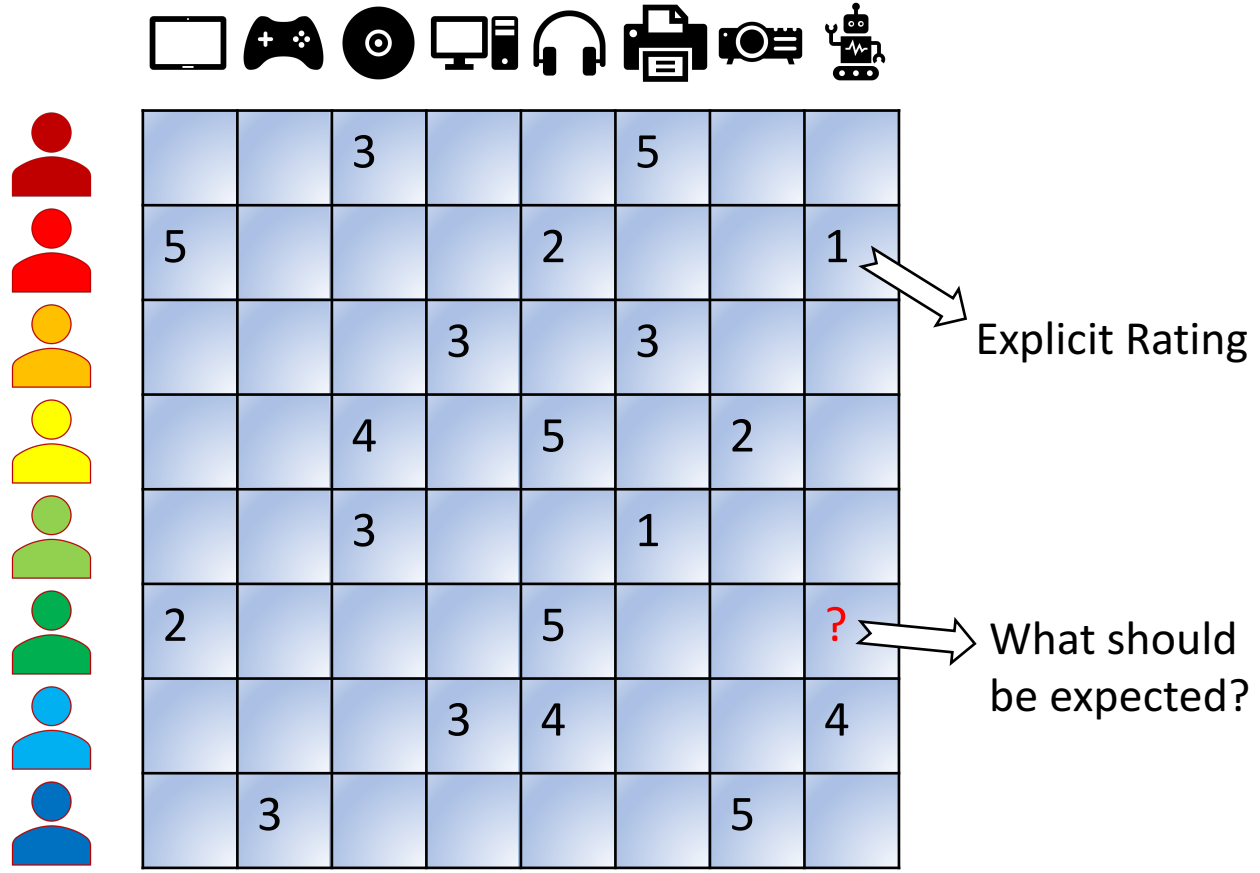
Ziwei Zhu, Xing Zhao, Yanze Li

Department of Computer Science and Engineering
Texas A&M University, USA

TEXAS A&M UNIVERSITY

Recommender System

- Recommender System is a subclass of information filtering **system** that seeks to predict the "rating" or "preference" a user would give to an item.
- Recommender System could employ:
  - Explicit info: e.g. ratings; or
  - Implicit info: e.g. purchase history, search history; or
  - Auxiliary info: e.g. review text.
- Evaluation: RMSE, @K, etc.



