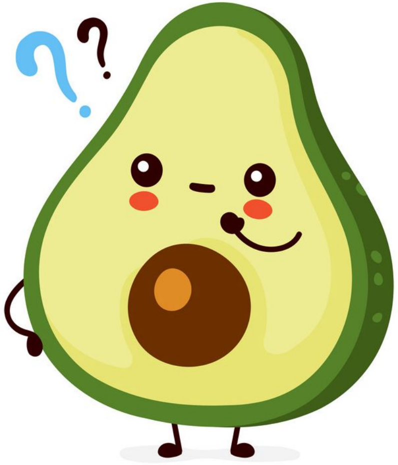
Medical Text Simplification

Anuja Tayal
Baris Karacan

Question

- What percentage of adults in the U.S have proficient health literacy?



12%

Introduction

- Patients have **more access** to their **health information** to support self-care.
- **Readability** measures of **health information** is **significantly higher** than patient **health literacy** abilities.
- Most of the **health documents** are **jargon-heavy** and contains **complex words** that are challenging to comprehend for **lay people**.
- **Rehospitalization** rate is **very high** for patients who suffer critical and chronic conditions such as **heart failure**.

Problem Statement

- Simplify Complex Terms in Medical Text to make the text more comprehensible to patients
- By **simplification** of complex terms in medical documents, it could be possible to:
    - Increase self-confidence and motivation of patients.
    - Enhance patient engagement with the treatment
    - Reduce readmission rate.

Simplification

- Task of reducing a complex document into its simpler version
- Retain important, meaningful content
- Fluent,continuous
- size of initial text- not necessarily changed

Types of Text Simplification

Replacement

- Identify Complex Terms
- Replace complex Terms with simpler meaningful understandable word.

Generation

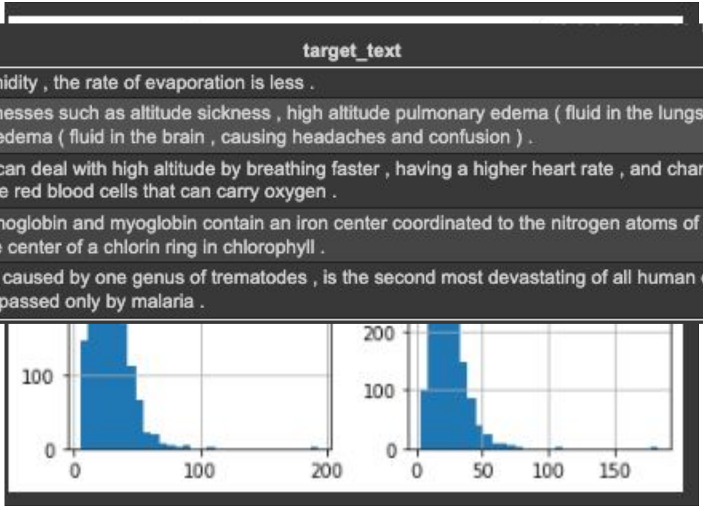
- Generating a new simplified sentence

Dataset

- **Medical EW-SEW** dataset is utilized which contains **2267** sentence pairs of **ordinary** and **simple English Wikipedia** sentence pairs.
- Sentences are filtered by **QuickUMLS**(a named entity recognition tool) to include at least one medical entity mention of type ***Disease or Syndrome*** and ***Clinical Drug***

Dataset

source_text	target_text
under conditions of high humidity , the rate of evaporation of sweat from the skin decreases .	with a higher humidity , the rate of evaporation is less .
the lack of oxygen above 2,400 metres (8,000 ft) can cause serious illnesses such as altitude sickness , high altitude pulmonary edema , and high altitude cerebral edema .	this can cause illnesses such as altitude sickness , high altitude pulmonary edema (fluid in the lungs) , and high altitude cerebral edema (fluid in the brain , causing headaches and confusion) .
the human body can adapt to high altitude by breathing faster , having a higher heart rate , and adjusting its blood chemistry .	the human body can deal with high altitude by breathing faster , having a higher heart rate , and changing the bloo itself to have more red blood cells that can carry oxygen .
for example , hemoglobin and myoglobin contain an iron center coordinated to the nitrogen atoms of a porphyrin ring ; magnesium is the center of a chlorin ring in chlorophyll .	for example , hemoglobin and myoglobin contain an iron center coordinated to the nitrogen atoms of a porphyrin r magnesium is the center of a chlorin ring in chlorophyll .
schistosomiasis , caused by one genus of trematodes , is the second-most devastating of all human diseases caused by parasites , surpassed only by malaria .	schistosomiasis , caused by one genus of trematodes , is the second most devastating of all human diseases cau by parasites , surpassed only by malaria .



