

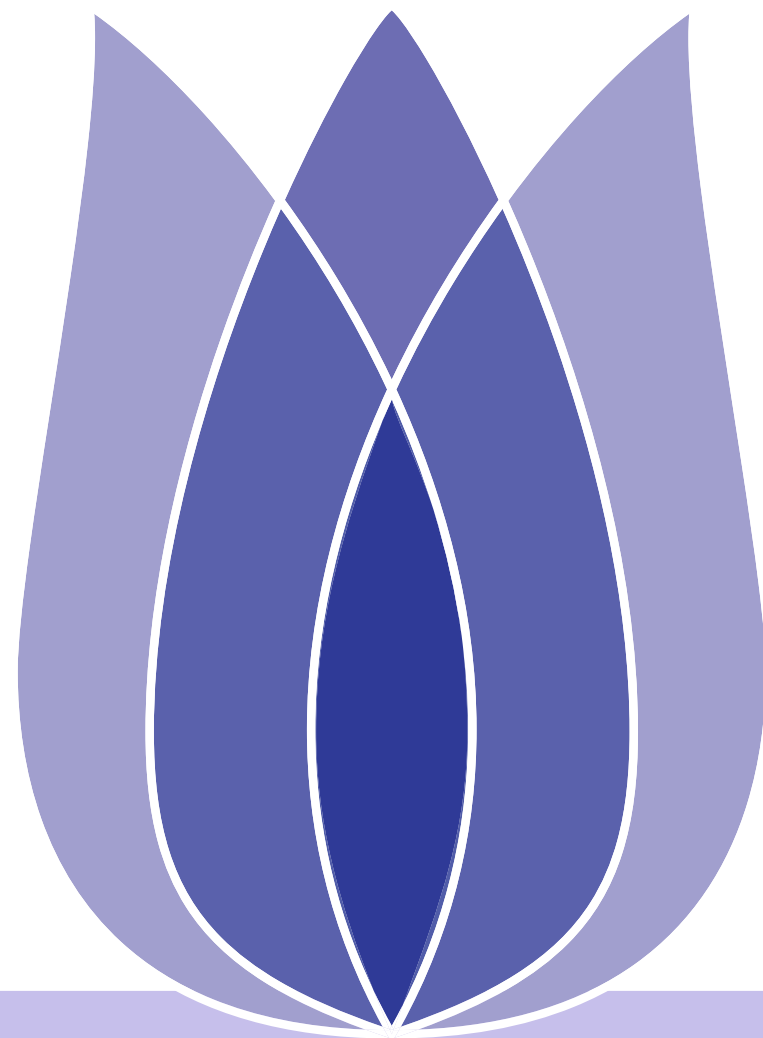


# Jigsaw-Unintended-Bias-in-Toxicity-Classification-solution

Pengcheng Jiang

JiLin University

2021-07-24





# Overview

[Problem Definition](#)

[Text preprocessing](#)

[Embedding](#)

[moudle](#)

## Problem Definition

## Text preprocessing

## Embedding

## moudle



Problem Definition

Text preprocessing

Embedding

moudle

# Problem Definition



# Jigsaw-Unintended-Bias-in-Toxicity-Classification-solution

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

given	A tagged dataset containing comments. Target 0 for malicious comments and 1 for friendly comments.
target	detect toxic comments and minimize unintended model bias.
evaluate	ACC.



# Train

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

Data

id	target	comment_text
59848	0.000000	This is so cool. It's like, 'would you want yo...
59849	0.000000	Thank you!! This would make my life a lot less...
59852	0.000000	This is such an urgent design problem; kudos t...

analysis

```
a sentence has max words: 1903  
a sentence has min words: 1  
a sentence has average words: 297  
there are total 289197 unique words
```



[Problem Definition](#)

[Text preprocessing](#)

[Embedding](#)

[moudle](#)

# Text preprocessing



# Text preprocessing

[Problem Definition](#)

[Text preprocessing](#)

[Embedding](#)

[moudle](#)

- Count the total number of words contained in all texts, the maximum and minimum number of words contained in a text
- Check for missing data
- Change abbreviations to full:isn't -> is not(via dictionnary)
- clean\_numbers
- Find all non alphabetic characters and clean\_special\_chars
- Solve the problem of misspelling words
- lower



**TULIP**

*Team for Universal Learning and Intelligent Processing*





# Text preprocessing

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

comment_text	comment_text
This is so cool. It's like, 'would you want yo...	this is so cool it is like would you want y...
Thank you!! This would make my life a lot less...	thank you this would make my life a lot less...
This is such an urgent design problem; kudos t...	this is such an urgent design problem kudos t...
Is this something I'll be able to install on m...	is this something I will be able to install on...
haha you guys are a bunch of losers.	haha you guys are a bunch of losers



- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)**
- [moudle](#)

# Embedding



# Tokenizer

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

concept	What tokenizer does is actually very simple. It divides the words it sees into spaces, and then uses numbers to correspond one by one. Then we take the first num_ Words is the word with the highest frequency, others are not recognized.
Instruction	First learn the dictionary of the text, and then get the corresponding relationship between words and numbers, and then convert the text into a number string through this relationship, and then use the padding method to make up the number string to the same degree, then you can proceed to the next step : embedding
Method	collections.counter,pytorch:torchtext.vocab
result	The text is transformed into the corresponding number sequence, and the text that does not meet the length is filled with 0