Previous Methods

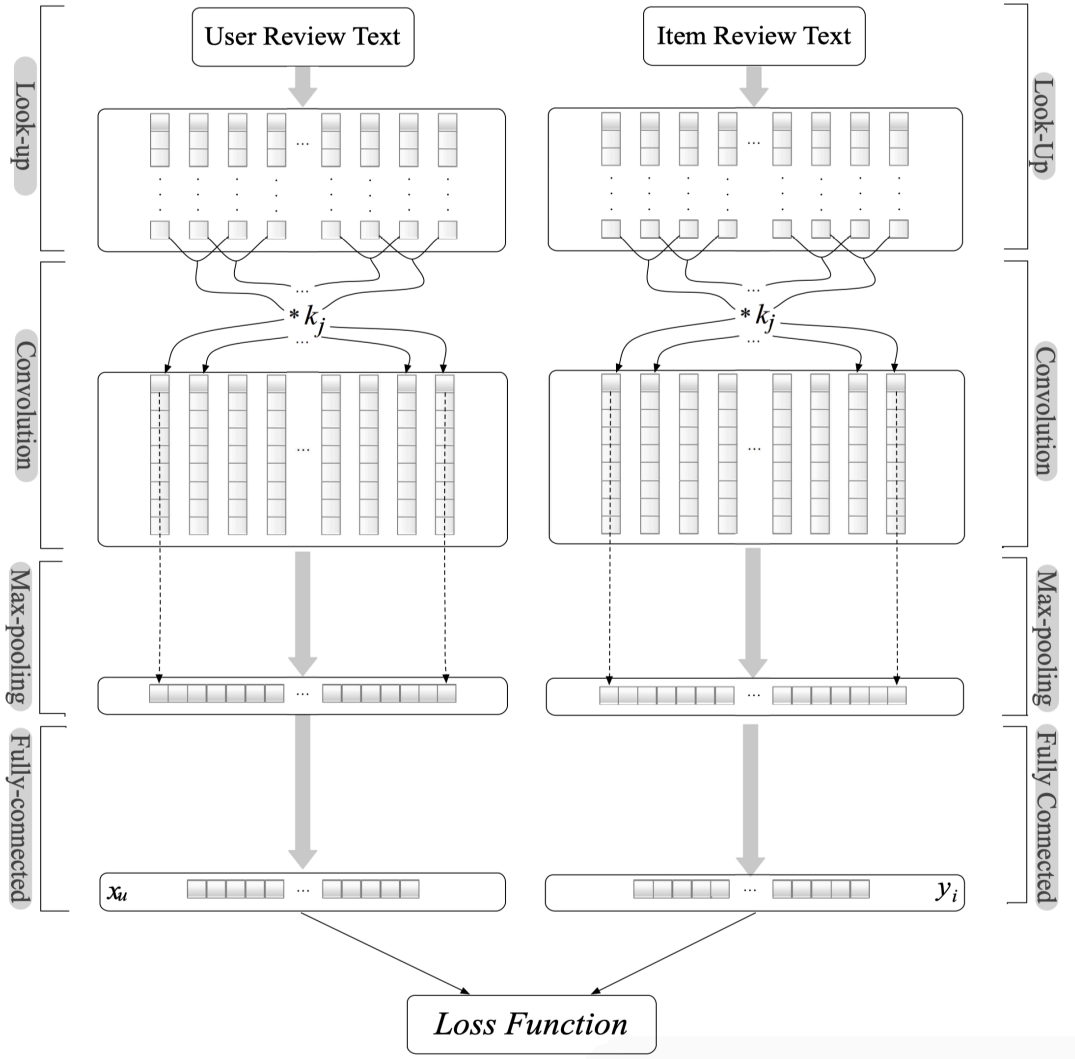
- Matrix Factorization
  - Performs well in many state-of-the-art recommender system.
  - Captures latent factors for users and items.
  - Purely based on numeric ratings, poor accuracy due to sparsity.
- HFT
  - Utilizes textual review data to enhance the recommendation quality using latent topic modeling.
  - Using bag-of-word method which ignore the words order information.
- DeepConn
  - Fully utilizes the textual reviews to profile items and users.
  - Expensive training cost (time and space).