Readability Measures

- TextStat Python Library
- Determine readability, complexity and grade level
  - Automated Readability Index(ARI)
  - Flesch Kincaid Grade
  - Smog Index
- Value of 9 means 9th grader can understand the text
- Metric Relies on shallow cues, length of words, sentences, documents

Readability Measures

Measure	Source	Target
ARI	10-17	5-17
Flesch Kincaid Score	10-17	8-17

Replacement

- **Ordinary** subset is used for **generating** candidates.
- **Simple** subset is used for **evaluating** the results.
- Replacement technique consists of three steps:
  - Identification of **complex** words
  - Generating candidates using **Masked Language Modeling (MLM)**
  - Retrieving the best candidate using **Sentence Transformers**

Identification of Complex Words

- **MetaMap** is utilized to extract **medical concepts** from each sentence in the ordinary dataset.
- All the medical concepts are stored in a text file.

```
# initialize metamap
mm_home = '/Users/bariskaracan/Downloads/public_mm/bin/metamap16'
mm = MetaMap.get_instance(mm_home)
```

```
concept:  ConceptMMI(index='-e 1', mm='MMI', score='14.64', preferred_name='Myocardial Infarction'
```

Identification of Complex Words

- **Zipf frequency** values of medical concepts are calculated.
- **Zipf frequency** score of a word is the base 10-logarithm of the number of times it appears per billion words.
- **Medical concepts** lower than the threshold **4** are identified as **complex words.**

```
from wordfreq import zipf_frequency
zipf_frequency('stop', 'en')

```

```
5.49
```

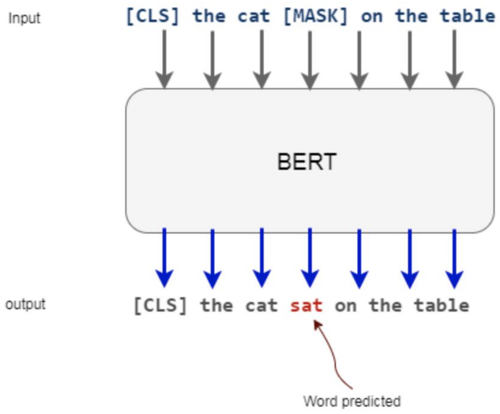
```
from wordfreq import zipf_frequency
zipf_frequency('thwart', 'en')

```

```
3.06
```

Generating candidates using MLM

- Implemented **BERT** which is optimized by **MLM** task in which **BERT** predicts the **missing tokens** in a sequence given its **left** and **right** context.
- For each complex word **w** in a sentence **S**, we mask the word **w** in **S** using special symbol "**[MASK]**".



Generating candidates using MLM

- From generated candidates, **non-alpha numericals**, **stopwords** and words that have **higher zipf score** than the masked word are **removed**.
- MLM of pre-trained **BERT-base**, **BioClinicalBERT** and **PubMedBERT** are implemented via **HuggingFace** and **compared** to each other.
- **BioClinicalBERT** was trained on all notes from **MIMIC 3**.
- **PubMedBERT** is pre-trained from scratch using **abstracts** from **PubMed** and **full-text articles** from **PubMedCentral**

```
words to replace:  pulmonary
candidate words:  ['pulmonary', 'cerebral', 'lung', 'respiratory', 'lungs', 'cardiac', 'muscular', 'systemic', 'peripheral', 'vascular', 'pedal',
```

```
words to replace:  edema
candidate words:  ['edema', 'congestion', 'disease', 'swelling', 'infection', 'symptoms', 'illness', 'syndrome', 'issues', 'irritation', 'pressures',
```

