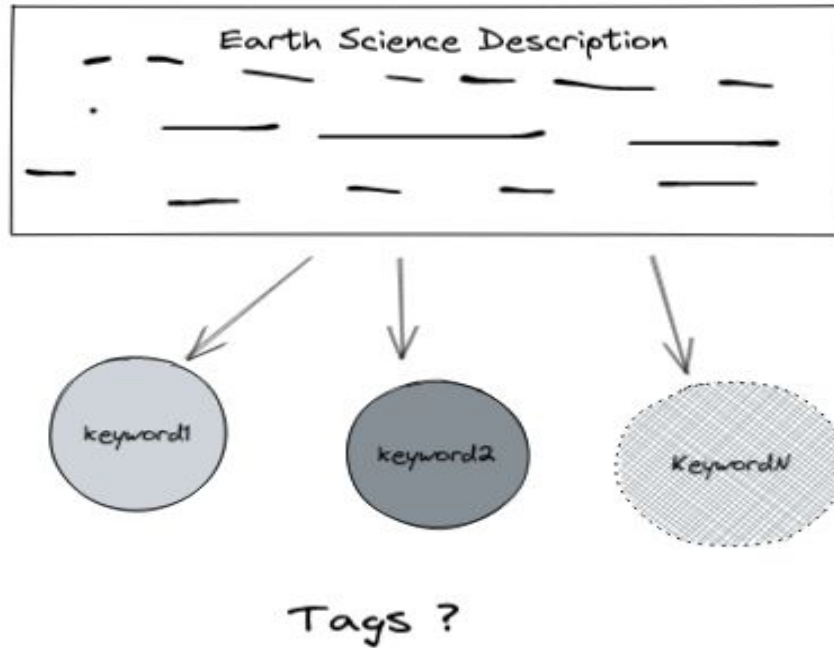




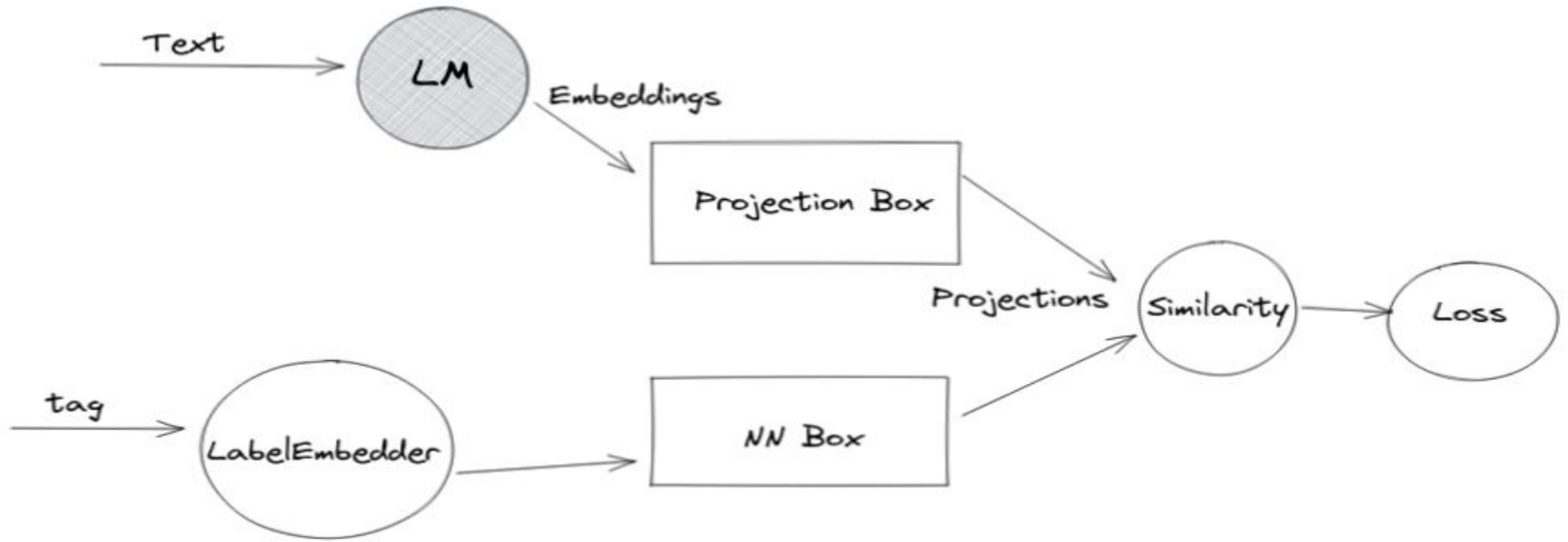
# Earth Science