DeepConn- Architecture

Word Embedding

User/Item Profiling

Shared Layer



Our Method: DeepCoNN+

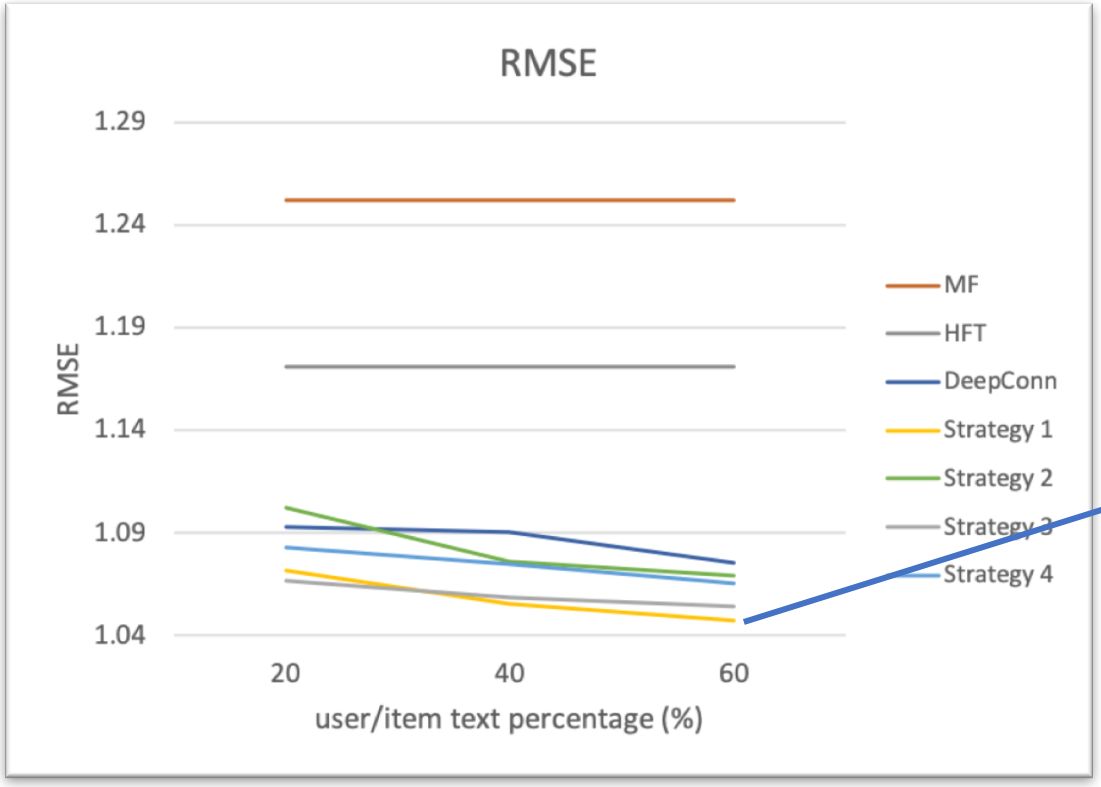
- Review Text Filtering using POS Tagging
  - RQ1. Which kind of words plays most important role?
    - Strategy 1: ['JJ', 'JJR', 'JJS’]
    - Strategy 2: ['JJ', 'JJR', 'JJS', 'NN', 'NNS', 'NNP', 'NNPS’]
    - Strategy 3: ['JJ', 'JJR', 'JJS', 'VBD', 'VBP', 'VBN', 'VBG’]
    - Strategy 4: ['JJ', 'JJR', 'JJS', 'NN', 'NNS', 'NNP', 'NNPS','VBD', 'VBP', 'VBN', 'VBG’]
  - RQ2. How good is the our representation?
  - RQ3. How fast is the training processes of our method?

Experiment

- Dataset
  - Amazon Video Game Reviews (rating 1-5)
    - Explicit ratings 1 - 5
    - 4212 users, 2928 products, and 71184
    - the density is around 0.58%
- Evaluation: Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)}{T}}$$

RQ1. Which words plays most important role?