Retrieving the best candidate

- **SentenceTransformer** is used to capture **similarities** between the list of candidate words and the source complex word.
- From **HuggingFace**, **SentenceTransformer** model "***all-mpnet-base-v2***" is used to generate **embeddings** from list of candidates.
- "***all-mpnet-base-v2***" is trained on a large and diverse dataset over **1 billion** training pairs and provides the best quality among **HuggingFace SentenceTransformers**
- **Cosine similarity** function is implemented to **score** the **similarity** of generated **embeddings**.
- The word with **highest** score is **swapped** with original complex word.

```
words to replace:  pulmonary          (0.81462264, 'lung')

words to replace:  edema              (0.52083147, 'swelling')
```

Evaluation

- For each **model**, resulting sentences with updated complex words are stored in separate **documents**(text files).
- Each **document** is compared to **simple dataset** by computing **ROUGE** score.
- **ROUGE** is a set of metrics that compares **automatically produced** documents against a set of **reference** documents.
- From **ROUGE**, **ROUGE-1**(overlap of **unigrams** between the **automated** and **reference** documents) and **ROUGE-L**(measures **longest matching sequence** of words.) scores

Evaluation

Model	Rouge-1 Precision	Rouge-1 Recall	Rouge-1 F-measure	Rouge-2 Precision	Rouge-2 Recall	Rouge-2 F-measure	Rouge-L Precision	Rouge-L Recall	Rouge-L F-measure
BERT-base	64.06	74.51	66.31	50.71	58.93	52.45	61.65	71.53	63.80
BioClinicalBERT	64.00	74.42	66.24	50.63	58.83	52.37	61.60	71.46	63.75
PubMedBERT	**64.11**	**74.58**	**66.37**	**50.78**	**59.04**	**52.54**	**61.69**	**71.59**	**63.85**

Generation

- Replicated the paper “Paragraph-level Simplification of Medical Texts”
- New Masked Language Model based Measure to score readability/technicality
- Analysing and Understanding style of words used in complex(source) and simple(target) text
- Cochrane Dataset- available in HuggingFace

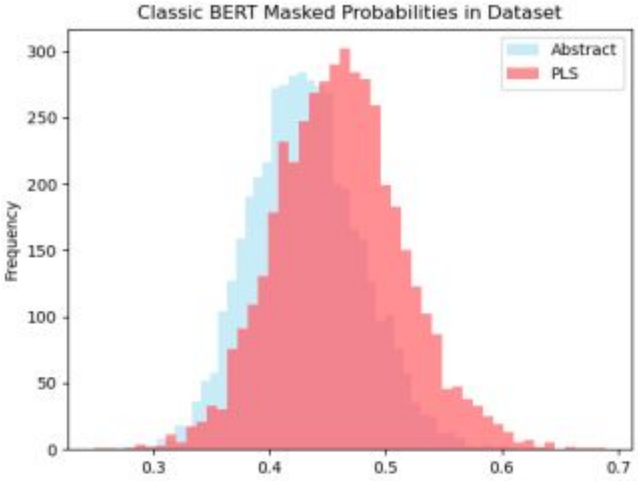
[Paragraph-level Simplification of Medical Texts](https://aclanthology.org/2021.naacl-main.395) (Devaraj et al., NAACL 2021)

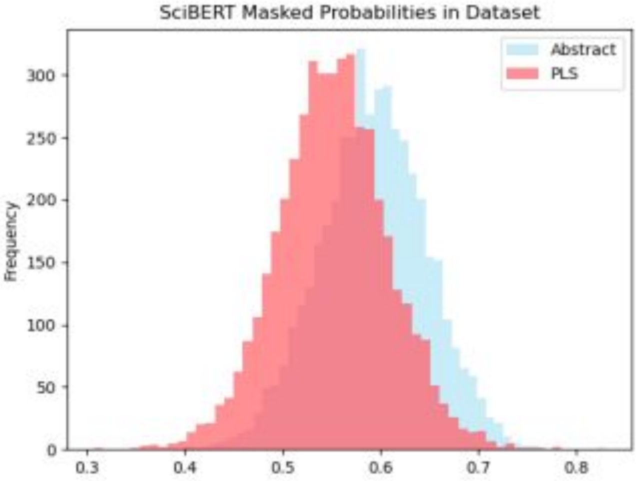
Bert vs Scibert Masked Language Model for Readablity

