CS 2731 / ISSP 2230
Introduction to Natural Language Processing

Session 28: Final project presentations

April 24, 2024

University of Pittsburgh | School of Computing and Information

Course logistics

- Final project reports due **tomorrow, Thu Apr 25, 11:59pm**
- *Thanks for a great semester!*

Instructions

- Plan for **5 min max** presentations + a brief Q&A
- Cover at least these key points
  - Project motivation (briefly)
  - Data
  - Methods, or annotation/collection approach for dataset projects
  - Results
- Put your slides in this presentation after your project name slide by class session, 3:00pm on Wed Apr 24

Schedule

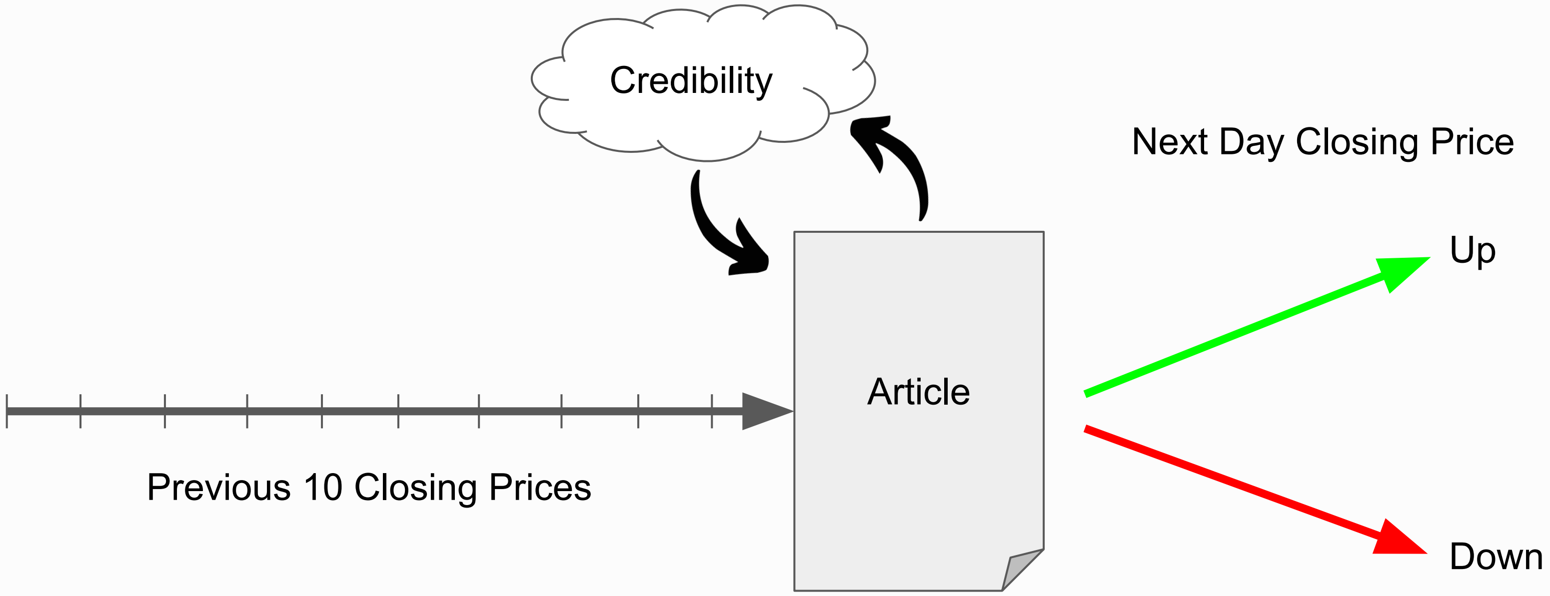
1. Kartik, Kasvitha, Brian, Owen
2. Purva, Fatemeh, Ayush, Shayan
3. Sai, Deyasini, Shiva, Aparna
4. Yuning, Na, Ken, Yuelong
5. Werner, Yuelyu, East, Anfeng
6. Hongtao, Chonghao, Sean, Bo-Chen
7. Shiyuan, Yingda
8. Nick, Arushi, Trung
9. Noah, Annanya, Jayden, Xiaoyan

1. Kartik, Kasvitha, Brian, Owen

Context & Motivation

- Stock prices are volatile and hard to predict
- Influenced by many external factors
    - Public opinion
    - Company restructuring
    - General market trends
    - Etc.
- Accurately predicting stock movement can yield high returns
- Use financial news articles

Our Approach



Datasets and Models

Credibility Classification

- Bert-based model
- Fake news classification and Fake news classification2 - Kaggle
  - Title, Text, Credibility Label
- Domain-based credibility scores
  - Lin, H., Lasser, J., Lewandowsky, S., Cole, R., Gully, A., Rand, D. G., & Pennycook, G. (2023). High level of correspondence across different news domain quality rating sets. PNAS Nexus, 2(9), 1-8.

Stock Price Prediction

- Bert-based model
- US financial news articles - Kaggle
  - Title, Text, Publisher, Date
  - January-May 2018
- Yahoo Finance API

Results

	Accuracy	Precision	Recall	F1
Baseline Model	0.8052	0.8319	0.8052	0.7962
Stock Only	0.5619	0.3157	0.5619	0.4043
Article Only	0.8044	0.8307	0.8044	0.7953
Model-Predicted Credibility	0.7959	0.8046	0.7959	0.7904
Domain-Based Credibility	0.8025	0.8296	0.8025	0.7931

Thank you!

2. Purva, Fatemeh, Ayush, Shayan

Detecting Humor-Infused Hate Speech in Online Content

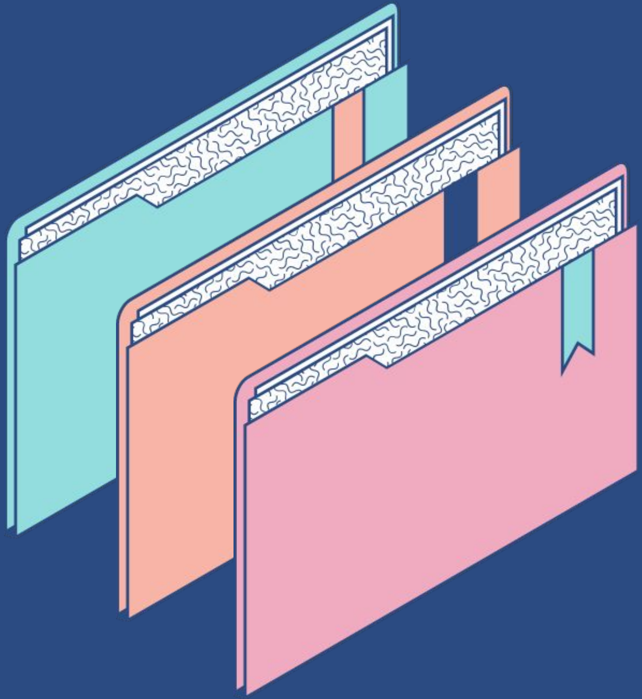
By -

Purva Chaudhari

Shayan Paigambari

Ayush Malik

Fatemeh Golshan

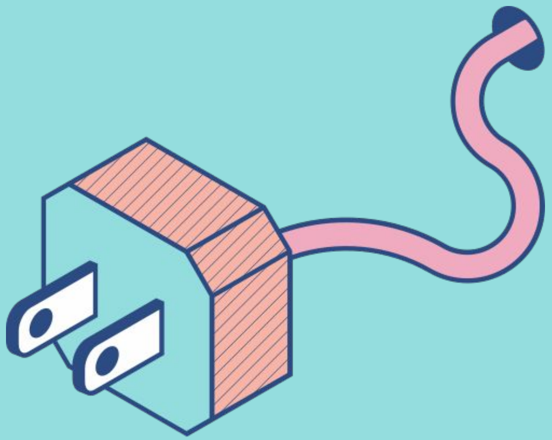


Challenge and Motivation

DETAILS AND OUTLINES OF PROJECT

- Challenge: Distinguishing hate speech disguised as humor on social media is crucial for maintaining online integrity.
- Goal: Develop a sophisticated dataset capturing the complexity of humorous hate speech.
- Purpose: Enable advanced NLP classifiers to accurately differentiate harmful content from innocuous humor.

- Methodology: Analyze existing datasets, collect and annotate new data, and utilize ML models like Logistic Regression, DistilBERT, and LSTM.
- Impact: Enhance content moderation tools' efficacy in identifying and mitigating hate speech across diverse platforms.



OUR APPROACH

- HaHackathon dataset:
- Offers insights into detecting and rating humor and offense.
- Uses binary and scalar annotations to indicate humor presence and offense level.
- SALT NLP Implicit Hate Speech dataset:
- Introduces nuanced classification with categories like not_hate, implicit_hate, and explicit_hate.
- Provides a detailed view of hate speech possibly intertwined with humor.
- Hate Speech and Offensive Language Dataset:
- Includes binary and categorical annotations for hate speech and offensive language.
- Facilitates critical binary distinction necessary for initial model training.

METHODLOGY AND FLOW OF PROJECT

1 — 2

Data Collection

- Curate diverse data from multiple online platforms where humor is intermixed with hate speech.
- Prioritize platforms with user-generated content, incorporating informal language, slang, and cultural references.
- Gather data in various formats like text posts, comments, and transcribed audio snippets to capture linguistic diversity.

Annotation Scheme Development (5.2)

- Create a robust annotation scheme informed by linguistic theory and practical online communication considerations.
- Define multiple categories of humor and hate speech, accounting for irony, sarcasm, and contextual nuances.
- Annotated data are labeled by linguists and domain experts using a detailed codebook to ensure consistency and accuracy.

ML , DEEP LEARNING AND MODEL REFINEMENT

- Classifier Development
  - Employ a blend of traditional ML algorithms and state-of-the-art deep learning models.
  - Initial experiments include logistic regression and support vector machines with TF-IDF features.
  - Explore advanced models like BERT and LSTM networks to capture deeper linguistic structures and context.

- Training and Evaluation
  - Train each classifier using the annotated dataset.
  - Evaluate classifier performance using metrics such as precision, recall, and F1 score.
  - Implement cross-validation to ensure model robustness and prevent overfitting.
  - Test the best-performing models on new, unseen data in a real-world setting.

HOW WE STRUCTURED OUR ALGORITHM DESIGN

Algorithm Design: Approach and Methodology

- Objective: The goal is to train a BERT-based sequence classification model to differentiate between different categories of text data, specifically focusing on hate speech detection.

- Dataset Preparation:
    - Data Splitting: The dataset is split into training, validation, and potentially testing sets using train_test_split from sklearn.model_selection.
    - TensorDataset Creation: Construct TensorDataset objects containing input IDs, masks, and labels for training and validation data.

HOW WE STRUCTURED OUR ALGORITHM DESIGN

- Model Selection:
  - BERT Initialization: Load a pre-trained BERT model ('bert-base-uncased') using BertForSequenceClassification.
  - Number of Labels: Configure the model for a specific number of output labels (in this case, 3).
- Training Configuration:
  - Optimizer: Utilize AdamW optimizer from transformers with a learning rate (lr) of 2e-5.
  - Scheduler: Implement a linear scheduler with warm-up steps (num_warmup_steps) and total training steps (num_training_steps) for learning rate adjustment during training.
- Training Loop:
  - Batch Processing: Iterate through batches of data using DataLoader with random sampling (RandomSampler).
  - GPU Acceleration: Move input data and model to GPU ('cuda') for faster computation.
  - Forward Pass: Execute a forward pass of the model on input batches to obtain predictions and calculate the loss.

HOW WE STRUCTURED OUR ALGORITHM DESIGN

- Backpropagation: Backpropagate the loss to update model parameters using gradient descent (optimizer.step()).
- Gradient Clipping: Prevent exploding gradients by clipping gradients to a maximum norm (torch.nn.utils.clip_grad_norm_).
- Learning Rate Scheduling: Adjust learning rate during training using the scheduler (scheduler.step()).
- Training Evaluation:
- Loss Calculation: Compute average training loss over all batches within each epoch.
- Performance Monitoring: Track model training progress by printing epoch-wise training loss.
- Hardware Utilization:
- GPU Acceleration: Ensure efficient utilization of GPU resources for faster model training.

COMPREHENSIVE RESULTS AND MODEL OUTCOMES

```
0.7568931338979996 %
1.4957650027031897 %
2.97350874031357 %
4.559380068480808 %
6.001081275905569 %
7.460803748423139 %
8.830419895476663 %
10.380248693458281 %
12.074247612182374 %
13.389799963957469 %
hateBERT model detecting hate Humor: 0.424090338770389
```

Fine-tuned hateBERT model detecting straight hate: 0.9614344927013876

Fine-tuned hateBERT model detecting hate humor: 0.875784190715182

Baseline hateBERT model detecting straight hate: 0.35646062353577224

Baseline hateBERT model detecting hate Humor: 0.424090338770389

ETHICAL CONSIDERATIONS - WE 'GENUINELY' CARE

- Apart from just project work , we conducted peer survey of 20 individuals and got feedbacks.
- 3 out every 5 individual agree that exisiting models cant neccesarliy differentiate between memes and hate speech!
- Data Anonymization and Privacy Preservation
- Respect for User Sensitivity and Content
- Balance Between Detecting Hate Speech and Preserving Free Speech

FUTURE SCOPE OF WORK

1. Advanced Model Architectures:
   - Explore more sophisticated transformer-based architectures beyond BERT, such as GPT-3, RoBERTa, or XLNet, for enhanced performance in hate speech detection.
2. Multi-lingual Support:
   - Extend the model to handle multilingual text data by incorporating language-specific tokenization and pre-training on diverse language corpora.
3. Fine-tuning Strategies:
   - Investigate advanced fine-tuning strategies like ensemble learning, transfer learning from related tasks, or domain adaptation to improve model robustness and generalization.
4. Data Augmentation Techniques:
   - Implement data augmentation techniques such as back-translation, synonym replacement, or adversarial training to enhance model's ability to handle diverse linguistic variations.

THANK YOU SO MUCH & CHEERS TO CS 2731 INTRO TO NLP

3. Sai, Deyasini, Shiva, Aparna

DETECTING TOXICITY IN HINGLISH MEMES

What did we do?