# Text preprocessing

[Problem Definition](#)

[Text preprocessing](#)

[Embedding](#)

[moudle](#)

```
1 somestr = ['ha ha gua angry', 'howa ha gua excited naive']
2 tok = tt.Tokenizer()
3 tok.fit_on_texts(somestr)
4 tok.word_index
5 Out[90]: {'angry': 3, 'excited': 5, 'gua': 2, 'ha': 1, 'howa': 4, 'naive': 6}
6 tok.texts_to_sequences(somestr)
7 Out[91]: [[1, 1, 2, 3], [4, 1, 2, 5, 6]]
```



**TULIP**

*Team for Universal Learning and Intelligent Processing*



# Embedding

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

Instruction	The embedding layer is the same as word2vec. Whether it is skip gram or cbow model, they infer each other from the context and the current, so we consider the relationship between the preceding and the following.
Method	glove.42B.300d.txt



Problem Definition

Text preprocessing

Embedding

moudle

moudle





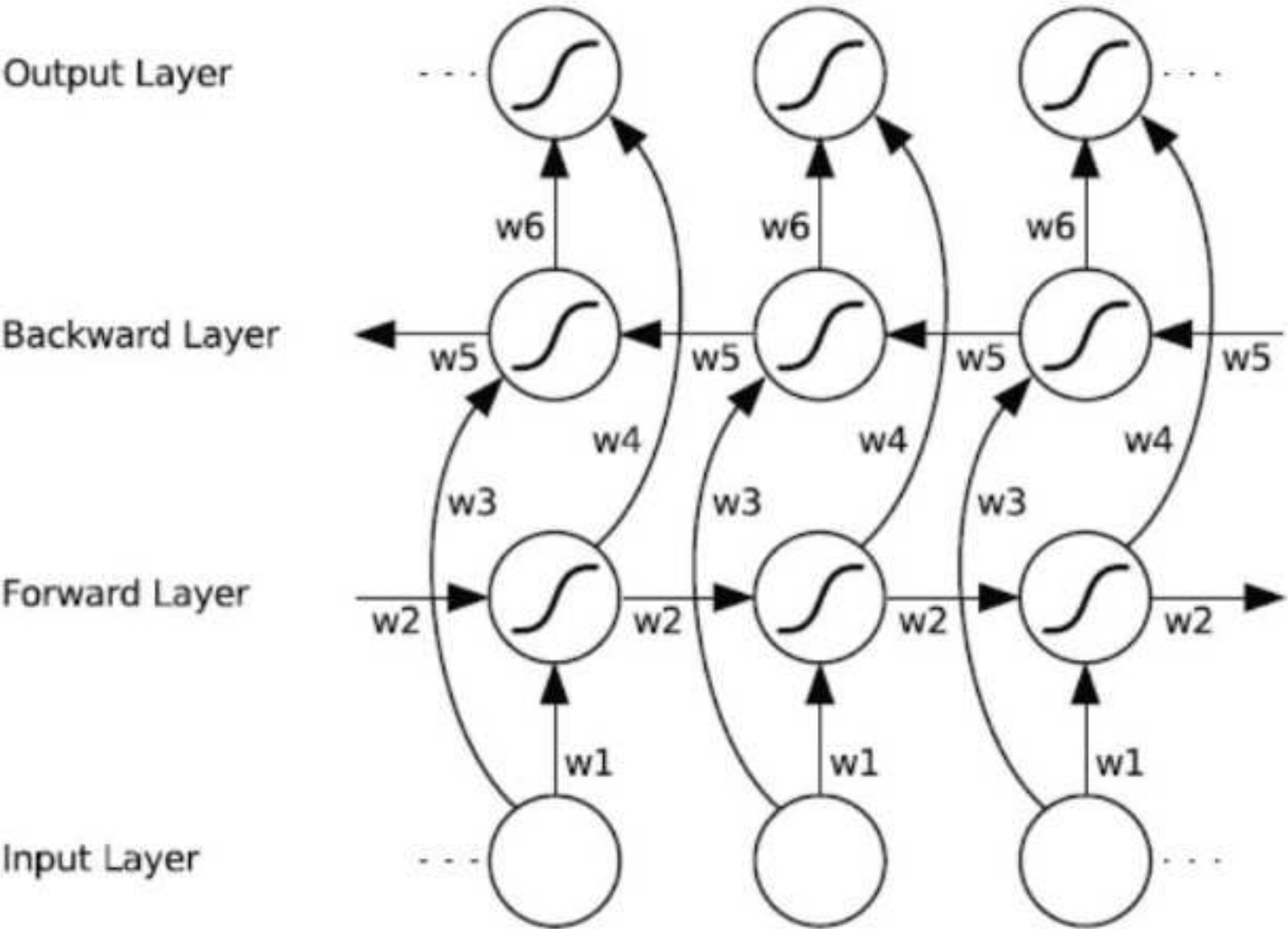
- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)

Instruction	BiRNN:In practical problems, there are also problems that not only rely on the previous sequence, but also rely on the subsequent sequence for prediction. For those problems, we need to use bidirectional RNN (birnn)
setting	embed_size, num_hidden, num_layers = 300, 100, 2



# Birnn

- [Problem Definition](#)
- [Text preprocessing](#)
- [Embedding](#)
- [moudle](#)







# Result

Problem Definition

Text preprocessing

Embedding

moudle

acc: 0.9468