- Adopt Bert and Scibert MLM to measure readability and technicality
- Based on the notion that as Bert is trained on lay language corpus, it generates or gives more preference to simple words as compared to scibert
- Bert base uncased
- allenai/scibert_scivocab_uncased

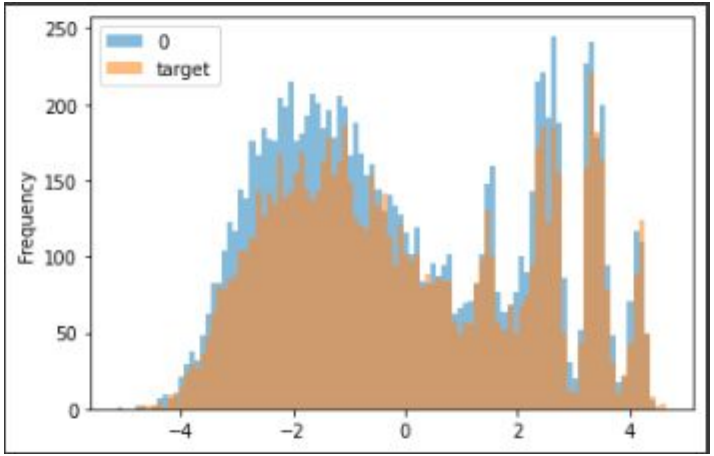
procedure MASKED-PROB(D, M)
    sents $\leftarrow$ SENTENCE-SPLIT(D)
    $P \leftarrow$ Initialize empty list
    for $i = 1 \ldots |\text{sents}|$ **do**
        $T \leftarrow$ TOKENIZE(sents[i])
        for $j = 1 \ldots 10$ **do**
            $A \leftarrow$ sample 15% from $1 \ldots |T|$
            $T' \leftarrow T$
            for all $a \in A$ **do**
                $T'[a] \leftarrow$ [MASK]
            outputs $\leftarrow$ FORWARD(M, T')
            for all $a \in A$ **do**
                prob $\leftarrow$ outputs[a][$T[a]$]
                APPEND(P, prob)
    return mean(P)

Expected Results

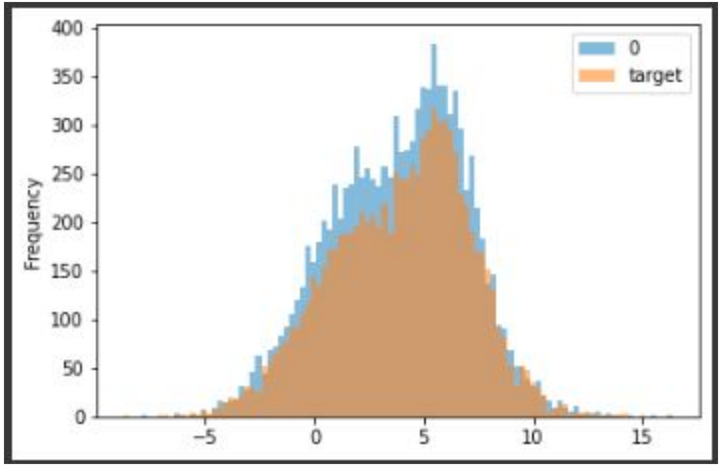




Bert Masked Language Model for Readibility



Bert MLM



Scibert MLM

Analyse Style of Words

- Trained Logistic Regression
- Classify Text whether it is complex or simple
- Weights learned is used to train custom loss function
- As training on Bart xsum dataset, represented text as bag of words frequency vector