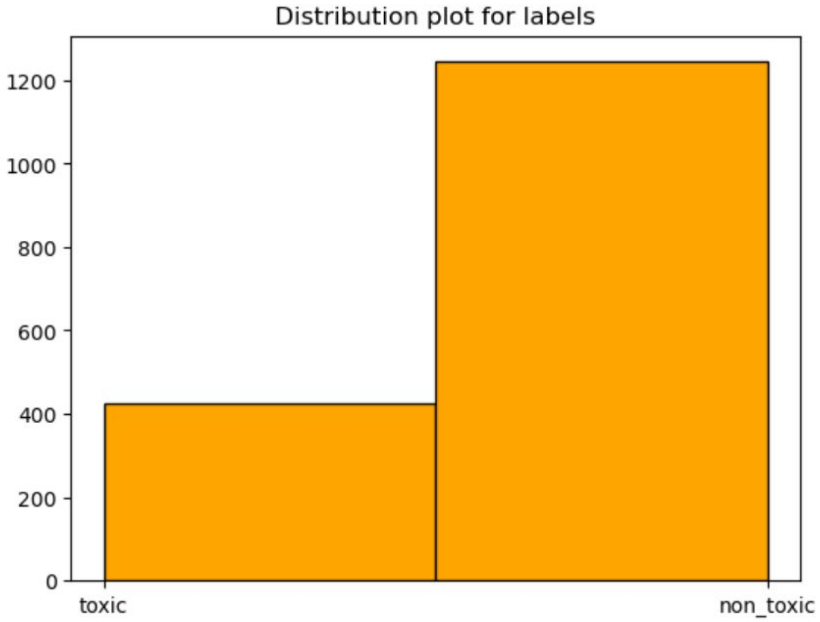
Data Collection

Data Sorting and Annotation

Approach

Outcomes

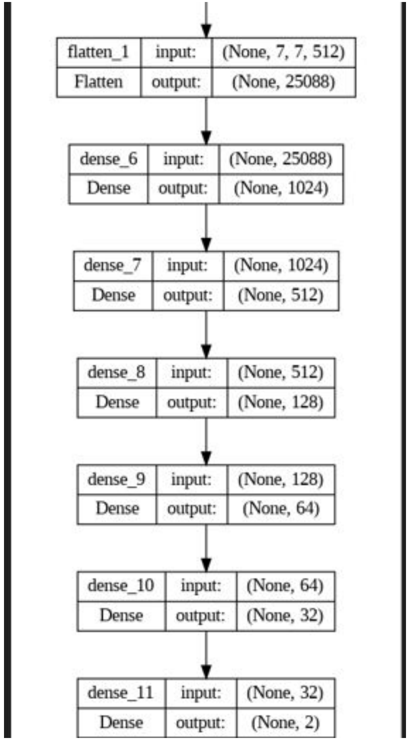
Data Viz




```
data.head()
```

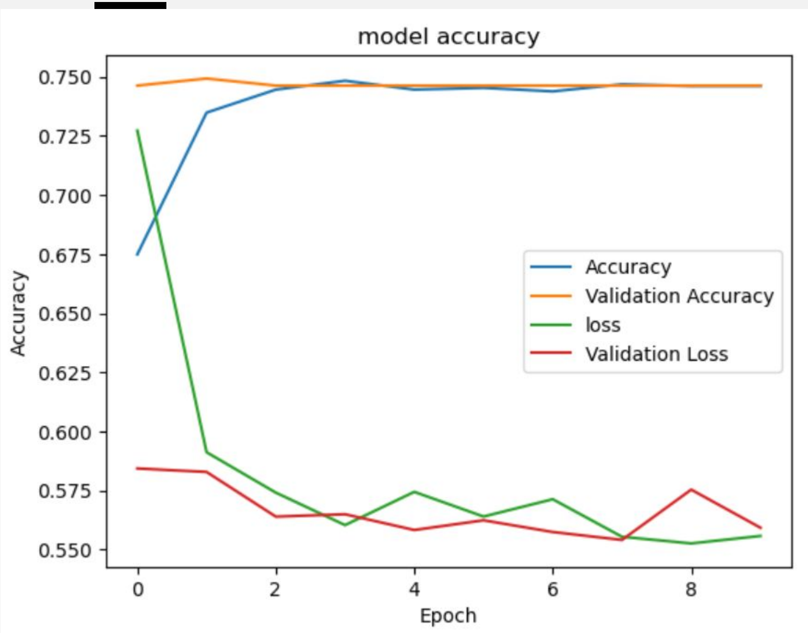
	Path	Label	Drive_type	Text
0	D:\MSIS\SEM_II\NLP\Project\NLP_Project\Images\...	toxic	Mixed	rukjao Maqbeol babuji ko tum nah maarogey f hu...
1	D:\MSIS\SEM_II\NLP\Project\NLP_Project\Images\...	toxic	Mixed	Maa ch * d denge
2	D:\MSIS\SEM_II\NLP\Project\NLP_Project\Images\...	toxic	image_driven	HEAVY DRIVER OP @heavydriver_OP Aisi Bandi Sab...
3	D:\MSIS\SEM_II\NLP\Project\NLP_Project\Images\...	toxic	Mixed	15 min Trip shuru hote hi :
4	D:\MSIS\SEM_II\NLP\Project\NLP_Project\Images\...	toxic	Mixed	RALIA HCLTe Jeet Ka Itna Bhi Ghamand Mat Karo ...

Model Architecture

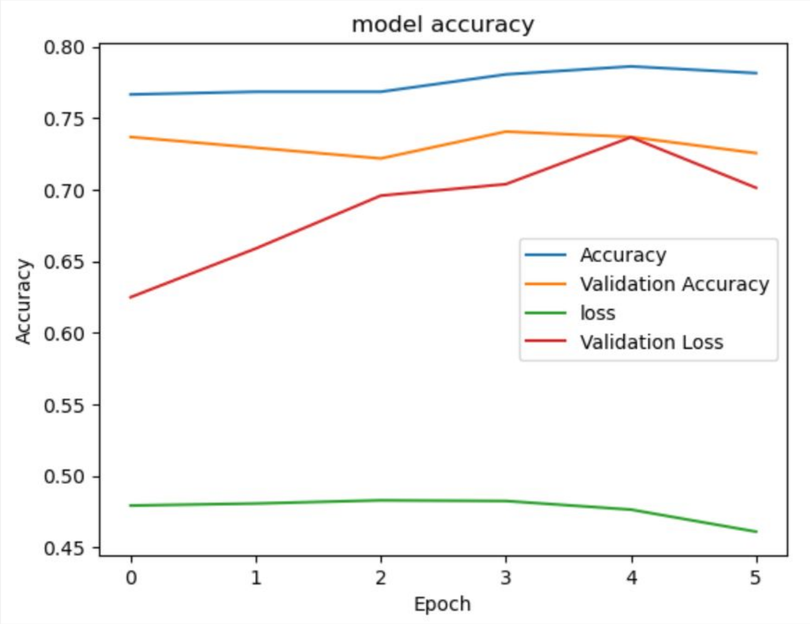



```python
for layer in base_model.layers:
    layer.trainable = False

# Adding custom layers on top of the base model
x = Flatten()(base_model.output)  # Flatten the

# Adding multiple Dense and Dropout layers
# x = Dense(2048, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(1024, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(128, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(64, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(32, activation='relu')(x)
# x = Dropout(0.5)(x)
predictions = Dense(2, activation='softmax')(x)
```

Results



Performance of model without augmented data



Performance of model with augmented data

Future additions and Variations

- The current model is trained by finetuning the VGGNet19.
- As shown in the data viz, we also have text extracted from the images and drive factor.
- We are currently using BERT for text embedding and finetuning the model on top of it which may not be an effective practice.
- We also have additional feature driveType which can be used as an attention mechanism for the text/image features and see if it can impact the performance of the model.

THANK YOU

- APARNA SRINIVASAN
- DEYASINI MITRA
- SAI VIVEK CHAVA
- SHIVA PATIBANDLA

4. Yuning, Na, Ken, Yuelong

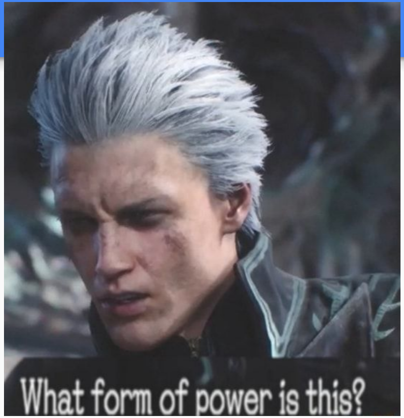
Character prediction

By Chuming Wang, Na Tang, Yuning Luo, Yuelong Xu

Motivation

- It can use as a score system for the Language cosplay
- Help identify the scam from text
- Can use for chatbot styling

- For language cosplay score system, it's a classification task with thousand or even more labels, low ACC, but it's fine.
- For identify the scam, small amount of labels, but need high ACC.



Source: https://www.reddit.com/r/DevilMayCry/comments/18mhan1/what_form_of_power_is_this/


Source: https://www.interbank.com/info/scam-trends-2022-elderly-exploitation/

Dataset

The dataset being used for project is the Cornell Movie--Dialogs Corpus.

This corpus comprises

- 220,579 conversations between
- 10,292 pairs of movie characters, involving
- 9,035 characters from
- 617 movies, with a
- total of 304,713 utterances .

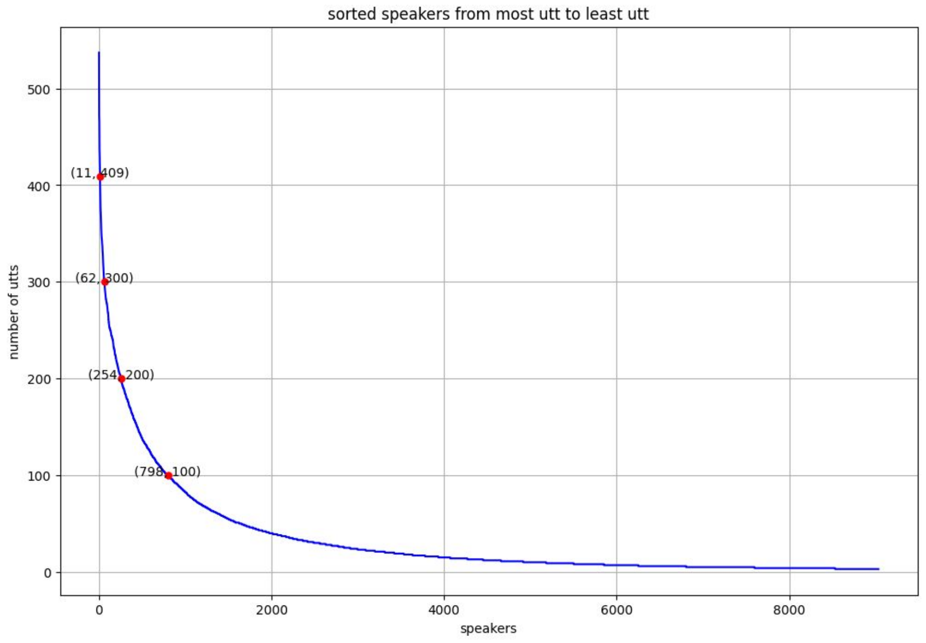
Dataset

```
utterances.head(5)
```

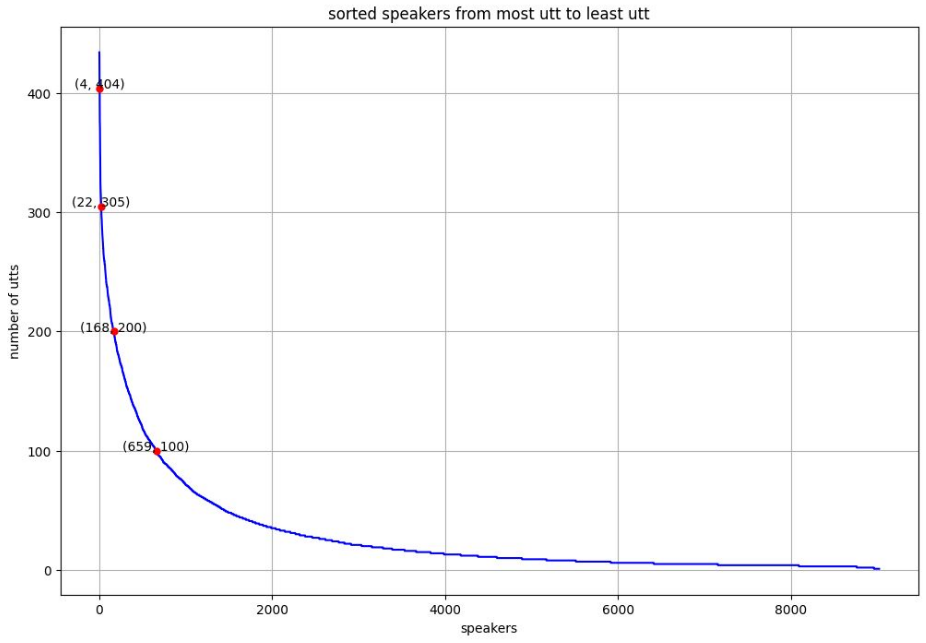
	timestamp	text	speaker	reply_to	conversation_id	meta.movie_id	meta.parsed	meta.original	vectors
id									
L1045	None	they do not!	u0	L1044	L1044	m0	[{'rt': 1, 'toks': [{'tok': 'They', 'tag': 'PR...	They do not!	[]
L1044	None	they do to!	u2	None	L1044	m0	[{'rt': 1, 'toks': [{'tok': 'They', 'tag': 'PR...	They do to!	[]
L985	None	i hope so.	u0	L984	L984	m0	[{'rt': 1, 'toks': [{'tok': 'I', 'tag': 'PRP',...	I hope so.	[]
L984	None	she okay?	u2	None	L984	m0	[{'rt': 1, 'toks': [{'tok': 'She', 'tag': 'PRP...	She okay?	[]
L925	None	let's go.	u0	L924	L924	m0	[{'rt': 0, 'toks': [{'tok': 'Let', 'tag': 'VB'...	Let's go.	[]

Data: Analysis and visualization

Before drop_duplicates



After drop_duplicates



Method - First try - Bert-base-uncased

Hyperparameters:

bert_model_name = 'bert-base-uncased'

NUM_speakers = len(speakerForModel)