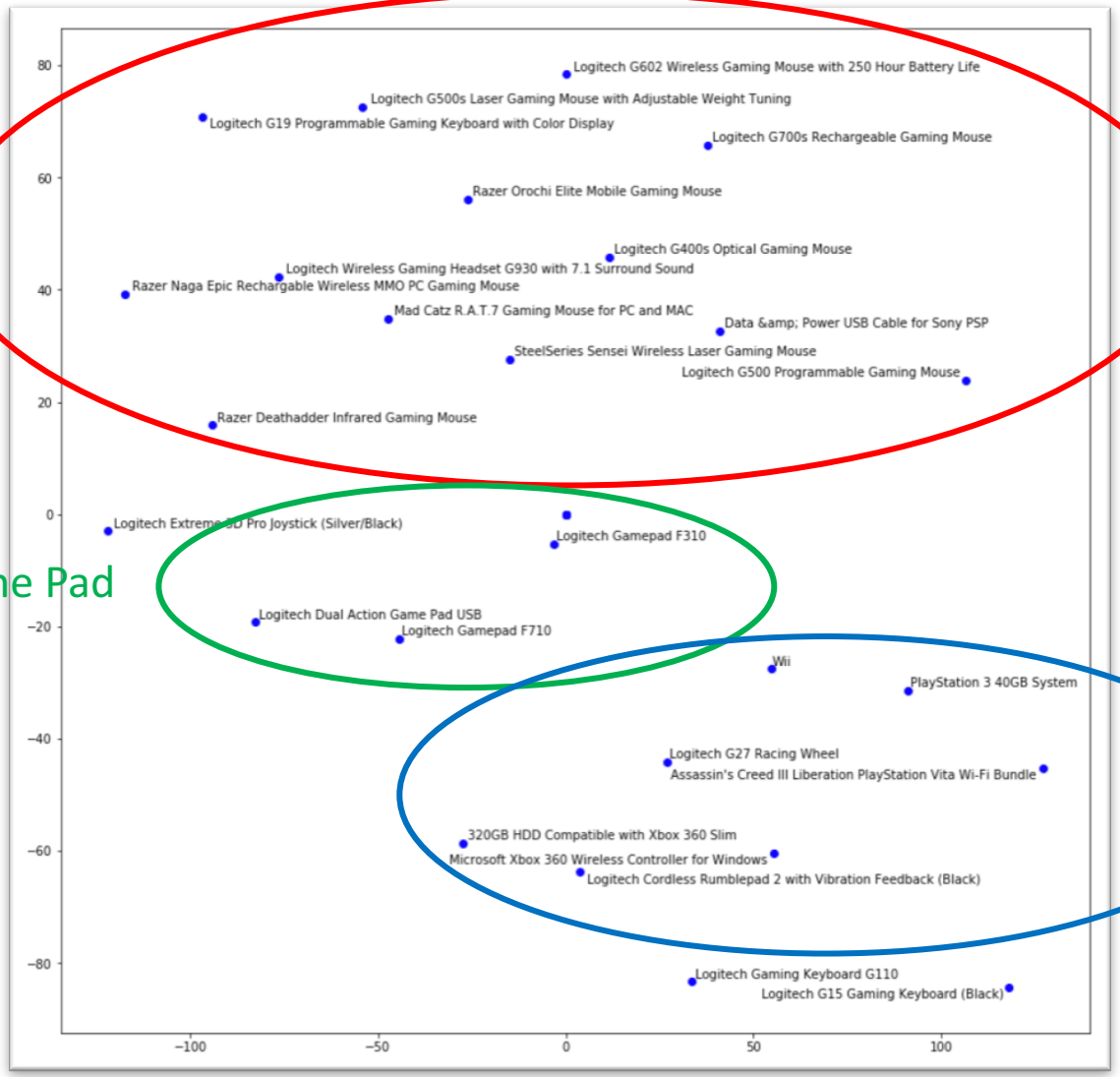


DeepCoNN+ (Strategy 1):

- 16.34% increase than MF
- 10.55% increase than HFT
- 2.59% increase than DeepCoNN

RQ2. How good is the our representation?

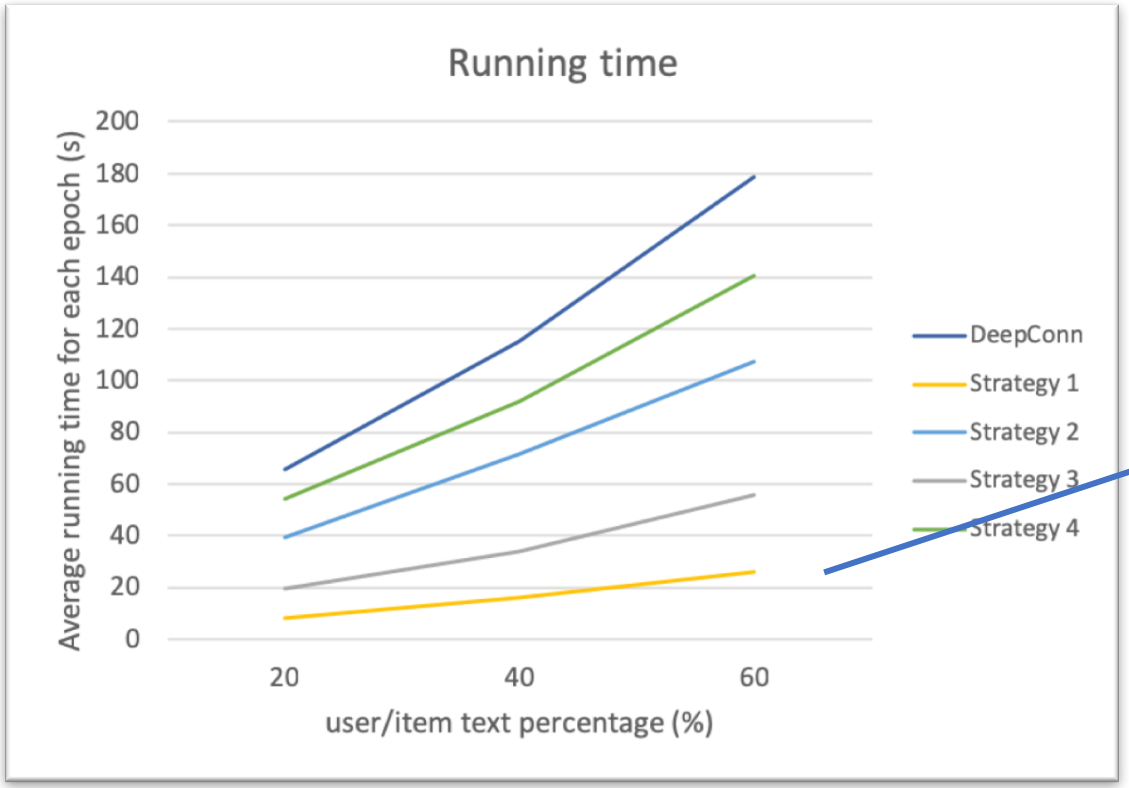
Gamming Mouse and Keyboard



Game Pad

Game consoles

RQ3. How fast is the training processes of our method?