Where Nish meets  
NLP meets UAH

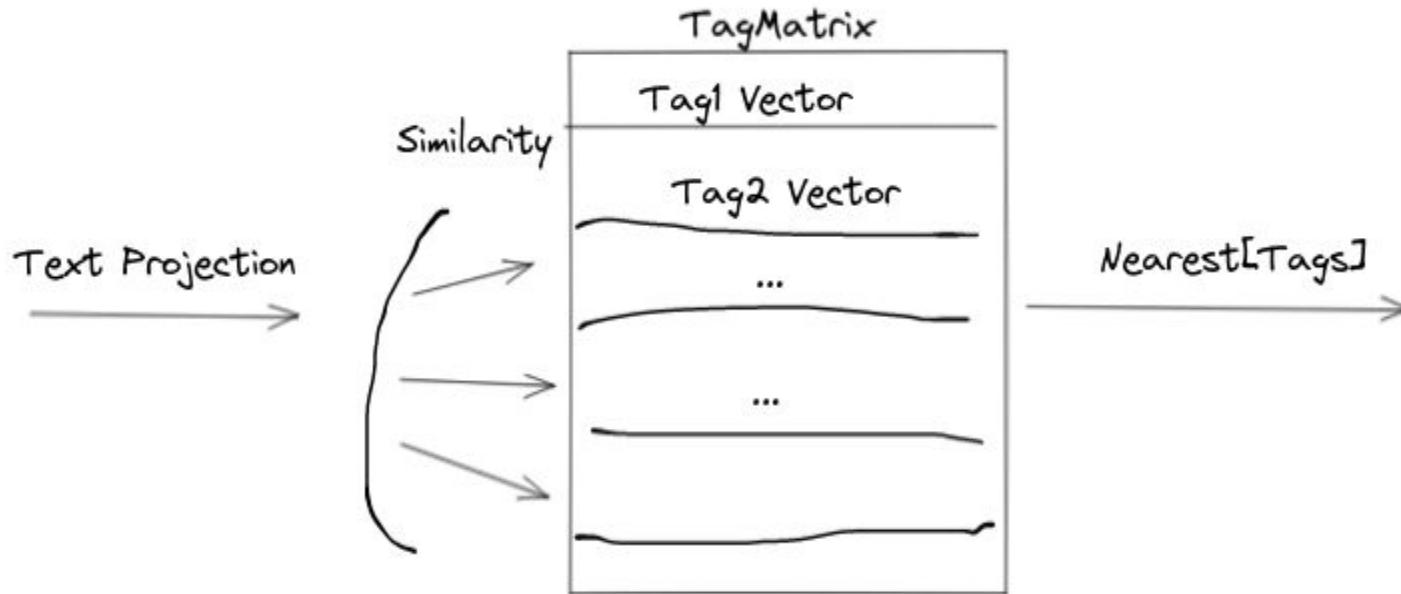
# Problem Statement