MAX_length = 128

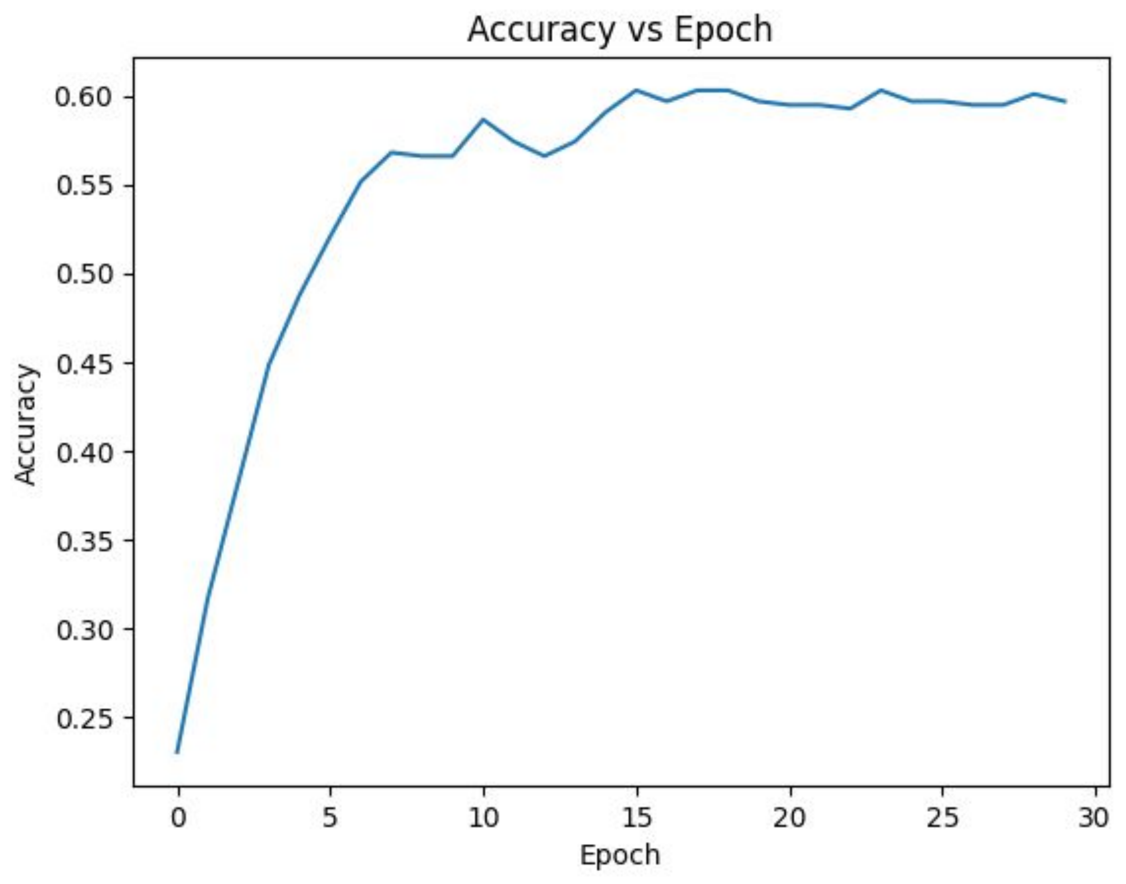
EPOCH = 30

BATCH_SIZE = 64

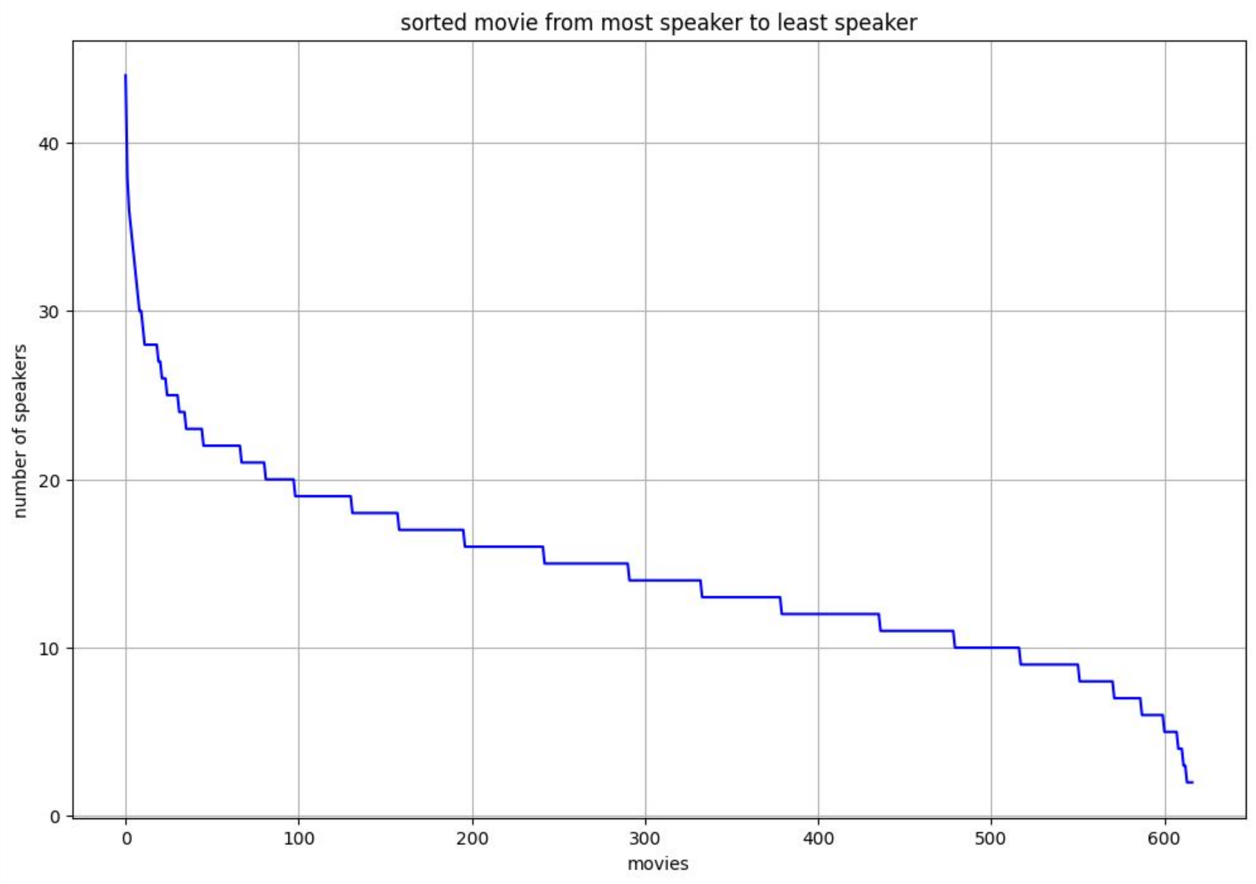
LR = 1e-5

DROP_OUT = 0.3

#5 Final Acc: 0.5967



Data: Analysis and visualization



casino : 44

lone star : 38

magnolia : 36

enemy of the state : 35

jfk : 34

the grapes of wrath : 33

the godfather: part ii : 32

mr. deeds goes to town : 31

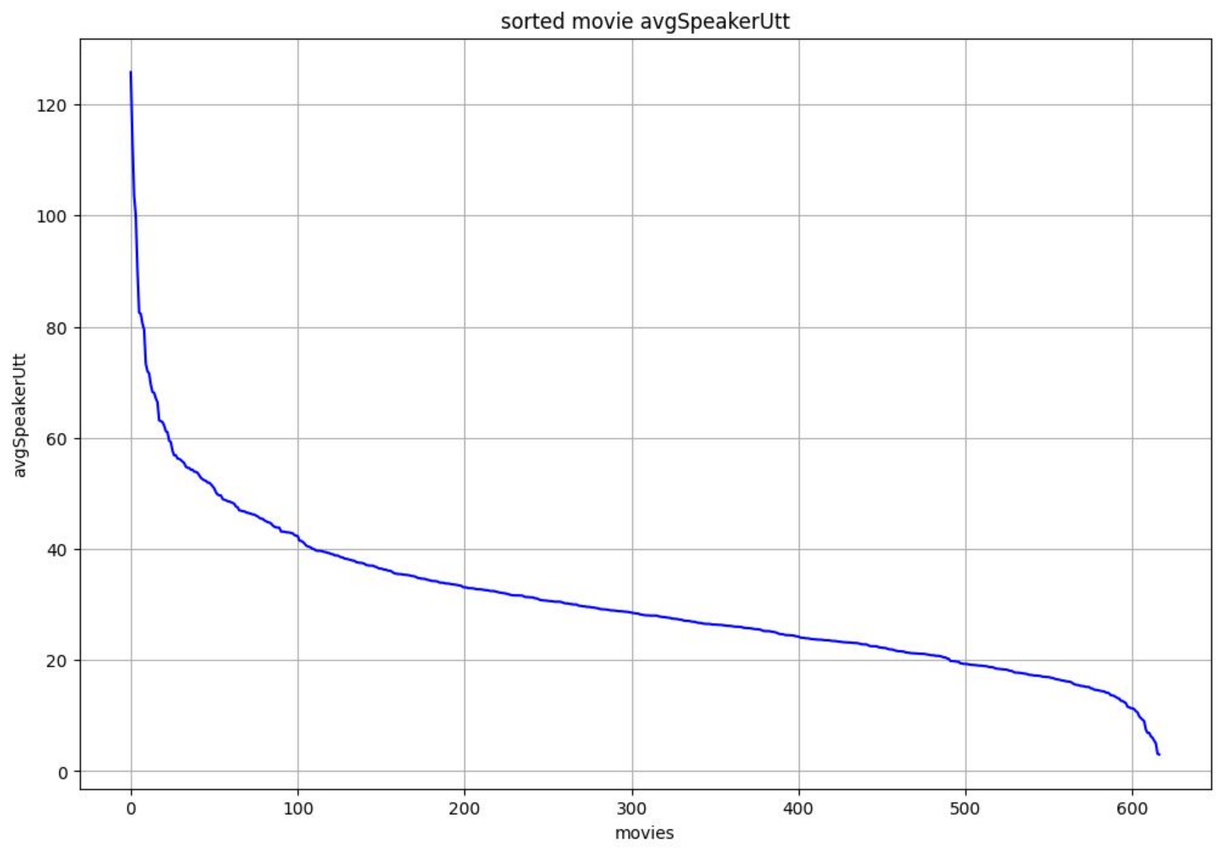
twin peaks: fire walk with me : 30

it's a wonderful life : 30

highlander : 29

Source : https://images.app.goo.gl/dYBPV5iaGq5iYjmR6

Data: Analysis and visualization



three kings : 125.8

happy birthday, wanda june : 113.4

glengarry glen ross : 103.6

entrapment : 100.0

all about eve : 90.25

stepmom : 82.6

jackie brown : 82.3

alien : 80.6

sex, lies, and videotape : 79.4

sphere : 73.3

interview with the vampire: the vampire chronicles : 72.0

Source : https://images.app.goo.gl/neND1xt5RZKYJDie7

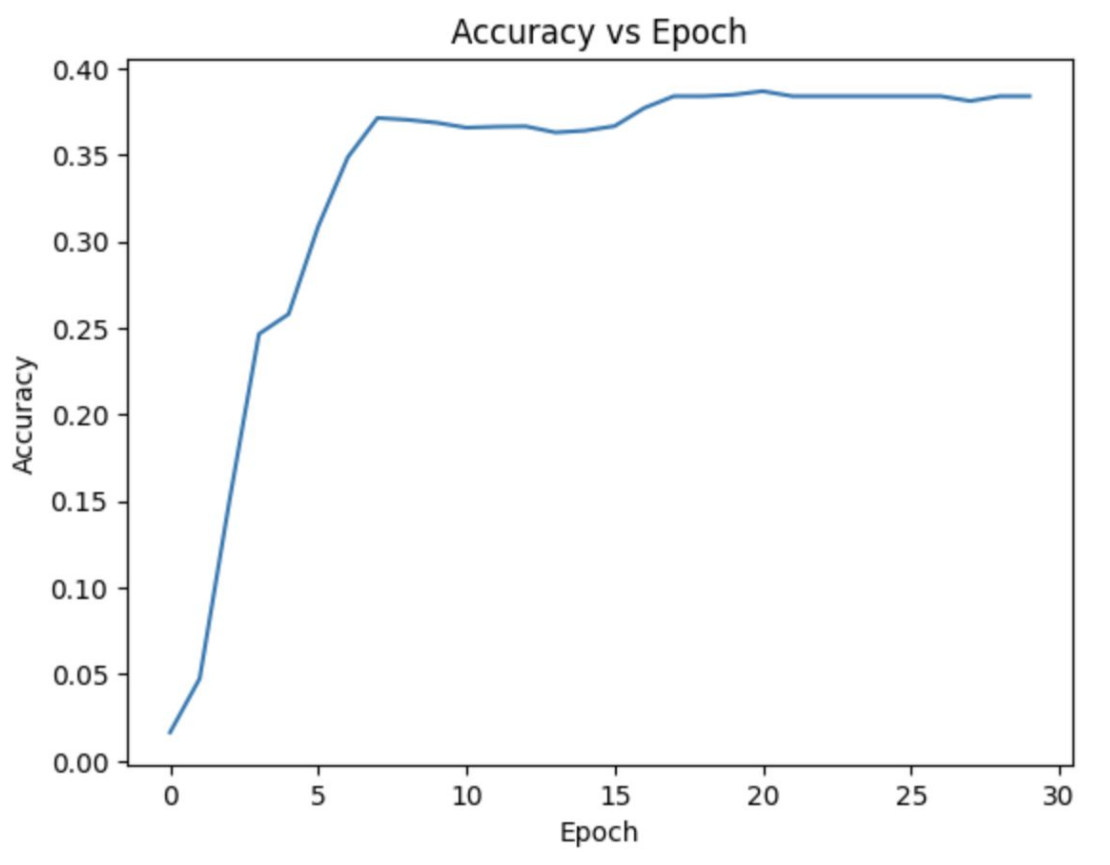
Method - LSTM

test_text: Hello, how are you?
Predicted speaker:
Speaker: u4331 ACE from casino, total_utts 465, Probability: 68.48%
Speaker: u4477 GITTES from chinatown, total_utts 428, Probability: 12.96%
Speaker: u1094 ENID from ghost world, total_utts 441, Probability: 7.19%
Speaker: u3681 ALVY from annie hall, total_utts 467, Probability: 6.32%
Speaker: u1475 JOE from innerspace, total_utts 472, Probability: 1.77%

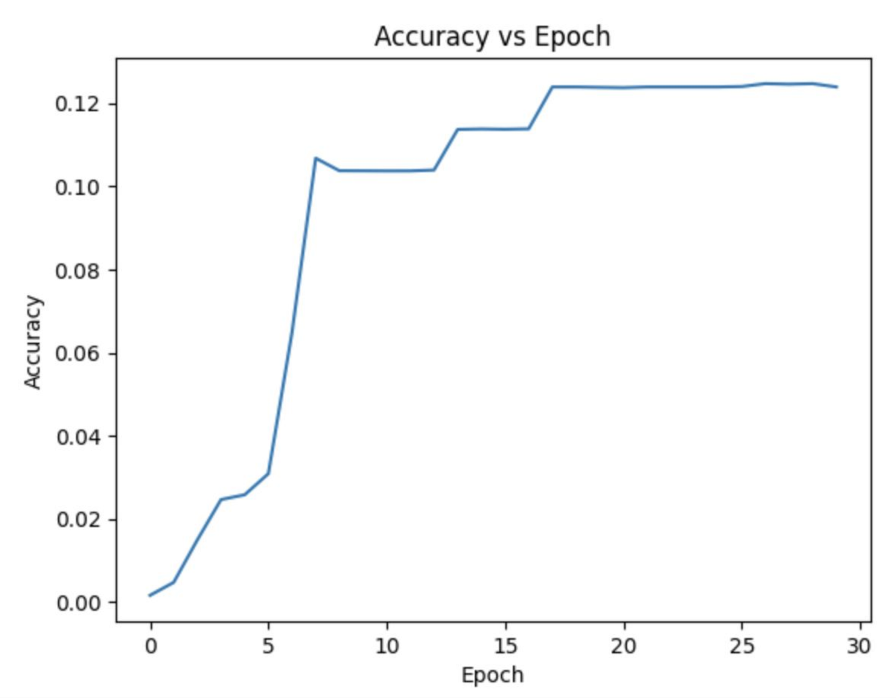
test_text: bravo six going dark
Predicted speaker:
Speaker: u4460 MASON from chill factor, total_utts 436, Probability: 44.69%
Speaker: u2340 NIXON from nixon, total_utts 434, Probability: 17.00%
Speaker: u4449 ARLO from chill factor, total_utts 425, Probability: 15.99%
Speaker: u8677 JOHN from u-turn, total_utts 414, Probability: 5.82%
Speaker: u4525 DANTE from clerks., total_utts 537, Probability: 3.01%

test_text: i am the storm that is approaching
Predicted speaker:
Speaker: u1240 HAROLD from happy birthday, wanda june, total_utts 409, Probability: 95.31%
Speaker: u2340 NIXON from nixon, total_utts 434, Probability: 0.98%
Speaker: u4460 MASON from chill factor, total_utts 436, Probability: 0.90%
Speaker: u1475 JOE from innerspace, total_utts 472, Probability: 0.55%
Speaker: u1169 BEN from the graduate, total_utts 489, Probability: 0.53%

#5 Final Acc: 0.3840



Method - LSTM



#23 Final Acc: 0.1246

Method - Bert-base-uncased, drop_duplicates

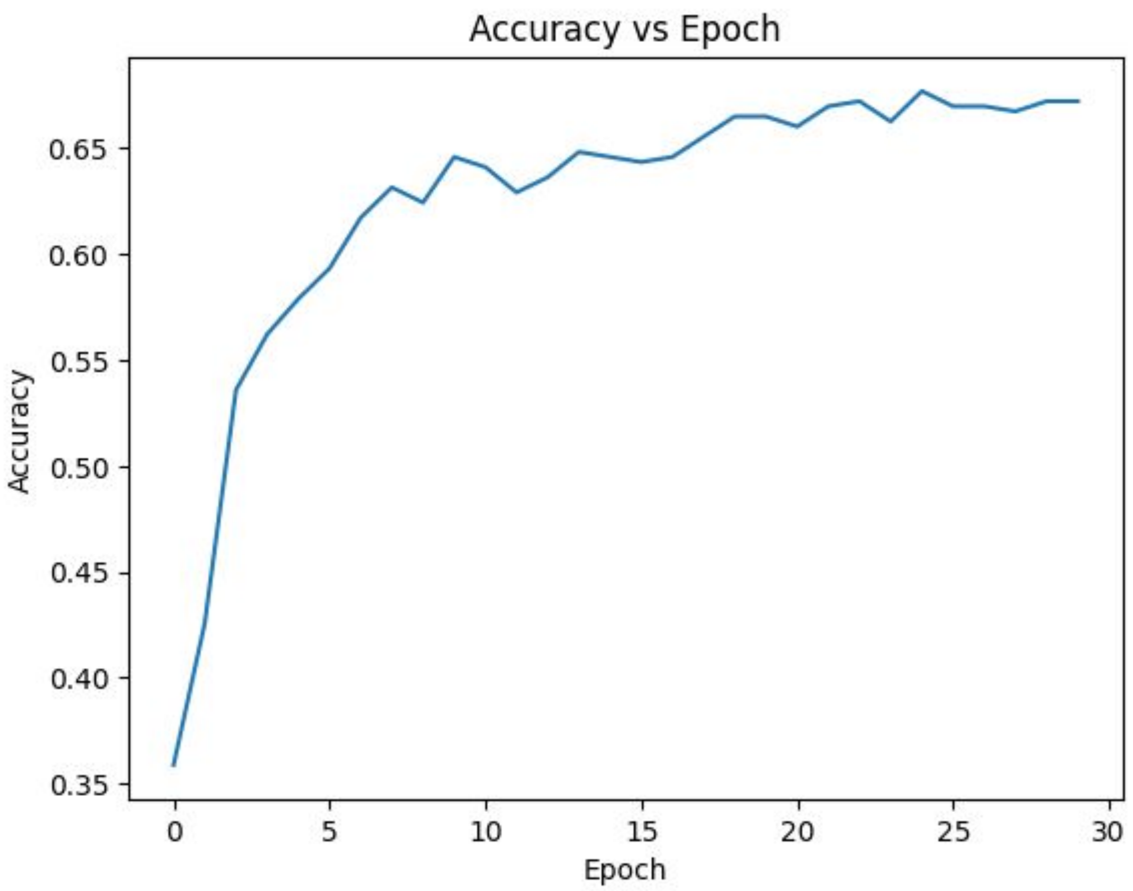
test_text: Hello, how are you?
Predicted speaker:
Speaker: u4331 ACE from casino, total_utts 465, Probability: 85.16%
Speaker: u4525 DANTE from clerks., total_utts 537, Probability: 6.42%
Speaker: u3681 ALVY from annie hall, total_utts 467, Probability: 6.11%
Speaker: u2340 NIXON from nixon, total_utts 434, Probability: 2.02%
Speaker: u1094 ENID from ghost world, total_utts 441, Probability: 0.29%

test_text: bravo six going dark
Predicted speaker:
Speaker: u4331 ACE from casino, total_utts 465, Probability: 35.22%
Speaker: u3681 ALVY from annie hall, total_utts 467, Probability: 23.90%
Speaker: u2340 NIXON from nixon, total_utts 434, Probability: 20.65%
Speaker: u4525 DANTE from clerks., total_utts 537, Probability: 18.38%
Speaker: u1094 ENID from ghost world, total_utts 441, Probability: 1.87%