DeepCoNN+ (Strategy 1):

- 5.82X speedup over DeepCoNN (|text| = 60%)
- 6.26X speedup over DeepCoNN (|text| = 40%)
- 7.09X speedup over DeepCoNN (|text| = 20%)

Conclusion

- Textual reviews can help us better profiling the preference of users and the properties of items
- The adjectives in textual reviews play the most important roles in the learning process
- Using POS Tagging to filter out redundant words not only improves the accuracy but also speedup the model performance significantly.

Thanks

Detecting Offensive Language using attention based sentiment flow

Amulya Agarwal

Introduction

- Social media conversations
- Sentiment analysis, Cyber-bullying and aggression detection
- Automated system for detection

Dataset

- Stack Overflow dataset

	Training	Validation	Testing
Offensive	450000	50000	70000
No Flag	50000	4961	7127
Total	500000	54961	77127

Please edit your code. It's a mess now. vs Thanks for catching that

Problem Statement

- Text Classification task
- Not all words are important
- Find the words and sentences which are important

Please edit your code. It's a mess now. vs Thanks for catching that

Related Work

- Hierarchical Attention Network (Yang et al)

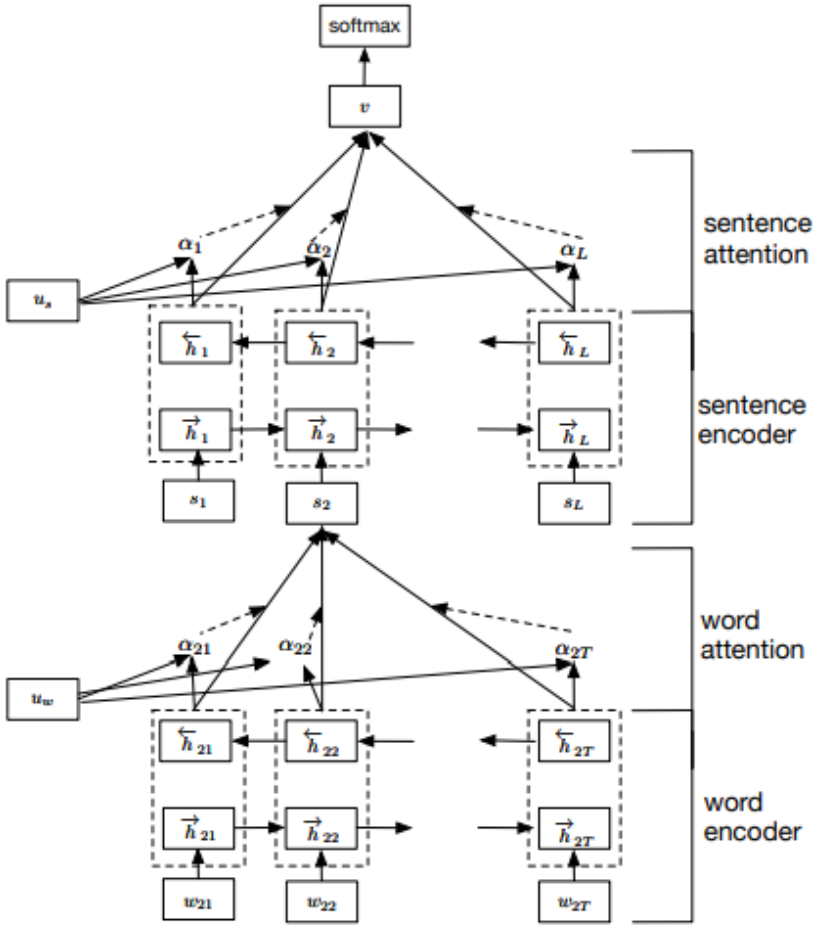
$$h_t^f = LSTM(x_t, h_{t-1}^f)$$

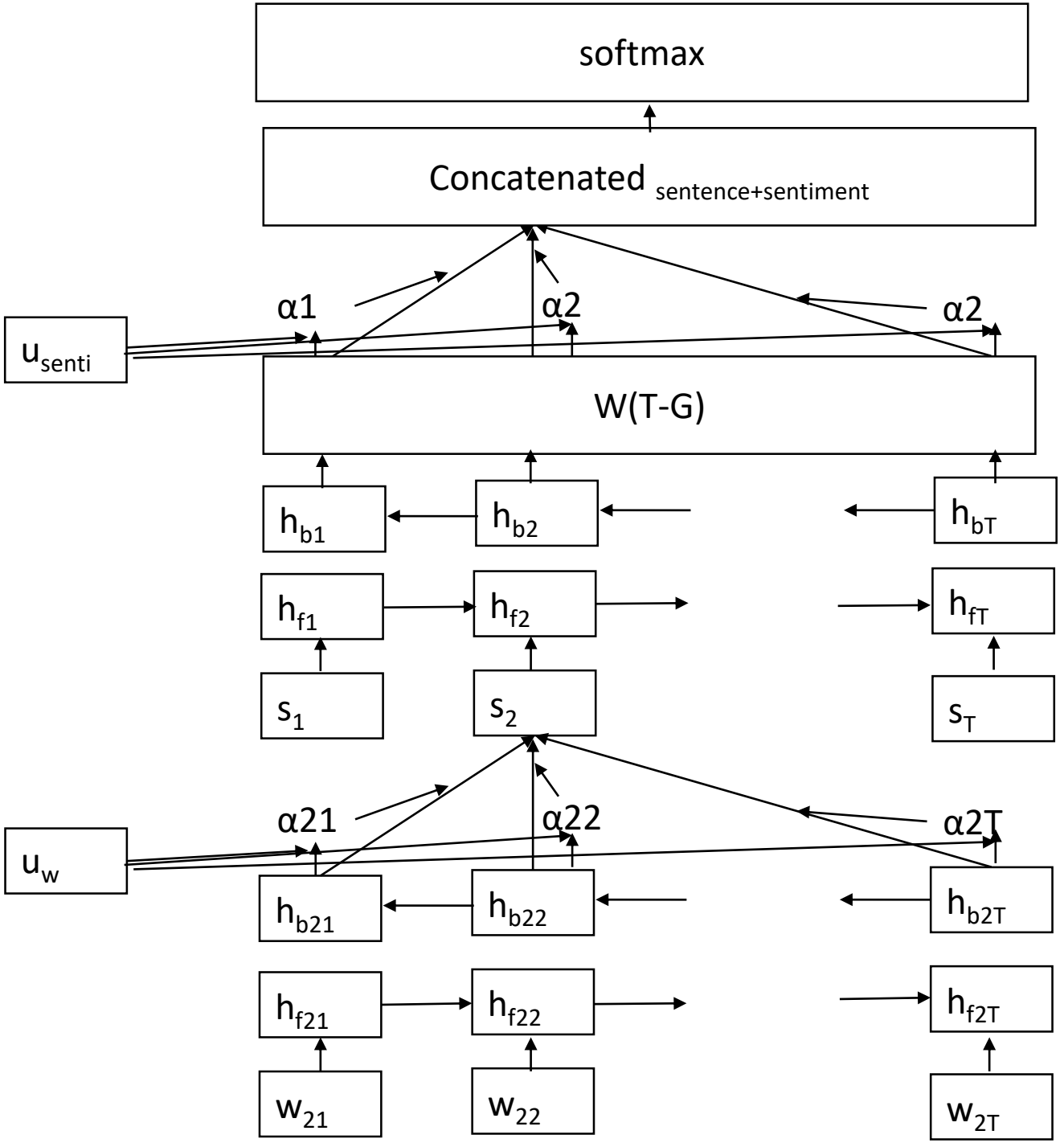
$$h_t^b = LSTM(x_t, h_{t-1}^b)$$

$$u_{it} = \tanh(W_w h_{it} + b_w)$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum \exp(u_{it}^T u_w)}$$

$$s_i = \sum \alpha_{it} h_{it}$$




softmax
Concatenated sentence+sentiment
α1
α2
α2
u senti
W(T-G)
h b1
h b2
h bT
h f1
h f2
h fT
s 1
s 2
s T
α21
α22
α2T
u w
h b21
h b22
h b2T
h f21
h f22
h f2T
w 21
w 22
w 2T

Results

	Precision	Recall	F-Score
Bi-LSTM + MaxPooling	0.76	0.42	0.54
Bi-LSTM + Bi-LSTM	0.78	0.43	0.56