Logistic Regression

- complex=0,simple=1
- Training = 2000
- Test 500
- Accuracy 71%

contracted	-0.5447289454
vision	-0.5383142094
anterior	-0.5379903169
following	-0.5365570684
several	-0.5321599392
e	-0.5305458822
described	-0.5279466268

typically	-0.9308641429
acute	-0.8623421949
17	-0.8537286316
infection	-0.822810355
in	-0.8010535807
include	-0.7969276697
been	-0.7713534821
known	-0.7542755013
as	-0.7174090741
affect	-0.7140461613
commonly	-0.6963812491
multiple	-0.686517685
cold	-0.6753312985
ated	-0.6660012127
risk	-0.6532455074
days	-0.6496920021
medical	-0.6455146047
ac	-0.6362494915
produce	-0.6331108215
b	-0.6330596658
;	-0.6249150579
,	-0.6237995398
treatment	-0.6222961221

time	0.8269188124
when	0.8270022262
some	0.8982492616
mental	0.9047473263
get	0.9285405925
have	0.9414086494
because	0.9707926811
person	0.9947926114
0	1
it	1.002098803
like	1.002864237
illness	1.068159053
this	1.079760657
they	1.109869653
people	1.152932066
said	1.602830924
called	1.612696342
.	2.776590693

Unlikehood Training

- Maximum Likelihood Training
- Explicitly penalise the model for producing seemingly technical words
- Add a term

```
class CustomTrainer(Seq2SeqTrainer):
  def __init__(self,*args,**kwargs):
    super().__init__(*args,**kwargs)
  def compute_loss(self,model,inputs,return_outputs=False):
    labels=inputs.get("labels")
    outputs=model(**inputs)
    logits=outputs.get("logits")
    loss1=unlikelihood_loss(logits,labels)
    return (loss1,outputs) if return_outputs else loss1
```

@inproceedings{ Welleck2020Neural, title={Neural Text Generation With Unlikelihood Training}, author={Sean Welleck and Ilia Kulikov and Stephen Roller and Emily Dinan and Kyunghyun Cho and Jason Weston}, booktitle={International Conference on Learning Representations}, year={2020},}

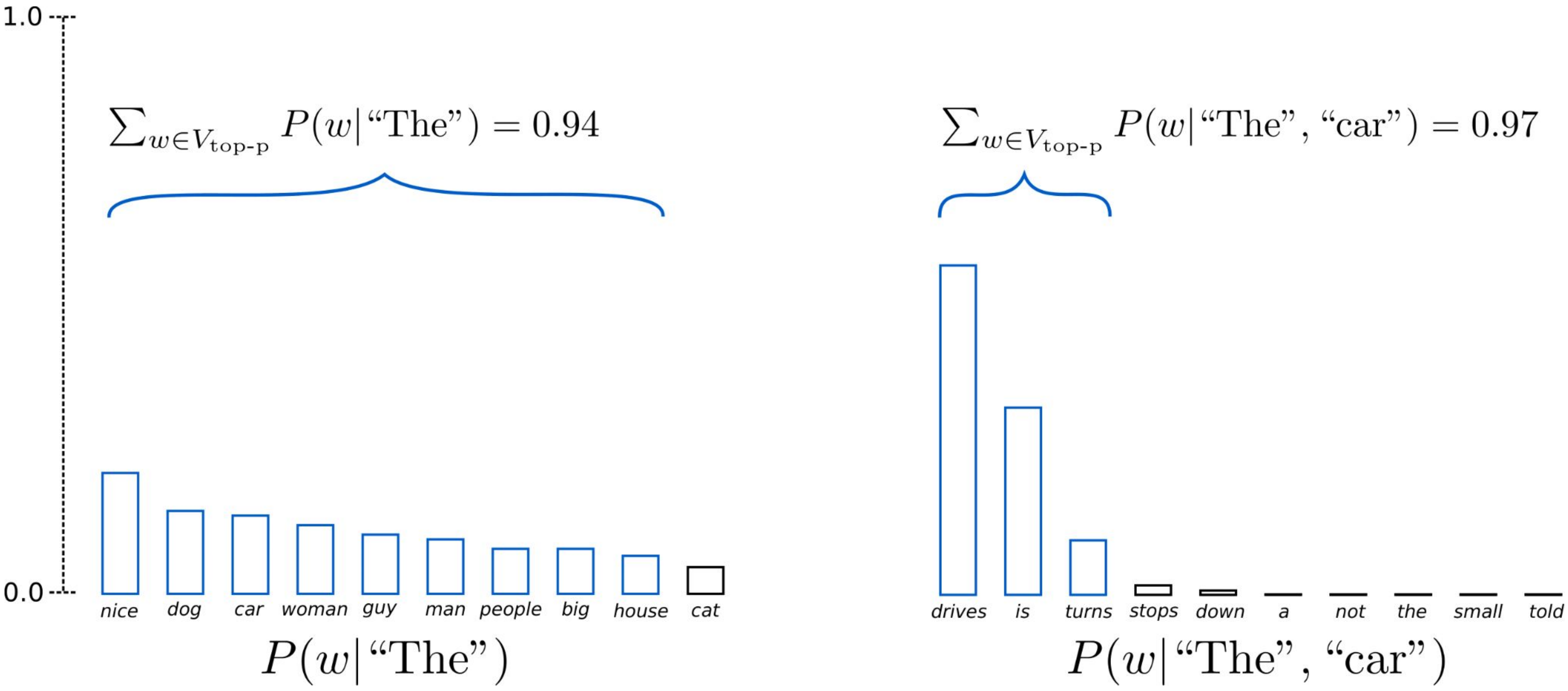
```python
def unlikelihood_loss(logits,labels):
    probs=F.softmax(logits,dim=-1)
    neg_probs=1-probs
    neg_probs+=(neg_probs==0).float()*1e-8
    log_neg_probs=torch.log(neg_probs)
    attention_mask=labels.eq(1).eq(0).float()
    attention_mask = attention_mask.unsqueeze(2).expand(-1,-1,logits.shape[2])
    log_neg_probs_masked=log_neg_probs*attention_mask
    N,s=logits.size()[:2]
    weight_mask_expanded=weight_mask.unsqueeze(0).unsqueeze(0).expand(N,s,-1)
    weighted_probs=log_neg_probs_masked*weight_mask_expanded
    return(-torch.sum(weighted_probs))
```