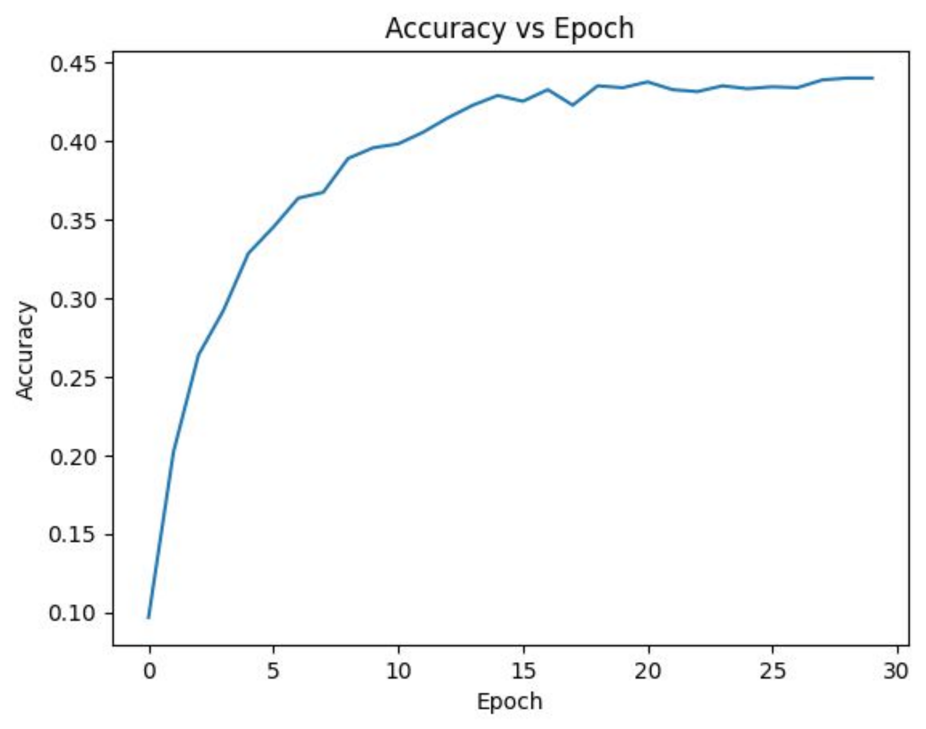
test_text: i am the storm that is approaching
Predicted speaker:
Speaker: u4331 ACE from casino, total_utts 465, Probability: 27.72%
Speaker: u3681 ALVY from annie hall, total_utts 467, Probability: 26.41%
Speaker: u4525 DANTE from clerks., total_utts 537, Probability: 21.65%
Speaker: u2340 NIXON from nixon, total_utts 434, Probability: 20.26%
Speaker: u1094 ENID from ghost world, total_utts 441, Probability: 3.95%

Hyperparameters:
bert_model_name = 'bert-base-uncased'
NUM_speakers = len(speakerForModel)
MAX_length = 128
EPOCH = 30
BATCH_SIZE = 64
LR = 1e-5
DROP_OUT = 0.3

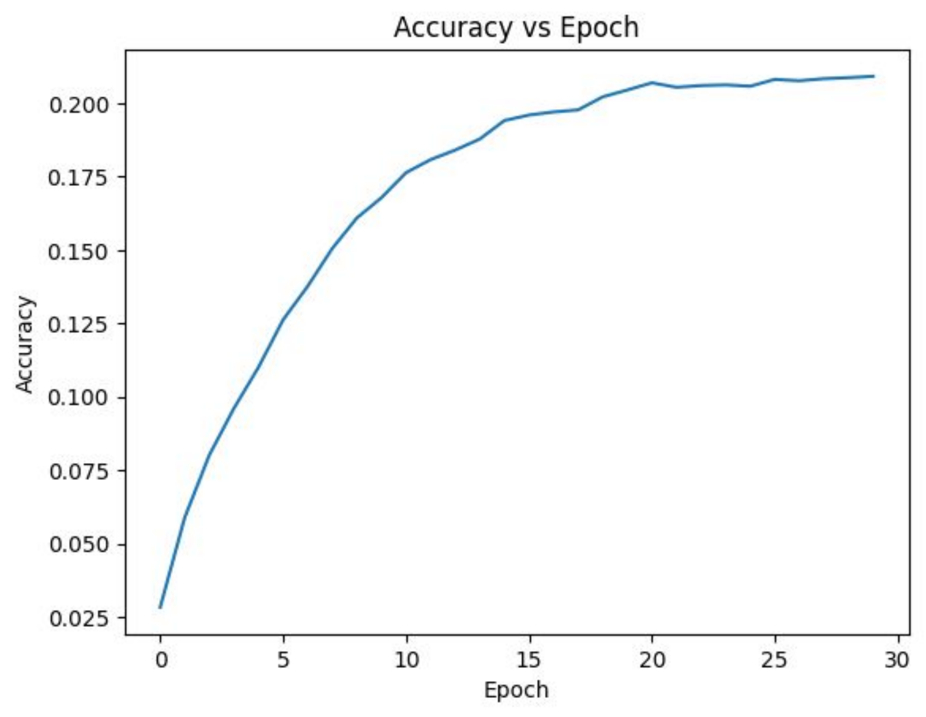
#5 Final Acc: 0.6722



Method - Bert-base-uncased, drop_duplicates



#23 Final Acc: 0.4399

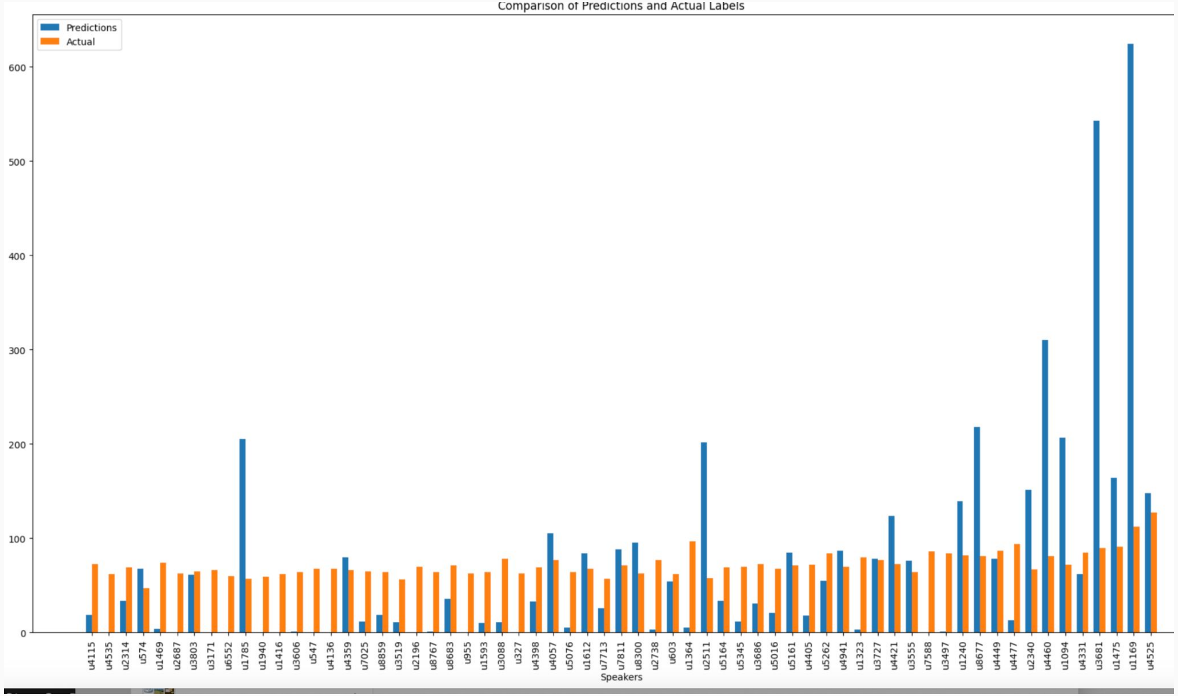


#169 Final Acc: 0.2091

Result analysis - topK

Model / Labels	5	23	169	660
LSTM	38%	12%	—	—
GRU	50%	18%	8%	7%
Bert-base-uncased (first try)	60%	—	—	—
Bert-base-uncased	67%	44%	21%	12%
Bert-large-uncased	73%	46%	—	—

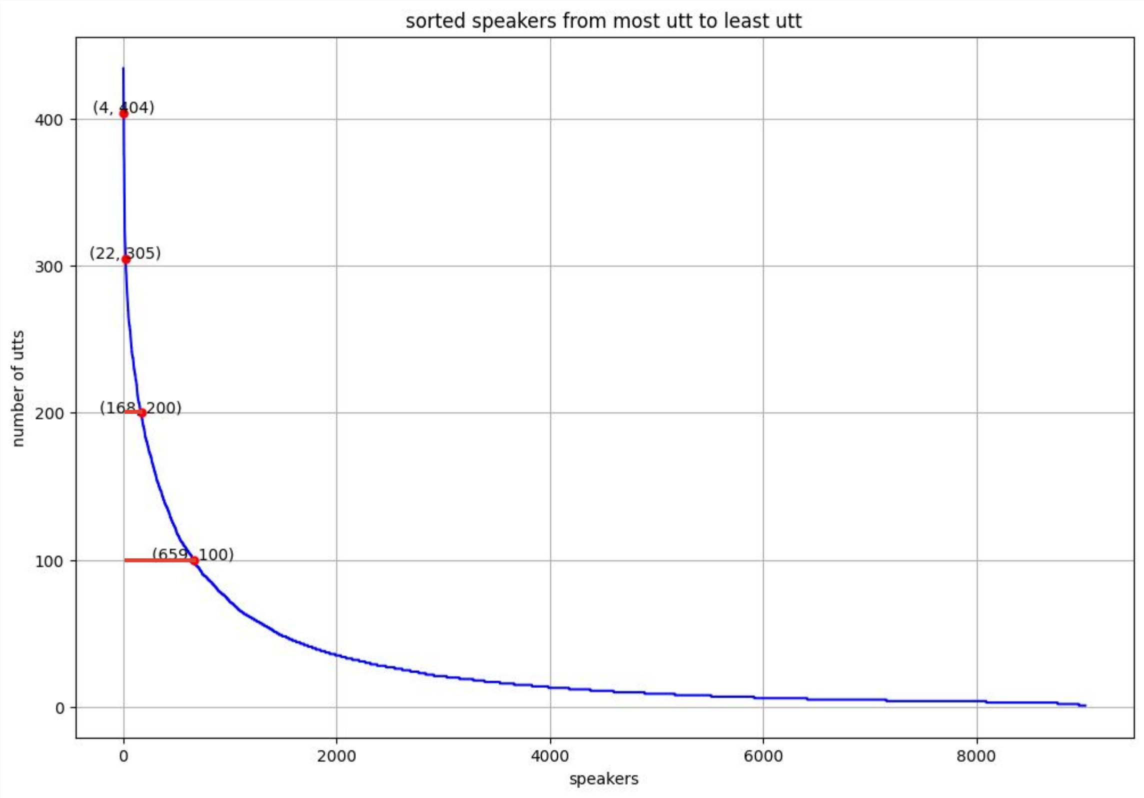
Result analysis



Predict and actual of each speakers

Result analysis - topK with Limitation

After drop_duplicates



Result analysis - topK with Limitation

topK (Baseline)	5	23	169
Bert-base-uncased	67%	44%	21%

topK with Limitation	5	23	169
Bert-base-uncased	66%	43%	19%

Next step:

- Add gender as input info,
- Add gender and movieID as input info

End

Thanks

5. Werner, Yuelyu, East, Anfeng

Error Detection in Medical Notes

By Pengyu Chen, Werner Hager, Yuelyu Ji, and Anfeng Peng

Motivation

Errors in clinical notes and misdiagnoses can cause a variety of issues, such as wasted medicine, delayed or harmful treatment, and could even result in major harm or death.

General purpose LLMs currently struggle with identifying these forms of specialized errors in text.

Explore how to implement LLMs using fact verification and commonsense reasoning for applications that require professional knowledge.

Dataset example:

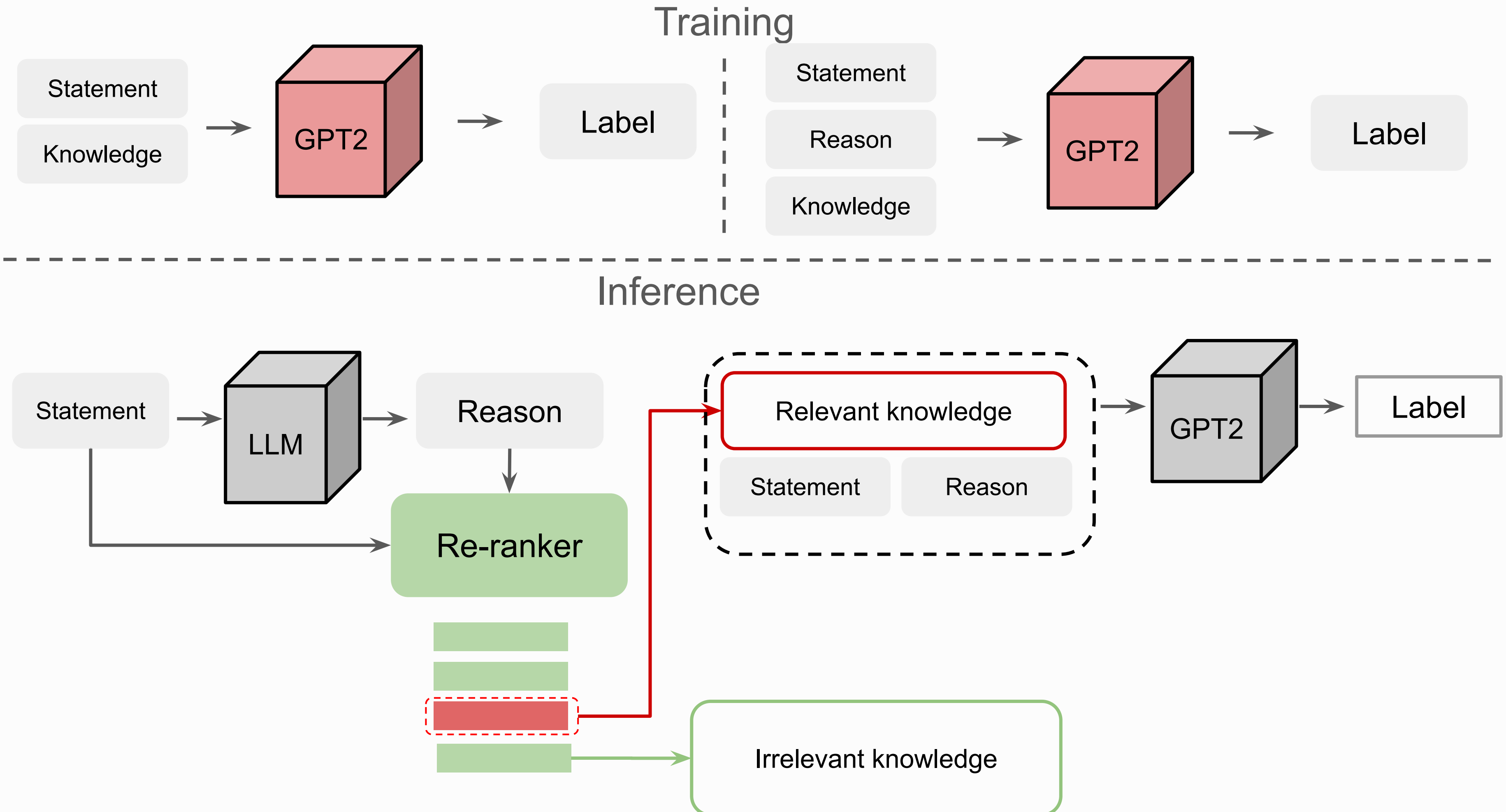
Statement: Blood cultures are sent to the laboratory. Intravenous antibiotic therapy is started. Transesophageal echocardiography shows a **large, oscillating vegetation** attached to the tricuspid valve. Causal organism is Staphylococcus epidermidis. There are multiple small vegetations attached to tips of the tricuspid valve leaflets. There is moderate tricuspid regurgitation. The left side of the heart and the ejection fraction are normal.