HAN	0.77	0.45	0.57
Proposed Model	0.72	0.51	0.59

Future Work

- POS tag based attention
- Dependency Parsing based attention
- Co-attention

Fake News Detection on Social Media

CSCE 638 Project

Presented by :
Buvaneish Sundar UIN : 927009383
Abhishek Gajuri UIN : 327005766
Rahul Baghel UIN : 627007808

Content..

Overview

Problem Statement

Data Set

Preprocessing And Feature Extraction

Algorithms

Results

Conclusion

Overview

- Social Media as a news platform has gained popularity
  - Low cost
  - Easy Access
  - Rapid Dissemination

- This increased the spread of '**fake news**' - low-quality news with intentionally false information.

- We have implemented a **style** based and a **stance** based approach

Problem Statement

- Two separate classification problems.
- Style based detection - a news article, along with it’s headline is classified as fake(1) or not(0).
- Stance based detection - decision is taken based on the semantic relationship between the headline and the body
- Every (article, headline) is divided into 4 classes, that tells whether the headline agrees with (0), disagrees with (1), discusses (2) or is unrelated (3) to the article.

Data Set

Style based detection

Kaggle Dataset for Fake News Detection

- 20800 articles with title and author information
- Split into train and test sets in the ratio 0.8:0.2

Stance based detection

Publicly available Fake News Challenge (FNC 1)

- 44974 (headline, body) pairs for training
- 25413 pairs for testing

Preprocessing..

- Removing non-alphanumeric characters
- Stemming of words
- Removal of stop words
- Removing Punctuations

Feature Extraction : Style Based

Readability Values

- Readability metrics like Flesch reading ease, Gunning fog index and Dale Chall readability score are used.