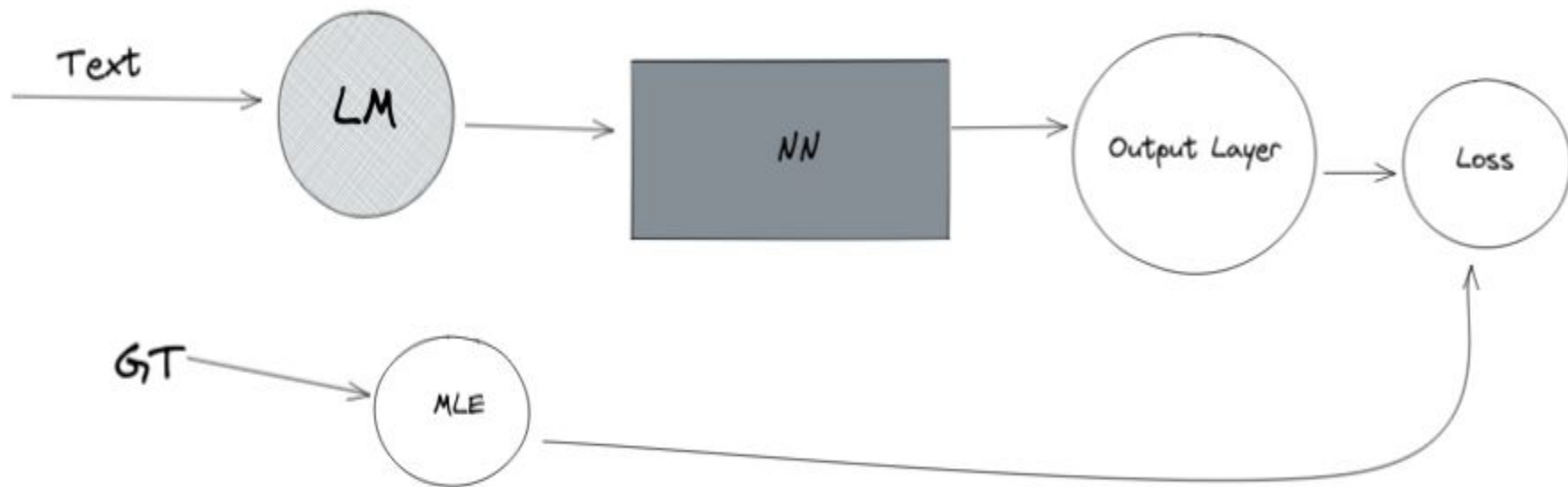
# Approach 1



# Approach 1 (Inference)

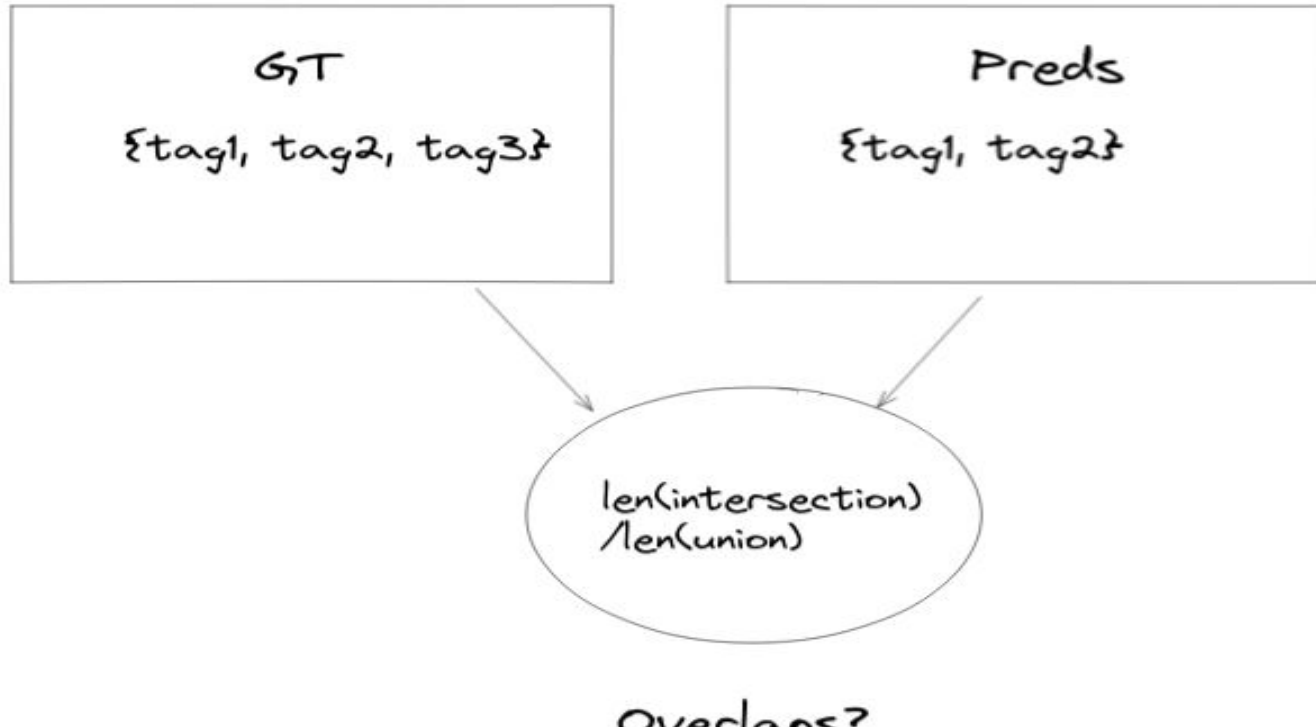