External knowledge:

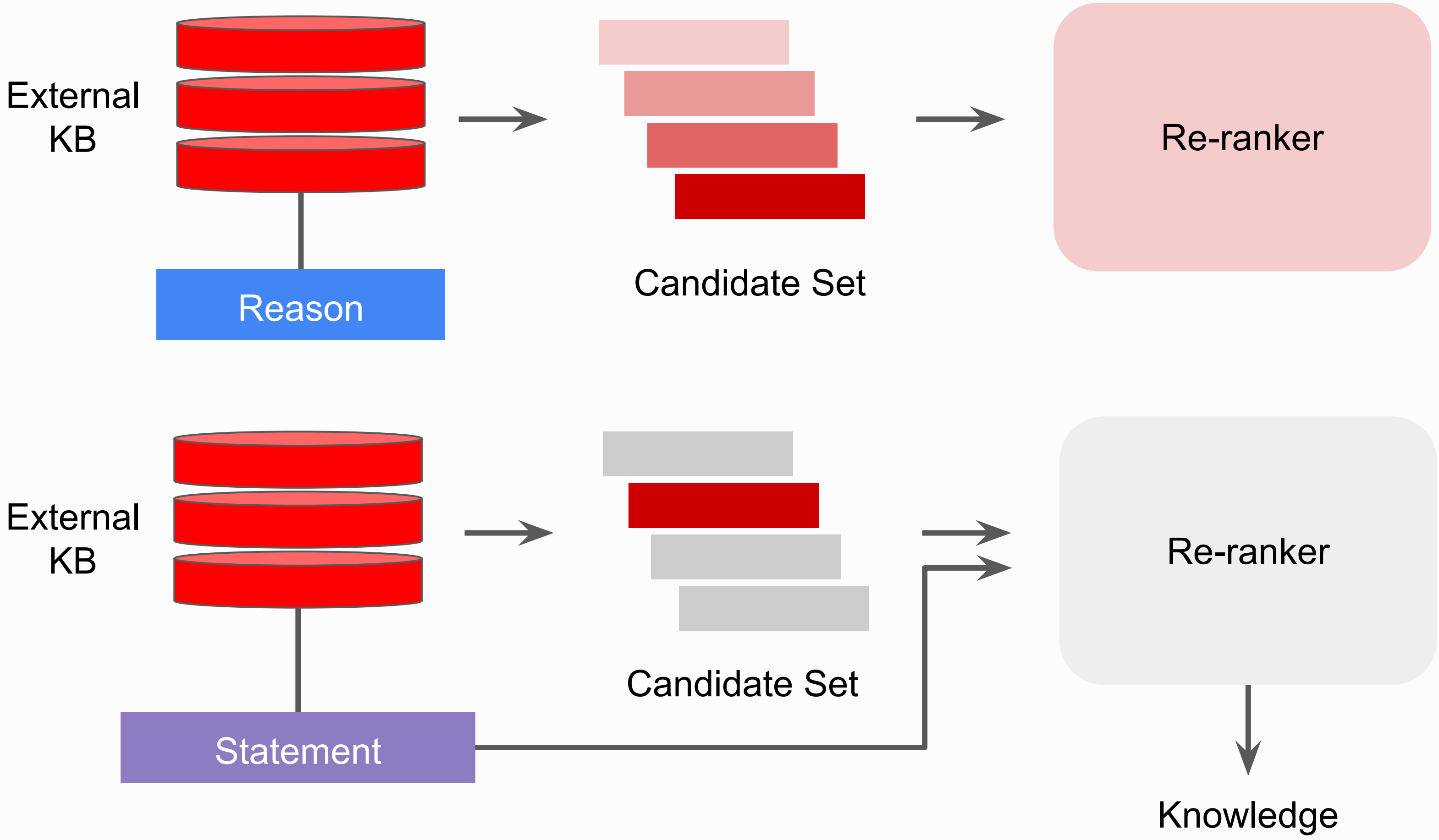
1. Staphylococcus aureus is a much more common pathogen, especially when **large, oscillating vegetations** are involved.
2. Staphylococcus epidermidis. usually associated with medical device-related infections and the infections it causes are **usually milder.**

LLM generate rationale: “Virulence Factors: Staphylococcus aureus is generally more virulent than Staphylococcus epidermidis and is more often associated with the formation of large, oscillating vegetations on native valves.”

Methods and Knowledge Augmentation



Reranker framework



Result

Method	Categories	Precision	Recall	F1-score	Accuracy
w/o rerank w/o reasons	Neg Pos	0.45 0.54	0.2 0.80	0.27 0.64	0.52
w/o rerank w reasons	Neg Pos	0.48 0.54	0.22 0.80	0.30 0.65	0.53
w rerank w reasons	Neg Pos	0.52 0.56	0.30 0.76	0.38 0.65	**0.55**

6. Hongtao, Chonghao, Sean, Bo-Chen

Active Learning with Agglomerative Clustering for Implicit Hate Speech Labeling

Sean Linton, Bo-Chen Kuo,
Chonghao Qiu, Hongtao Wang

Apr 24

University of Pittsburgh

Table of contents

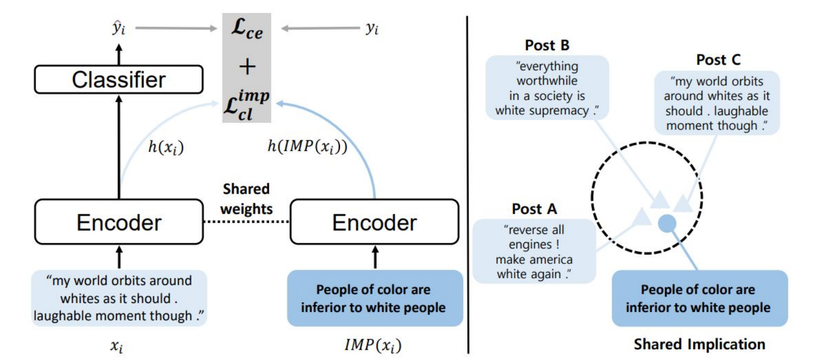
01 Motivation

01 Motivation

Generalizable Implicit Hate Speech Detection using Contrastive Learning, **Kim et al.**

$$\mathcal{L}_{overall}^{imp} = \lambda \mathcal{L}_{ce} + (1-\lambda)\mathcal{L}_{cl}^{imp}.$$

- **ImpCon**: pulls an implication and its corresponding posts close in representation space.
- **Dataset**:
  - Implicit Hate Corpus(IHC), ElSherief et al.
  - Social Bias Inference Corpus(SBIC), Sap et al.
  - Dynamically-Generated-Hate-Speech-Dataset, Vidgen et al.

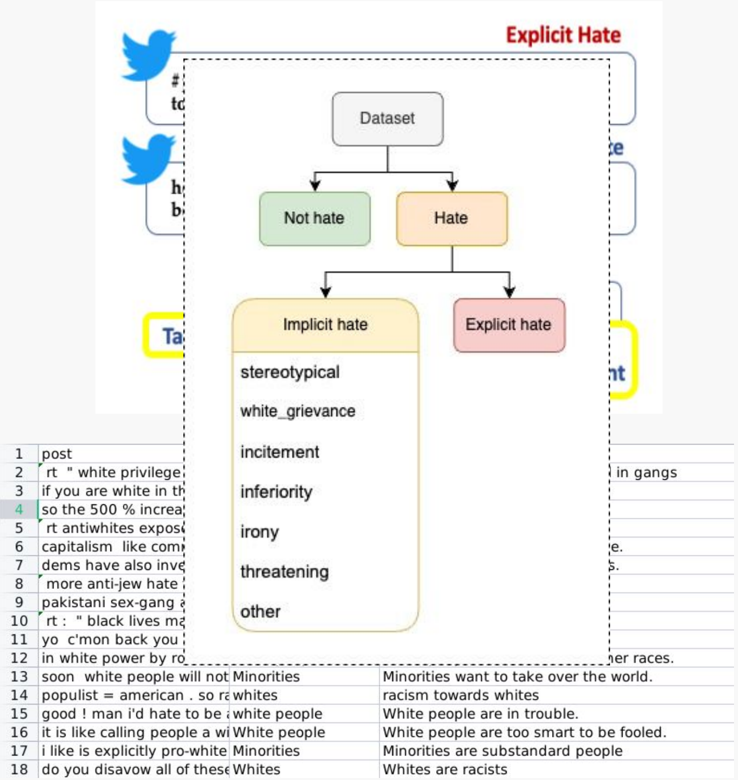


01 Motivation

Implicit Hate Corpus(IHC):

1. Divide the posts into **three** categories:
    a. Not hate
    b. Explicit hate
    c. Implicit hate

2. For each post categorized as 'Implicit hate,' label the **target** and the **implicit statement**



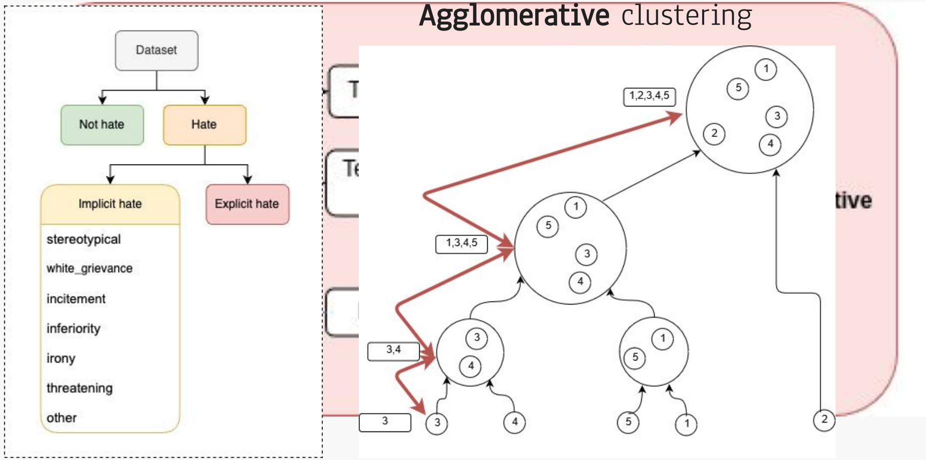
01 Motivation

Two things we can working on:

1. Utilize the explicit hate posts: Kim et al. did not.

2. Labeling is **time consuming**: we proposed a **Agglomerative** clustering based active learning method based on **ImpCon**.

1	post	target	implied_statement
2	rt " white privilege " " is v	minority groups	people in minority groups are all in gangs
3	if you are white in that stat	non-white people	non-white people are dangerous
4	so the 500 % increase in h	Immigrants	Immigrants are undesirable
5	rt antiwhites expose their	Anti-whites	Anti-whites hate white people
6	capitalism like communisr	Capitalists	Capitalists don't like white people.
7	dems have also invented a	Democrats	Democrats are manipulative liars.
8	more anti-jew hate from "	Islamists	Islamists hate Jews
9	pakistani sex-gang attacks	Pakistani people	Pakistani people are dangerous.
10	rt : " black lives matter &	BLM	BLM is violent.
11	yo c'mon back you little ill	Immigrants	Immigrants are inferior
12	in white power by rockwell	white people	White people are better than other races.
13	soon white people will not	Minorities	Minorities want to take over the world.
14	populist = american . so ra	whites	racism towards whites
15	good ! man i'd hate to be a	white people	White people are in trouble.
16	it is like calling people a wi	White people	White people are too smart to be fooled.
17	i like is explicitly pro-white	Minorities	Minorities are substandard people
18	do you disavow all of these	Whites	Whites are racists

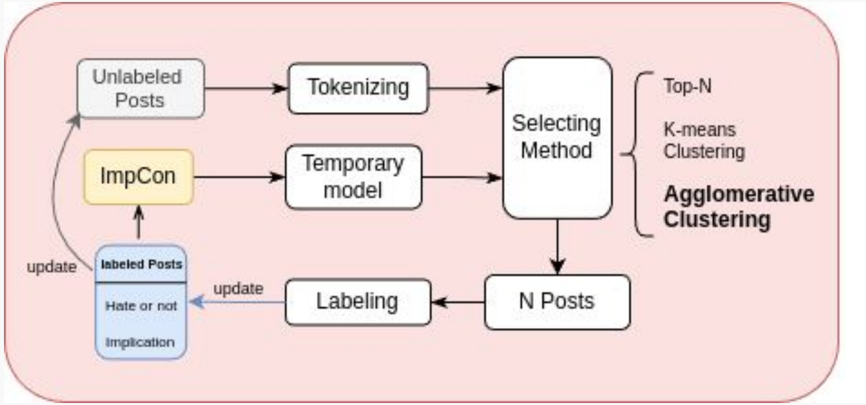


02 Methodology

According to a **survey**: A Survey of Active Learning for Natural Language Processing, Zhang et al.

1. Informativeness: **Output Uncertainty**

2. Representativeness: **Clustering method**



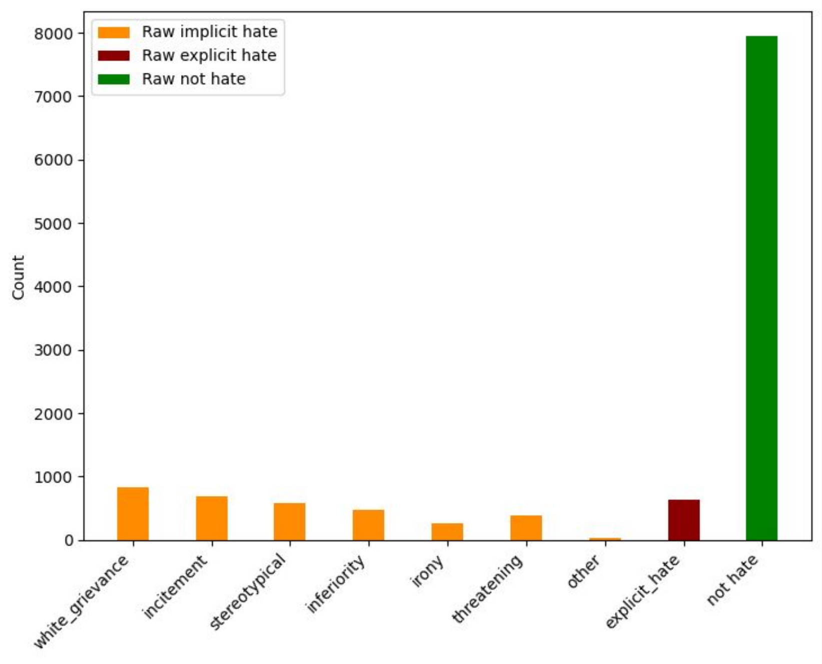
03 Results

Experiments setup:

1. Fine-tuning the BERT and HateBERT model
2. The dataset is split into **60% training**, 20% validation, and 20% test sets
   a. IHC for training
   b. Testing on IHC, SBIC and DynaHate(cross-validation)
   c. Explicit and Implicit ----> Hate(binary classifier)
3. Baseline method:
   a. Random
   b. Top-N with active learning
   c. K-means clustering with active learning(9 clusters)
   d. Agglomerative clustering with active learning