Punctuation Values

- 32 punctuations from string.punctuation, normalized by document length

Psycholinguistic Features

- **Function words** : 176
- **Words per Sentence**
- **POS Tags**: For each document, we calculate the number of occurrences of each of the 38 POS tags
- **Sentiment Features**:
  - Positive sentiment score
  - Negative sentiment score
  - Overall polarity of the document

Cosine similarity between headline and body:

- Headline and body are encoded to TF-IDF vectors
- Converted to 50 sized vector, with GLOVE

N-Grams:

- Unigram and bigram features encoded in TF values
- Headline and Author : 2000 unigrams (1500 MF + 500 LF)
- Body : 1000 unigrams (MF) and 1000 bigrams (MF)
- Total N-Gram features considered : 4000.

Normalization : By standard deviation (X* = (X - mean)/std)

Feature Extraction:Stance Based

KL Divergence

- This gives us an insight as to how diverse the words used (language model) in the headline and the body are.

Cosine similarity between headline and body

N-Gram Overlap:

- Refers to the number of N-Grams that co-occur in the headline and the body.
- The number of 1 gram, 2 gram, and 3 gram overlaps are summed up.

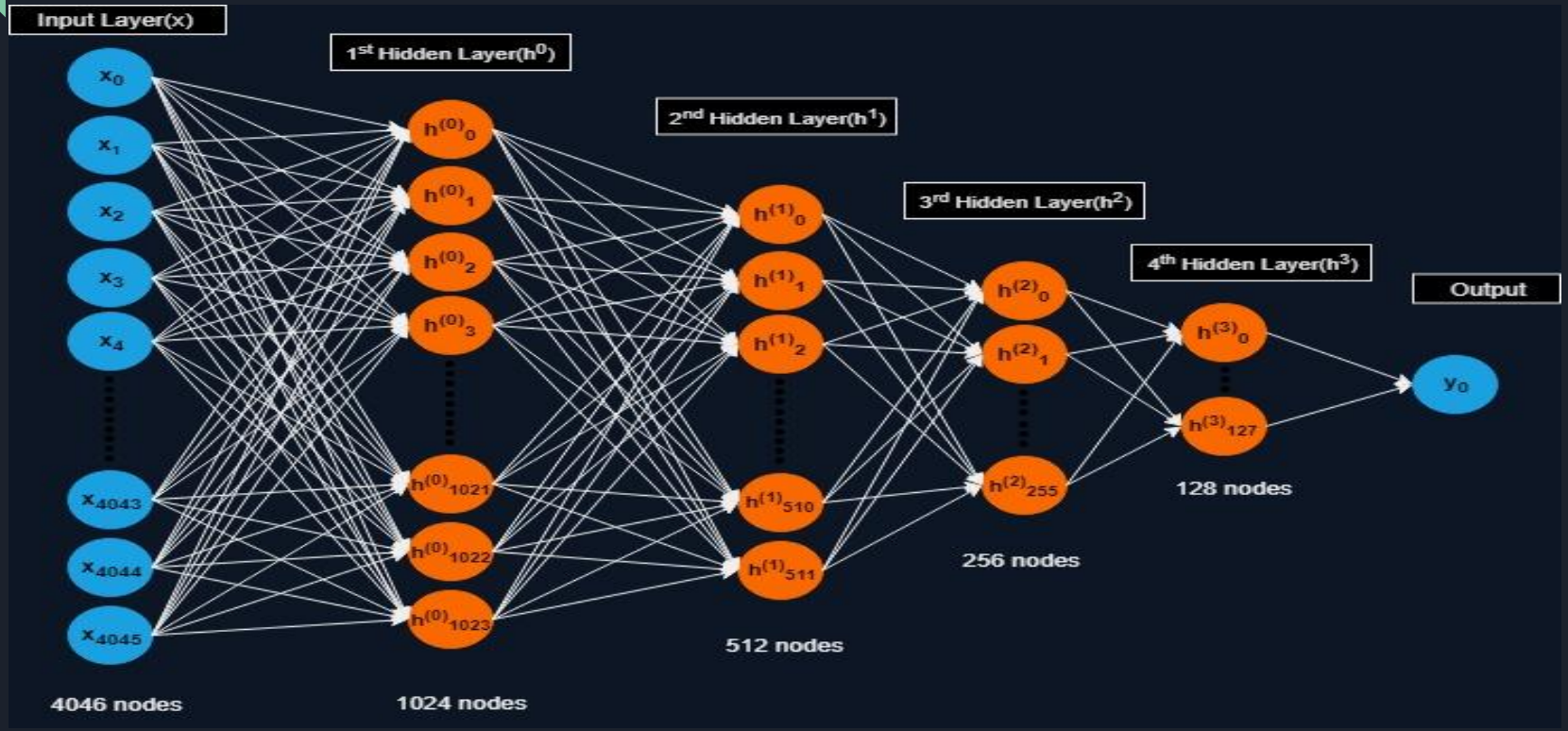
Algorithms

Neural Networks: All Fully Connected layers

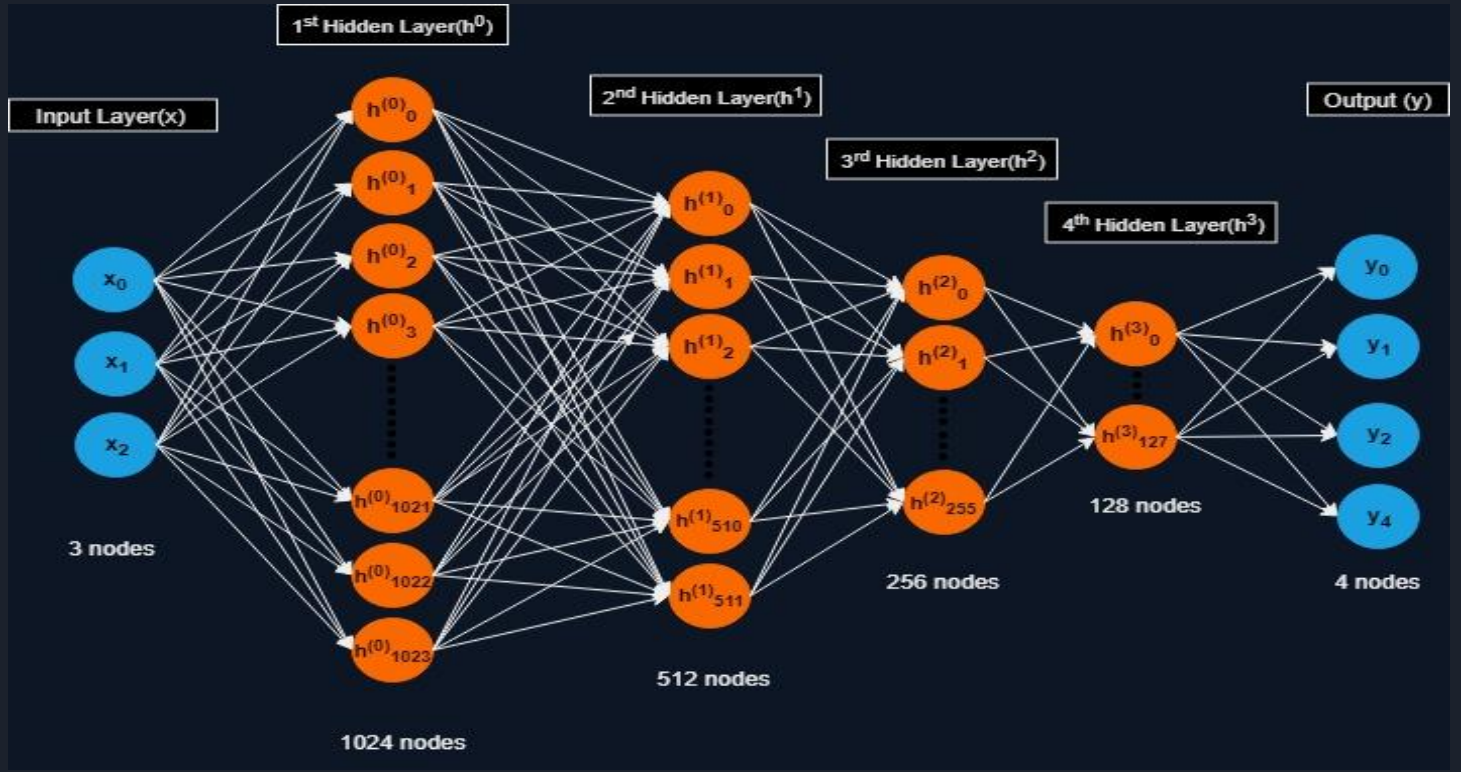
- Similar network architecture is used for both the style based and the stance based methods.

Parameter	Value
Number of Epochs	200
Tolerance Factor	0.0001
Activation	Relu for hidden layers and Softmax for output
Solver	‘lbfgs’
Learning Rate	0.001
Batch Size	200

Neural Network architecture for Style Based



Neural Network architecture for Stance Based



Algorithms

Logistic Regression: SKlearn's Logistic Regression classifier.

Parameter	Value
Number of Epochs	100
Tolerance Factor	0.0001
Solver	LibLinear
Multi_class	'ovr'(one vs the rest)
Penalty	'l2'

Result

Stance based method:

Algorithm	Accuracy
Neural Network	0.8637
Logistic Regression	0.8570

Style based method:

Algorithm	Accuracy
Neural Network	0.9552
Logistic Regression	0.9810

Conclusion and Future Work

- Logistic Regression performs better than the Neural Network architecture for style based detection.
- Alternatives: content-based approach - knowledge-based models.
- Social Context based detection - Propagation based methods.

Thank you!