## Approach 2

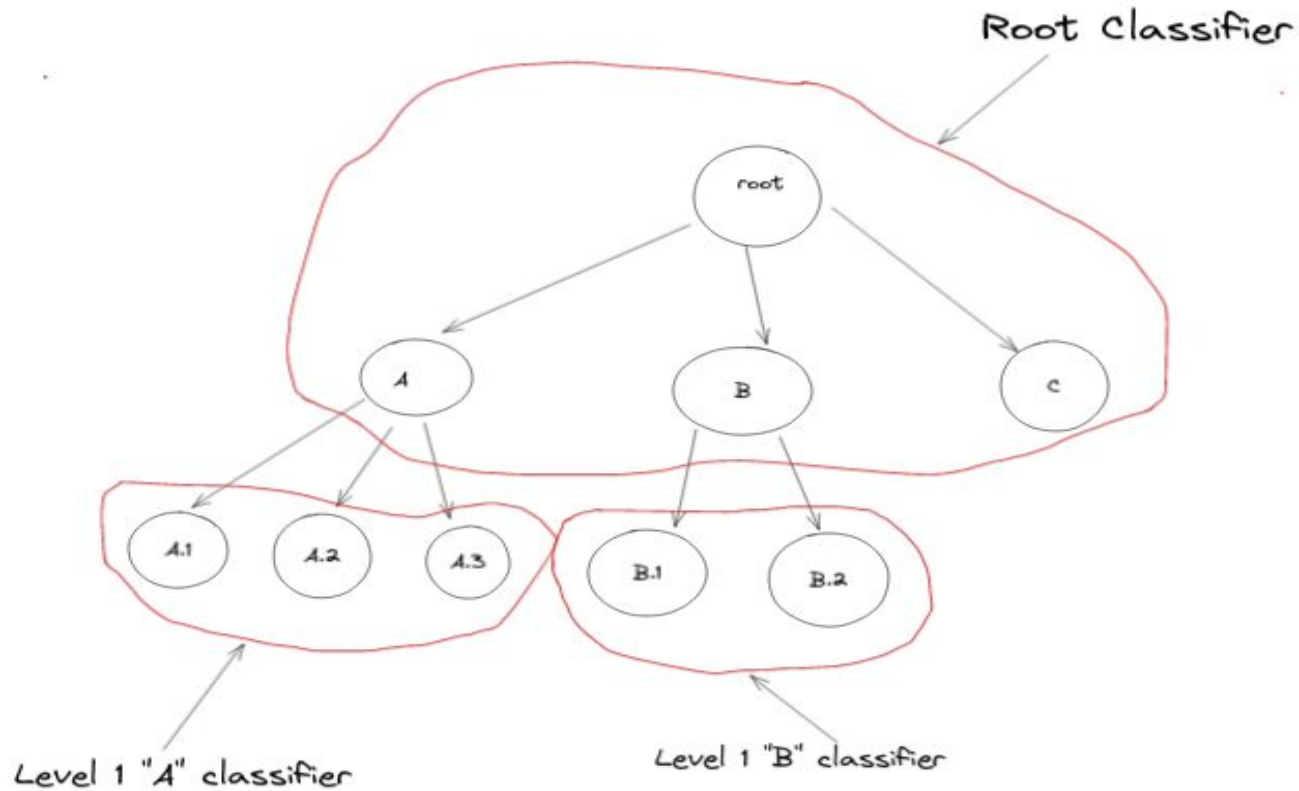


## Approach 2 (Inference Metric)