4. We try to use **5%**, **10%**, **20%**, **30%**, **40%**, and **100%** of the data for training

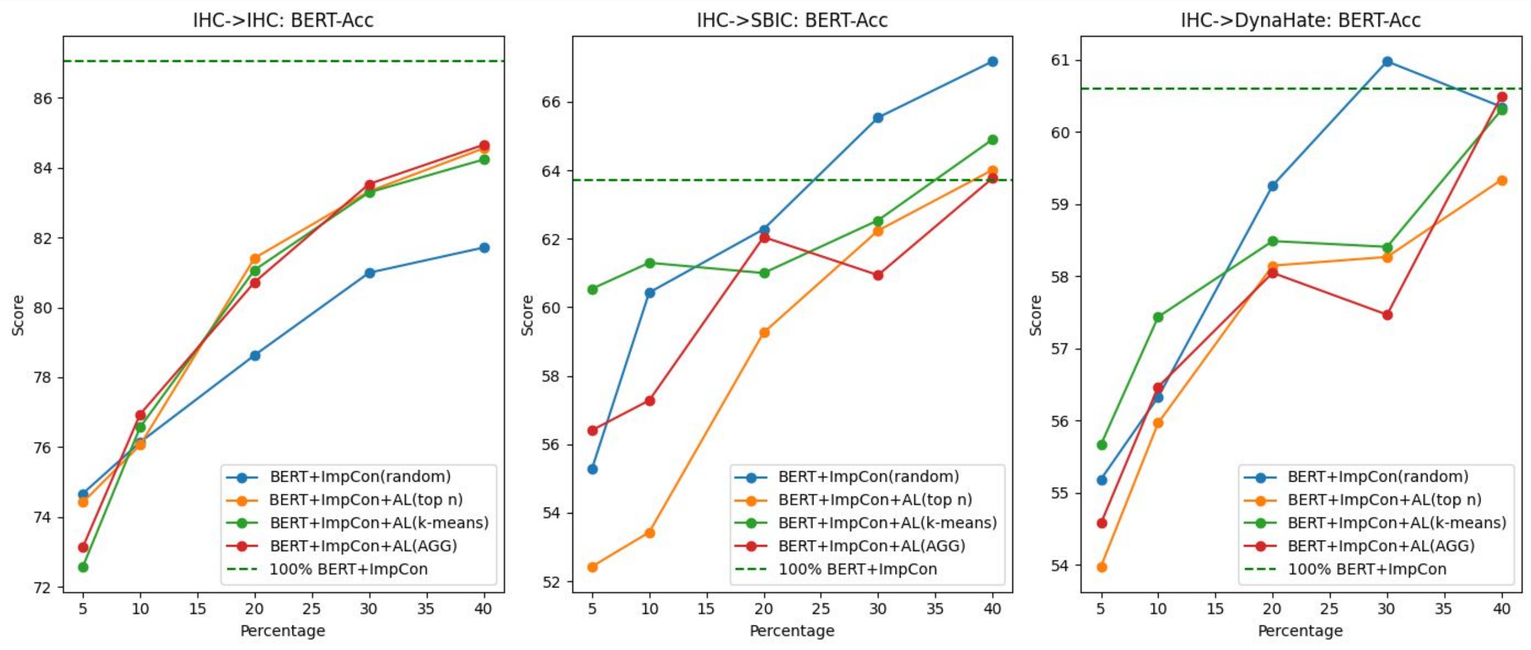
03 Results



Experiments setup:

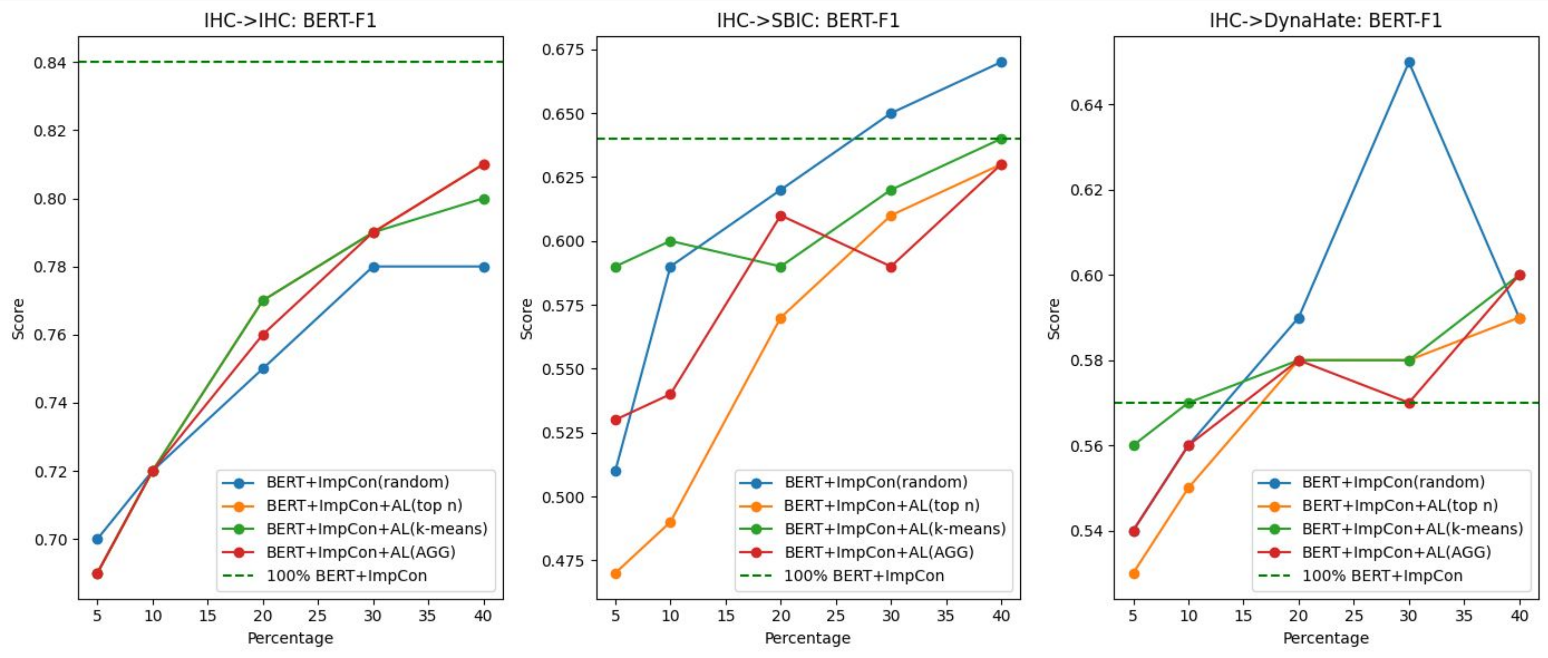
5. Data distribution in the training set:
- Explicit_hate: 632
- Implicit_hate: 3268
- Not hate: 7941

03 Results

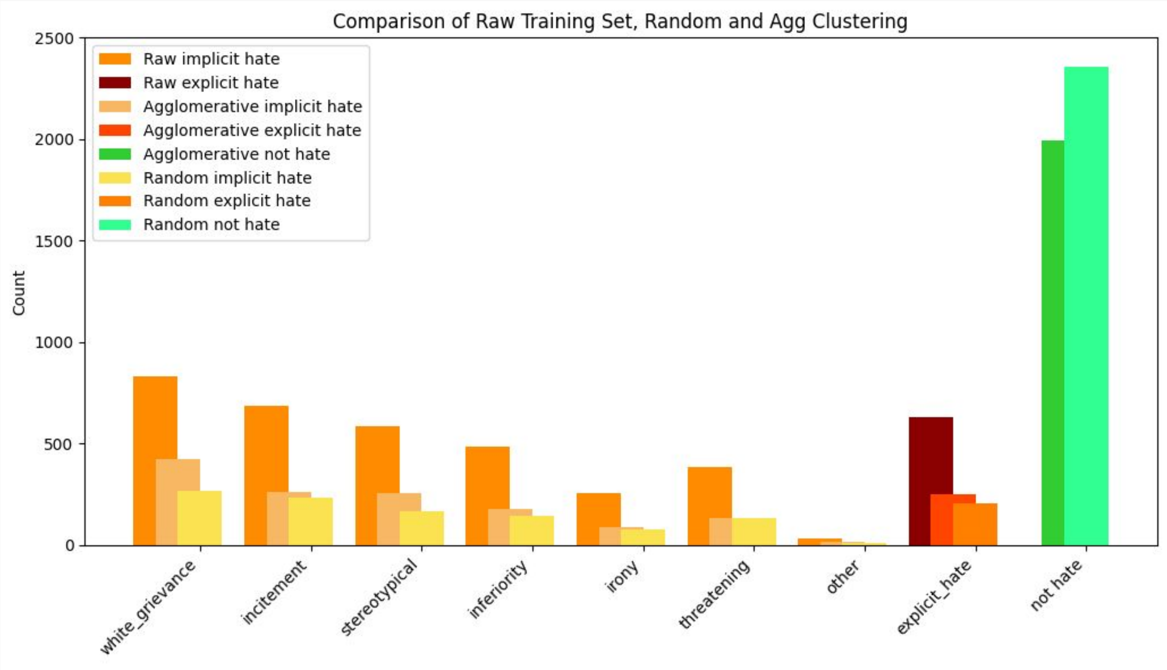


IHC->IHC: BERT-Acc
IHC->SBIC: BERT-Acc
IHC->DynaHate: BERT-Acc
Score
Percentage
BERT+ImpCon(random)
BERT+ImpCon+AL(top n)
BERT+ImpCon+AL(k-means)
BERT+ImpCon+AL(AGG)
100% BERT+ImpCon

03 Results


IHC->IHC: BERT-F1
IHC->SBIC: BERT-F1
IHC->DynaHate: BERT-F1
Score
Percentage
BERT+ImpCon(random)
BERT+ImpCon+AL(top n)
BERT+ImpCon+AL(k-means)
BERT+ImpCon+AL(AGG)
100% BERT+ImpCon

03 Results



Comparison of Raw Training Set, Random and Agg Clustering
Raw implicit hate
Raw explicit hate
Agglomerative implicit hate
Agglomerative explicit hate
Agglomerative not hate
Random implicit hate
Random explicit hate
Random not hate
Count
2500
2000
1500
1000
500
0
white_grievance
incitement
stereotypical
inferiority
irony
threatening
other
explicit_hate
not hate

03 Results

Noise in DynaHate dataset:

In the DynaHate dataset, “asian people are the shit” is labeled as “non_hate”.

Implement **noise correction** algorithm in the **future**.

04 Thank you

Q&A

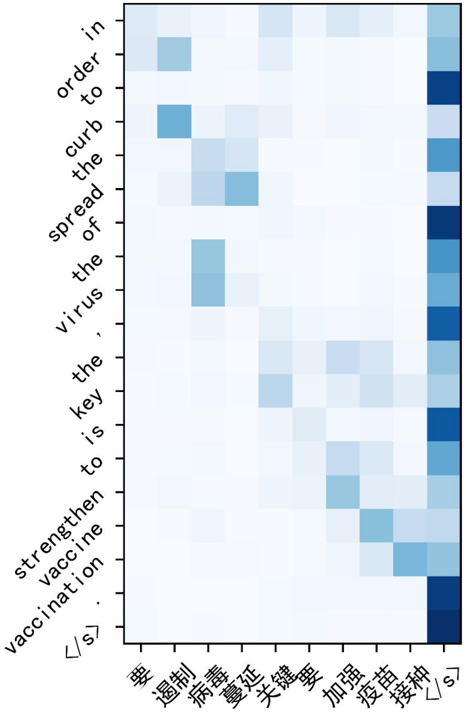
7. Shiyuan, Yingda

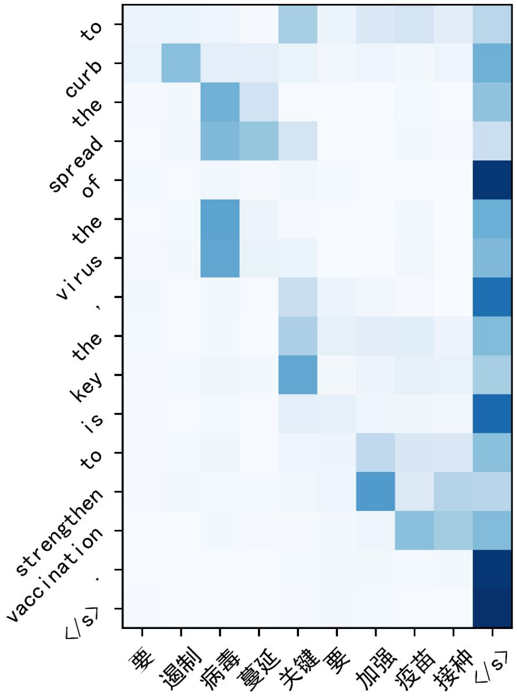
Attention Masking-Scaling Networks: Refine Attention for Transformer

Motivation

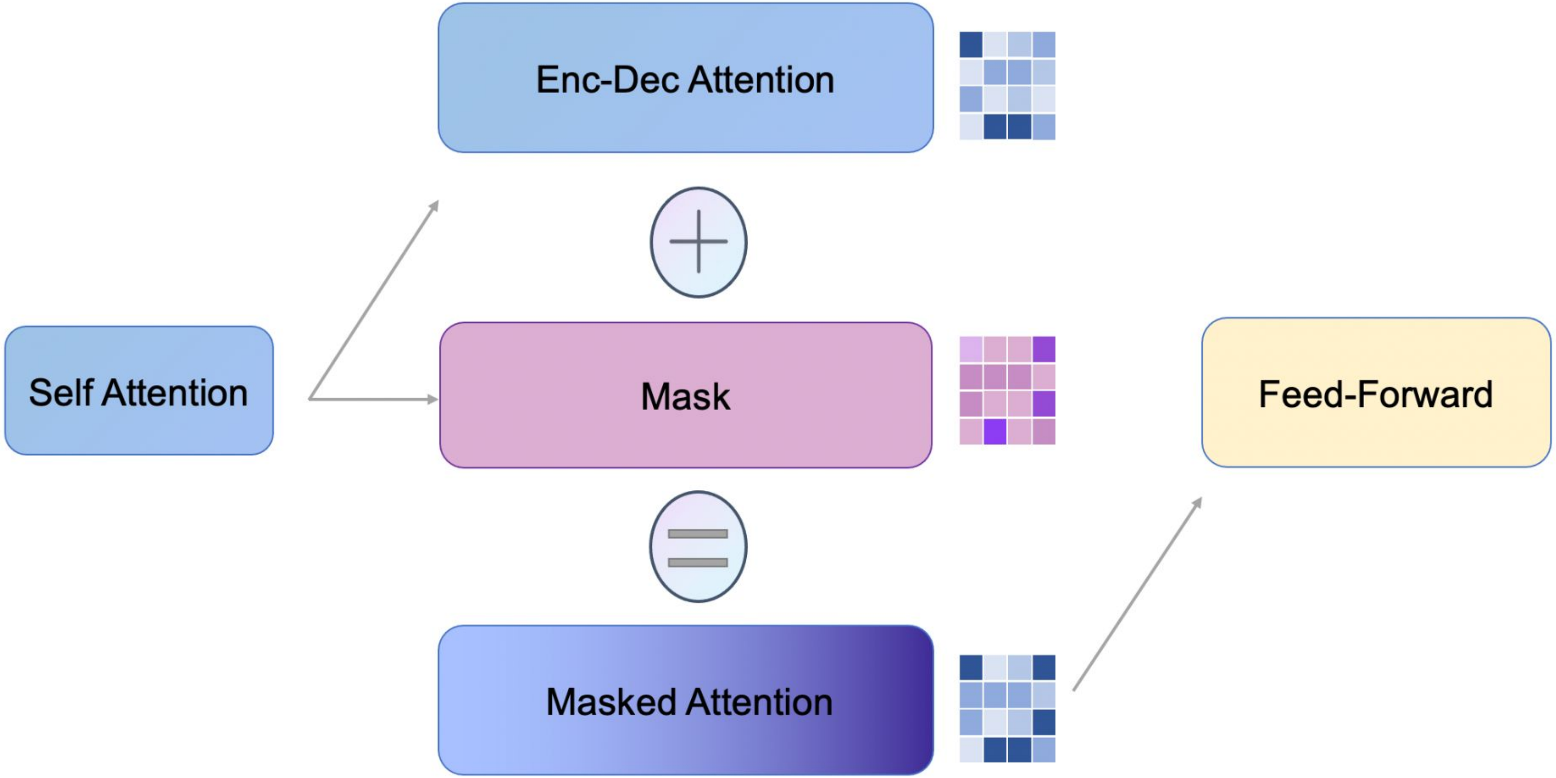
- Attention mechanisms are crucial in neural machine translation, focusing on relevant inputs for different predictions.
- Existing attention mechanisms have vulnerabilities, such as being easily perturbed, leading to translation inaccuracies.
- Previous modifications to attention mechanisms were isolated, lacking integration with other model components.
- Importance of refining attention to improve the detection of definitive information and overall translation performance.