$$UL = \sum_{j=1}^{|S|} -\log(1 - p_\theta(s_j | y_{<t}, x)),$$

Generating Strategies

- Greedy Search
- Beam Search- num of beams,choosing the output with highest prob
- Sampling- randomly picking the next word according to conditional probability distribution
  - Top k Sampling- K most likely next words are filtered and then probability mass is distributed.
  - Top-P Nucleus Sampling- at each step the next token is generated whose cumulative probability exceeds the prob p.

Nucleus Sampling



Hyperparameters

- Customise Seq2SeqTrainer for unlikelihood loss
- Bart for Conditional Generation
- Weights- logistic Regression Weights
- top-p=0.9
- temperature=1.0
- batch-size=1
- Learning rate=3e-5
-

Model Comparison

- For Evaluating Simplifying Techniques
- Recall Oriented Understudy for Gisting Evaluation (ROUGE)
- Metric for Evaluation of Translation with Explicit ORdering (METEOR)

Results

Model	Training Loss	Validation Loss	Rouge1	Rouge2	RougeL	RougeLsum	Meteor
Simple	0.4726	0.581	67.42	54.57	64.52	64.48	0.673
UL	-293.77	37.5	3.75	0	3.75	3.75	0.014

Source Text			
Under conditions of high humidity the rate of evaporation of sweat from the skin decreases	If the number of hours is raised in high humidity, this reduces the rate of evaporation of sweat from skin	During this time, the rate of evaporation of sweat from the skin lessens	If conditions of high humidity, there is higher rate of evaporation from the skin
the lack of oxygen above 2,400 metres (8,000 ft) can cause serious illnesses such as altitude sickness , high altitude pulmonary edema , and high altitude cerebral edema .	'this can cause symptoms such as altitude sickness , high altitude pulmonary edema , and high altitude cerebral edema .',	'this has some serious diseases such as altitude sickness , high altitude pulmonary edema , and high altitude cerebral edema .']	

Future Work

- Other metrics could be used besides zipf frequency for filtering candidates.
- Instead of word-level masking, concept-level masking could be implemented to have better perception over complex terms (e.x pulmonary edema, myocardial infarction, etc.)
- Come up with a better representation to map meaning of complex term to simple term
- Change Logistic Regression to something context dependent model
- Train with different hyperparameters

Conclusion

- Replicating the paper is tough
- Paper is highly data dependent
- Hard to find appropriate dataset
- Setting different Hyperparameters can be done in generating step, not in modeling step

THANK YOU
Questions?