Introduction

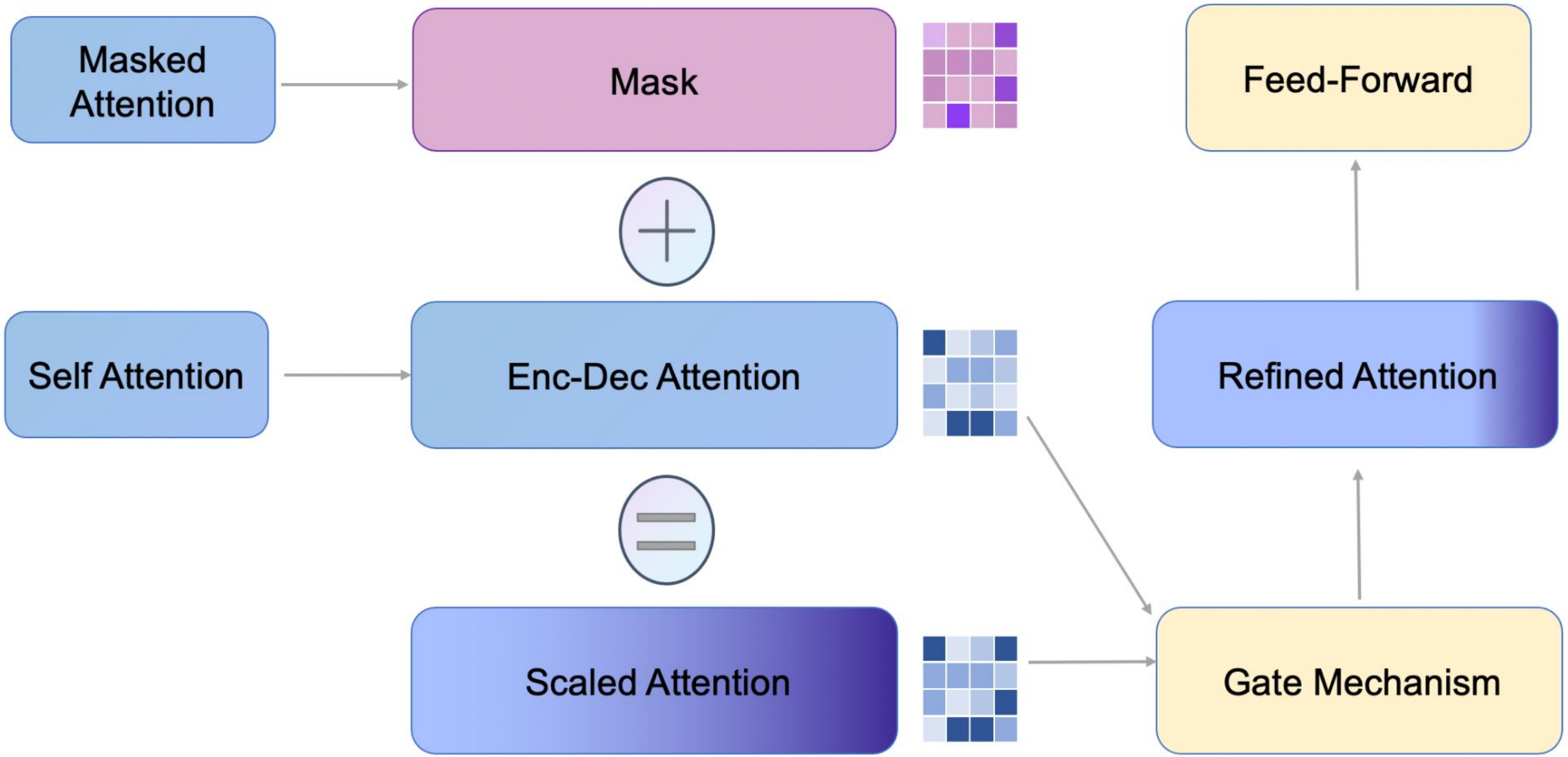




Methods

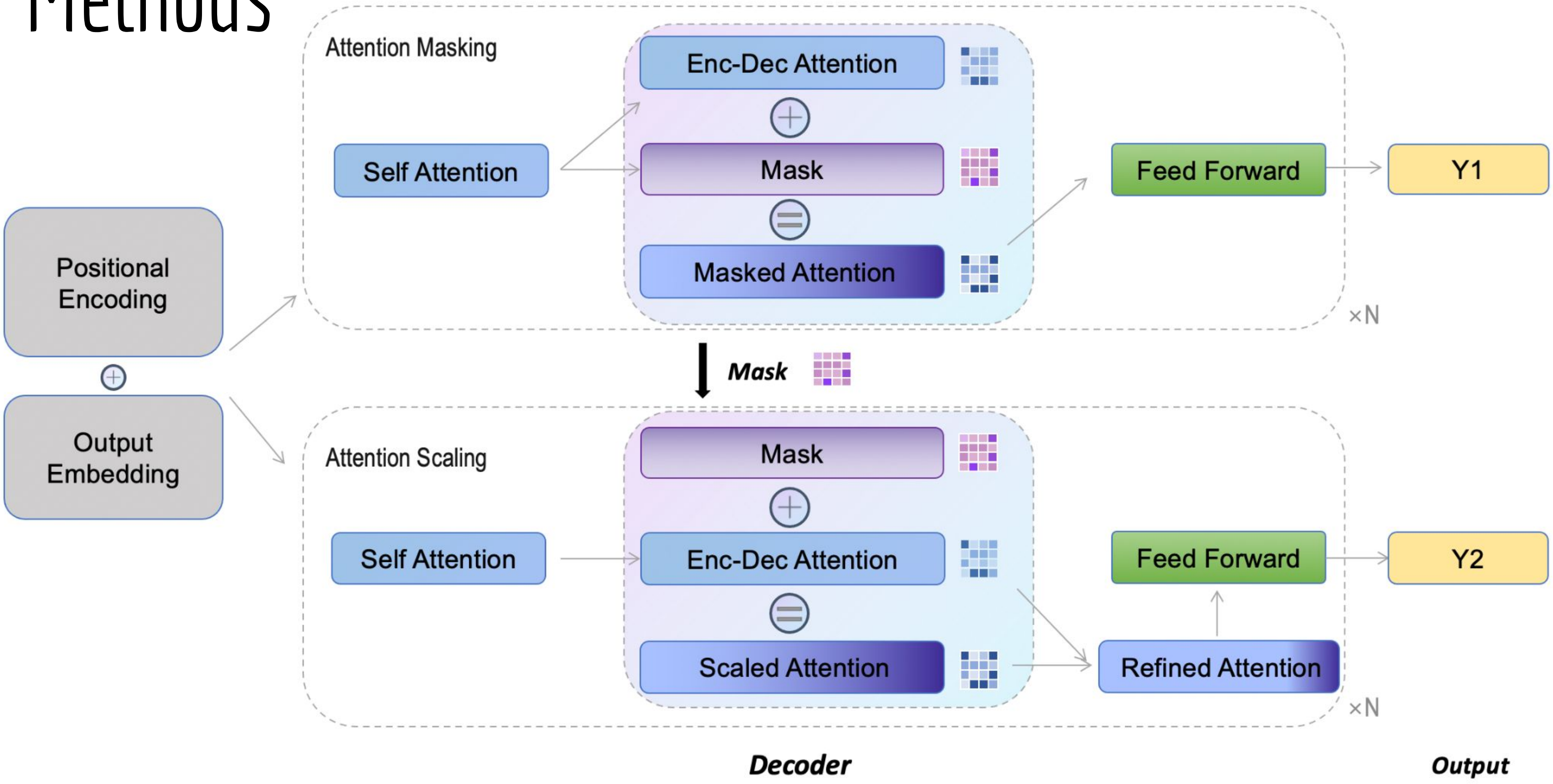

Enc-Dec Attention
Self Attention
Mask
Feed-Forward
Masked Attention

Methods

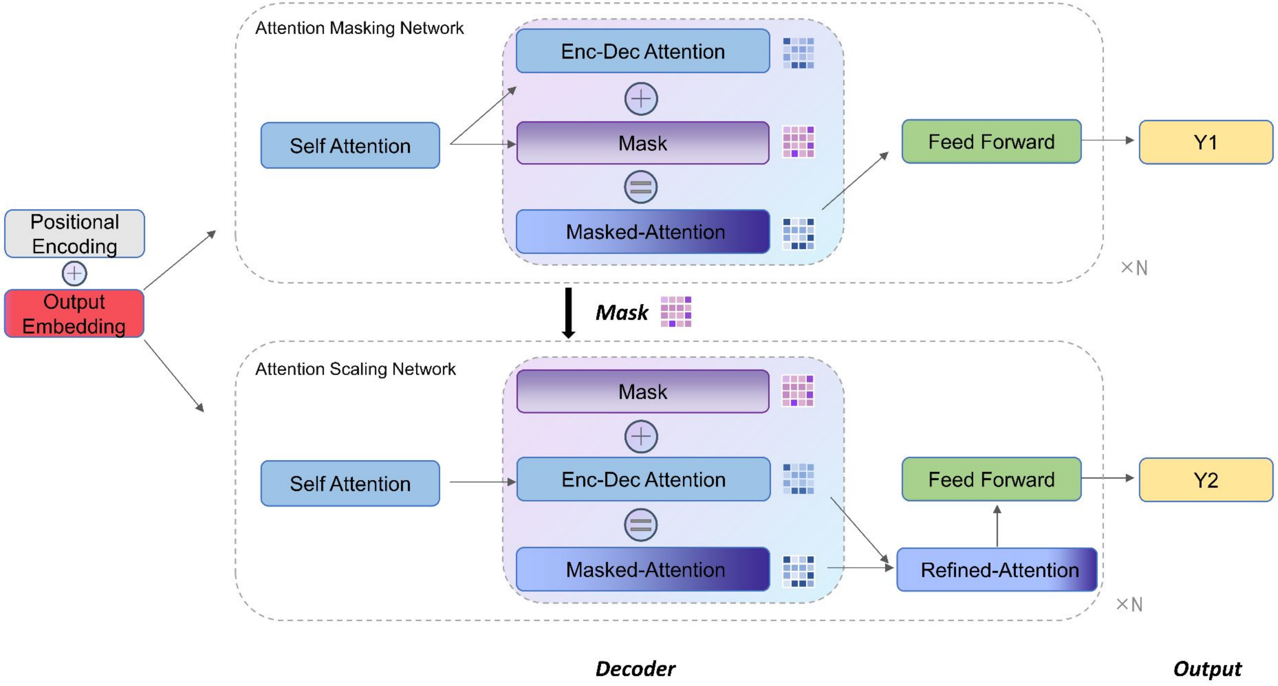


Masked Attention
Mask
Self Attention
Enc-Dec Attention
Scaled Attention
Feed-Forward
Refined Attention
Gate Mechanism

Methods


Attention Masking
Enc-Dec Attention
Self Attention
Mask
Masked Attention
Feed Forward
Y1
×N
Positional Encoding
Output Embedding
Mask
Attention Scaling
Mask
Self Attention
Enc-Dec Attention
Scaled Attention
Feed Forward
Refined Attention
Y2
×N
Decoder
Output

Methods



- AMSN jointly learns Mask Matrix and standard attention. AMN and ASN are trained simultaneously to obtain masked and scaled attention weights.
- AMN used to distinguish the decisive inputs and create a Mask Matrix
- The AMN-trained common factor mask is fed into the ASN, it uses the Mask Matrix to scale attention weights during training.

Formulation

Attention mechanisms map a query and a set of key-value pairs to an output shown in Equation 1.

$$Attention(Q, K, V) = \mathcal{A}(Q, K)V$$
$$\mathcal{A}(Q, K) = \left[\frac{\exp(Q_i K_j^T / \sqrt{d_k})}{\sum_k \exp(Q_i K_k^T / \sqrt{d_k})} \right] \quad (1)$$

where queries Q, keys K and values $V \in \mathbb{R}^{T \times d_k}$ are all matrices.

On the basis of attention function in Equation 1, we define the mask attention function:

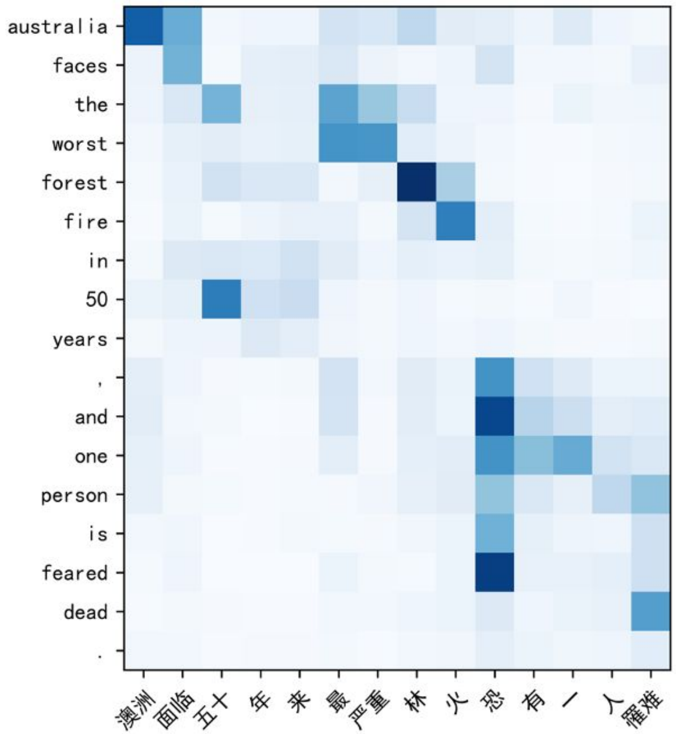
$$MaskAttention(Q, K, V) = \mathcal{A}_M(Q, K)V$$
$$\mathcal{A}_M(Q, K) = M \odot \left(\mathcal{A}(Q, K) - \overline{\mathcal{A}}\right) + \overline{\mathcal{A}} \quad (3)$$

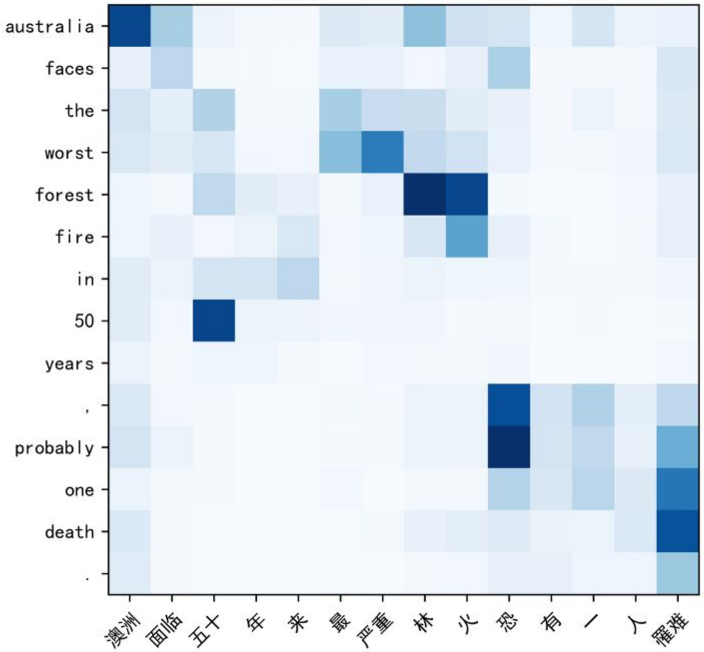
$$M_{i,j} = \sigma\left(Q_{M_i} K_{M_j}^T / \sqrt{d_k}\right) \quad (4)$$

$$\mathcal{A}_S = \mathcal{A}(Q, K) \odot exp\left(M\right) \quad (7)$$

$$\mathcal{A}_S = \mathcal{A}(Q, K) \odot exp\left(1 - M\right) \quad (8)$$

Results


australia
faces
the
worst
forest
fire
in
50
years
,
and
one
person
is
feared
dead
.
澳洲
面临
五十
年
来
最
严重
林
火
恐
有
一
人
罹难


australia
faces
the
worst
forest
fire
in
50
years
,
probably
one
death
.
澳洲
面临
五十
年
来
最
严重
林
火
恐
有
一
人
罹难

8. Nick, Arushi, Trung

Evaluating LLMs for Biomedical Lay Summarization

Nick Littlefield, Arushi Sharma, Trung Tran

Motivation

- Lay summarization of Biomedical publications is of interest to wide range of audiences.
- Technical and specialist language makes it hard for non-expert to understand.

Our goal: to develop an abstractive summarization model to generate lay summaries for non-technical people.

Datasets

Original plan: 2 datasets PLOS and eLife. Each data contains article, technical abstract and lay summary.

Dataset	# Train	# Dev	# Test
PLOS	24,773	1,376	142
eLife	4,346	241	142

Lay summary characteristics:

- PLOS: written by article's author, from 5 peer-reviewed journals
- eLife: written by editor in consultation with author, from eLife journals

Now: We focus on the eLife dataset

Methods

- Instruction finetuning
  - Training Gemma 2B, 7B models with instructions + training dataset examples
  - LoRA technique for efficient finetuning

- MapReduce BVR technique
  - Gemma, T5 - Long models (pretrained on PubMed documents)

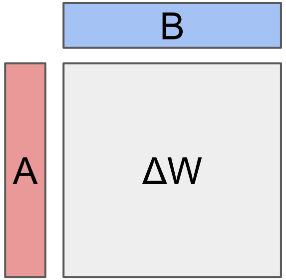
- BioGPT - inference

LoRA: Low Rank Adaptation Finetuning

Method for parameter efficient fine-tuning

- Given pretrained model weights *W*,
- Want to learn fine-tuned weights *W'*
- We instead learn a low-rank factorization

$$A \in \mathbb{R}^{m \times r}$$
$$B \in \mathbb{R}^{r \times n}$$
$$r \ll m, n$$



Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models." arXiv:2106.09685 [cs.CL]

Best Vector Representation Summarization

Useful for the summarization of long documents

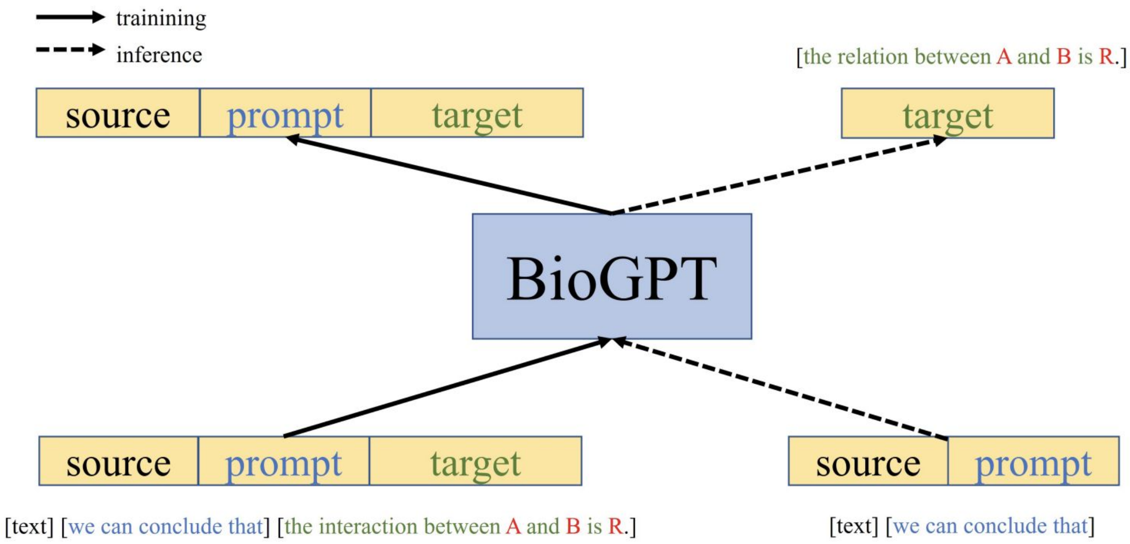
Used for Gemma 7B and Long-T5

Steps:

1. Load in article
2. Split article into chunks
3. Get embedding for each chunk
4. Cluster embeddings to see which vectors are similar
5. Choose the most representative document of each cluster
6. Summarize those documents (Map)
7. Summarize individual summaries into one single summary (Reduce)

BioGPT - inference

Leverage prompting techniques to fine-tune and infer with BioGPT



Luo et al. “BioGPT: Generative Pre-Trained Transformer for Biomedical Text Generation and Mining” arXiv:2210.10341v3 [cs.CL]

Training setup

HuggingFace library

Computation Requirements: A100 GPU (80GB)

Gemma-7B-it LoRA Finetuning time: 9 hours 29 minutes.

Long-T5 LoRA Finetuning time: 2 hours 25 minutes

Gemma 2B -it LoRA finetuned

Summary 1: hairc increa …

Summary 2: The spermathecal plate of the Brazilian cave insect Neotrogla is composed of two plate-like sclerites , a closure element , and in-and-outflow canals, which can be used to coercively anchor males and obtain voluminous semen. The switching valve is a morphological novelty enabling females to compete for seminal gifts in their nutrient-poor cave habitats through long copulation times and multiple injections. In addition to the reversal of the female, there is no evidence for a gain in fitness by actively selecting for male-derived nuptial gifts in this genus, which is considered to be an adaptation for direct benefits (i . e , propensity for multiple mating) and a copulating pair of N . aurora and truncata, respectively . The resilin-rich region surrounding the plate is also located at the dorsal connection, which allows the female to receive more nutrition from the same or other males, and the sperm is also used to be used in the sex-results, but not for the female-derived , but also to be positive and positive positive positives in the context of the ad s a positive and negative positive positive negative positives, positives and positive negatives for the , sassa st and sss . the d , the neosada , or syssids, ess-semsaas sesa-salal , in sneadsse-sassalaaadasas-s-adoaaa-, and
oseacsacss:1322a,gstssiscdagnsisasa, and
and-syeossyadaeas-ascentcentcentscentssodsnacadeaiamamao-ssamsydnealnsddscentsoansals:--s%%%%s%%-d-es-damia>>>>>>,
and=msdganiaanaamasiaa---gtcecsodcidsadedesamamama%%-%%a%addaadea:a1>2>s>d>%>,>->1-2%2-%,-%d%%s%%d%se%%,
and/oraltde.::a)3-g>nodiasea-ag-n-ccdts.sgd:sen2::-===1:343>a=2;/\xadenayataesmenoncentcent%centcent%%centcentatemega,abiceinys,dg:sst%:3%%
:%%;>=>:>3d:%1d2dd)s;=%>%//s/>/>>;>==12=,=3=%%/%, >100%% and//-/s>,>>/>-.>>->>a>e>100;
and)-:1=33;3:2),-35%%c%a...-)>a>.)>>.>c>m>t>;>>14>>:>>%%>> and=->///>/>, =.>)>.-)-:=%/%3%=/=-=d=s=a/d/ap0.05())(()%3/2/=202,=:=3>>
>:>;>-->,>d>>34-1,3,224,1;);;/)=a=224;-,.);;.31a2.=064%=/3a/a;-i>>=>>2>>(=0.03/m/g/d/%%==%,>; and> >g/c=> >>>
>>(>>)/(/;1\xad2\xad=74/,36=4=;=(=):())):---a-da%;(3)3(1/1)2(2 =172 and2 2? ==6781011 and/,/o/e/./i>(>0.05>17>0>7-(3=,).'

Gemma 7B -it LoRA finetuned

Summary of Findings and Implications

This research describes the functional switching valve within the female genitalia of the Brazilian cave insect Neotrogla. The valve complex is composed of two plate-like sclerites, a closure element, and in-and-outflow canals. Females have an intromittent organ to coercisively anchor males and obtain voluminous semen. The semen is packed in a capsule, whose formation is initiated by seminal injection. The valve complex has two slots for insemination, allowing Neotrogla to continue mating while the first slot is occupied.

Key Findings:

* The switching valve mechanism is located at the spermathecal plate.
* The valve allows females to control the direction of seminal flow.
* The presence of the spermathecal plate with a valve function to control seminal flow may further refine our understanding of the causes and consequences of the reversed sexual selection in this genus.
* The switching valve system reported here likely represents an adaptation for direct benefits rather than for genetic benefits.

Implications:

* The evolution of the switching valve may have been a prerequisite for the reversal of the intromittent organ in Neotrogla.
* The switching valve allows females to obtain more nutrition from a male.
* The switching valve could be a valuable tool for understanding the evolution of female reproductive organs and the mechanisms underlying sex-role reversal.

Overall, this research provides a detailed description of the functional switching valve within the female genitalia of Neotrogla, highlighting its significance in the evolution of the genus and its potential for further scientific investigation.

Inconsistent generations

- Gemma 7B generated summaries for 100 out of 241 articles.
  - Memory constraints
- Gemma 7B: useful generations were 49 out of 100 summaries.
- T5 generated 10 summaries (so far)

Evaluation

Evaluation is done for three criteria: relevance, readability, and factuality.

- Relevance:
  - ROUGE1, ROUGE2, ROUGE-L, BERTScore

- Readability:
  - Flesch-Kincaid Grade Level (FKGL), Dale-Chall Readability Score (DCRS), Coleman-Liau Index (CLI), Linsear Write Readability Formula (LENS)

- Factuality:
  - AlignScore, SummaC

Results: Readability

Model	FKGL	DCRS	CLI	LENS
Long-T5	15.35	**8.795**	**16.57**	**17.91**
Long-T5 LoRa-Finetuned	**13.68**	11.675	16.85	13.99
Gemma 7B LoRa-Finetuned	31.965	19.024	30.425	3.470
Gemma 7B inference				

Results: Relevance

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Long-T5	0.315	0.044	0.304	0.814
Long-T5 LoRa-Finetuned	**0.366**	**0.087**	**0.336**	**0.836**
Gemma 7B LoRa-Finetuned	0.103	0.005	0.09	0.774

Results: Factuality

Model	AlignScore	SummaC
Long-T5	0.383	0.508
Long-T5 LoRa-Finetuned	**0.966**	**0.921**
Gemma 7B LoRa-Finetuned	0.393	0.470

Conclusion

- On the 10 documents Long-T5 were validated on LoRa fine-tuning outperformed the out-of-the-box model for both factuality and relevance metrics
- Gemma7B for LoRa fine-tuning had good factuality scores, but needs improvement on readability and relevance.

9. Noah, Annanya, Jayden, Xiaoyan

INTRODUCTION

- Style Transfer involves altering the style or tone of a piece of text while preserving its underlying meaning or content. This can include changing a text’s sentiment, formality, or writing style.
- For example, converting a formal text into a more casual or conversational style, or changing the sentiment of a sentence from positive to negative.
- Machine Translation refers to the automatic translation of text from one language to another. This task involves converting text in a source language to text in a target language while preserving its meaning

GOAL FOR THE PROJECT

- Take an input sentence in English, convert to Japanese with a particular style
- Pipeline: translate and remove style in one step, then re-apply style in a second step

This is great, where is it from?

どこで手に入れた？(where did you get it?)

これはおいしいよ。どこで手に入れたの？(This is delicious. Where did you get it?)

これはまずいよ。どこで手に入れた？(this is awful. Where did you get it?)

MOTIVATION

- Maintain nuance in translations
- Screen content, e.g. for kids
- Balance model performances across styles - gender, dialects, etc.

DATA

- Japanese: chABSA dataset,
  - consists of business interactions.
  - labeled with positive, negative, and neutral sentiments
  - Used in reapplication of style step
- English: Yelp Reviews
  - Used as source of English sentences with sentiment in fine-tuning step

METHODOLOGY

- Utilizes back translation to remove style during the translation process.
  - Translate E->J->E to remove style, following Prabhumoye et al. 2018.
  - Use E->J->E->J as baseline

- Fine-tunes the model to translate from English to Japanese in a single step, aiming to remove style.
  - Use a reinforcement learning framework
  - KL Div not ideal as regularizer - not a direct measure of semantics

$$Loss = -Classifier\_BCE + \lambda \times KL\_Divergence$$

- Investigates sentiment as the style due to data availability
- Utilizes a cross-aligned autoencoder model for reapplying style
  - Shen et al. 2017
  - Approach applicable to any style dataset, only used in reapplication step
  - Applicable to non-parallel data

RESULTS

- Reinforcement learning step
  - HuggingF***???
- Cross aligned auto-encoder
  - Python >= 2.7
    - Also apparently Python <3.0
  - I don't have a GPU
  - I'm not rewriting this thing
  - C from 200 years ago still compiles, Python from 7 years ago breaks
  - 2 hours of my life I'll never get back

Dependencies

Python >= 2.7, TensorFlow 1.3.0

```
    print 'Loading model from', args.model
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
```

RESULTS

- Back-back-back-translation performance on sentiment

```
[8]: mse_loss(torch.tensor(translation_one_scores), torch.tensor(all_pointfive))

[8]: tensor(0.2134)

[9]: mse_loss(torch.tensor(translation_two_scores), torch.tensor(all_pointfive))

[9]: tensor(0.1925)

[10]: mean_difference = np.mean([x-y for x, y in zip(translation_one_scores, translation_two_scores)])

[11]: mean_difference

[11]: 0.036276199660458405
```

RESULTS

- Baseline model
  - Pretty bad, but very funny
  - Nearly all sentences lose a great deal of semantic meaning
- 92 sentences evaluated for readability on scale of 0-4
  - 0 - total failure
  - 1 - completely incoherent
  - 2 - some semantic meaning retained
  - 3 - some semantic meaning lost
  - 4 - good translation
  - Average score: 2.47

RESULTS

A very good translation:

ちょっと似顔絵のある場所ですが,その方がよいでしょう. '
'たいていのスタッフは郵便配達員の考え方に適合しています .殆どのスタッフは快活です.時折直線に到達できるので,船舶に問題はありません.とても協調的で精密で,すべてのパッケージを早く送ろうとしません.',
(I'm not trying to deliver the packages quickly)

'Kinda sketchy location, but getting better. \n'
'Most of the staff is consistent with the idea of a postal '
'worker. \n'
'Pleasant for the most part. Never had any issues with '
'shipping. \n'
'Workers are very helpful and thorough... which sometimes '
'leads to a bit of a line. No big deal as I want all '
'packages to be delivered promptly.'

RESULTS

- A more typical translation:

  '私は料理が好きで メニューは小さいし 飲み物もいい',
  (I like the food and the menu is small, the drinks are also good)

   'what can i say, small, simple , quaint. tasteful. i like '
   'the food. the menu is small and the drinks are awesome. '
   'dont plan on pigging out here. its made for your '
   'tastebuds, not ur belly. eat well, drink better. good '
   'place to sit and eat small before a night out. parking is '
   'a bitc*. steak is the main course here... wow.',

RESULTS

- Funny

'(笑)'
(lol)

'Amazing! Great selection of delicious appetizers! '
'Well-priced SAKE selections, and of course ! A wide range '
'of beautifully cooked UDON. This is the absolute Japanese '
'Izakaya experience. Cannot wait to go back!!',

RESULTS

- Funny

'* 私 は スペイン 語 * の おいしい パスタ・サンドイッチ を 持っ て い ました。',
(The delicious pasta of *I am the Spanish Language* ・I held the sandwich).

'*surprisingly good for in a sleeper way\n'
'*I had their savory pastry sandwich and it was superb\n'
"*hipster? you bet but where do you think you are? that's "
'right Rip City!',

RESULTS

- Total disasters:
- "よしよ
しよ
しよ
しよ
しよ
しよ
しよし!',

  'Great tour! Not only was it a fun Halloween activity, it '
  'was a great way to see Harvard. Our guide, Nathaniel, '
'really made the tour - great storyteller, fun sense of '
'humor, and smart! (What a vocabulary!) Nice mix of history '
'and entertainment made for an engaging 90 minutes for my '
'sister and I as well as our three 20-something kids.'

RESULTS

- Total disasters:

'いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,い
い,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,い
い,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい,いい.',

"Great food. Vegetarian friendly. Amazing coffee and '
'deserts. Its pretty small. But there is a patio in front '
'with heat lamps',

RESULTS

- Total disasters:

'!!
!!
!!
!!
!!',

'I LOVE, nay, am OBESSED with THEIR FAKE LASHES!!!!!!',

RESULTS

- Total disasters:
-

'(2018年3月25日). '
'https://www.oricon.co.jp/email/article/news/2017/ "2018年3月20日閲覧. ^ [リンク] [リンク] [] [] [リンク] [] [] []] '
'[リンク] []]] [リンク] []]]] ^ a b c d e f g h i ^ a b c d e '
'f g h i e f g h i h i f g h i f g h i h i f g h i h i '
'f g h i h i "s s.',

'The food is excelleeeeeeent \n'
'The most delicious veggi butter chicken ever\n'

RESULTS

Update while other groups were going: RL model works

娘と私は毎週金曜の朝を終えました。正直、一週間のハイライトです。ドントンホールは死にます。しかし、ここでのすべては良好です。他の評論家の言うように、彼らは速いから早くここに来てください!
(The donton holes are going to die).

[
3:56 PM
]

original: HOLY MOTHER OF DONUTS! This place is amazing. My daughter and I stop every Friday morning, and honestly, it's the highlight of my whole week. Donut holes are to die for, but everything here is good. Like other reviewers said, get here early because they go fast!