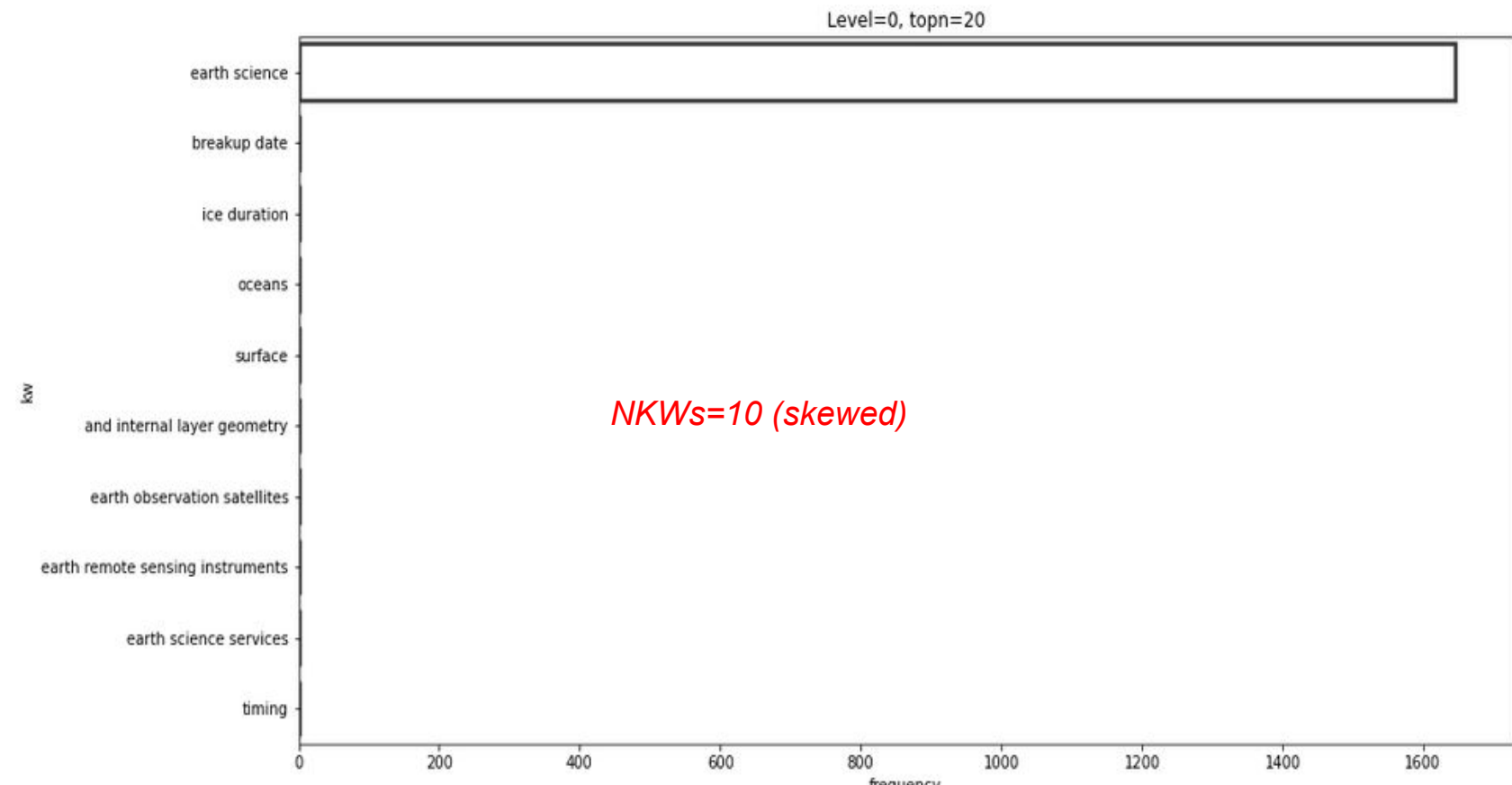
Jaccard Similarity



## Approach 3 (Hierarchical Classifiers)

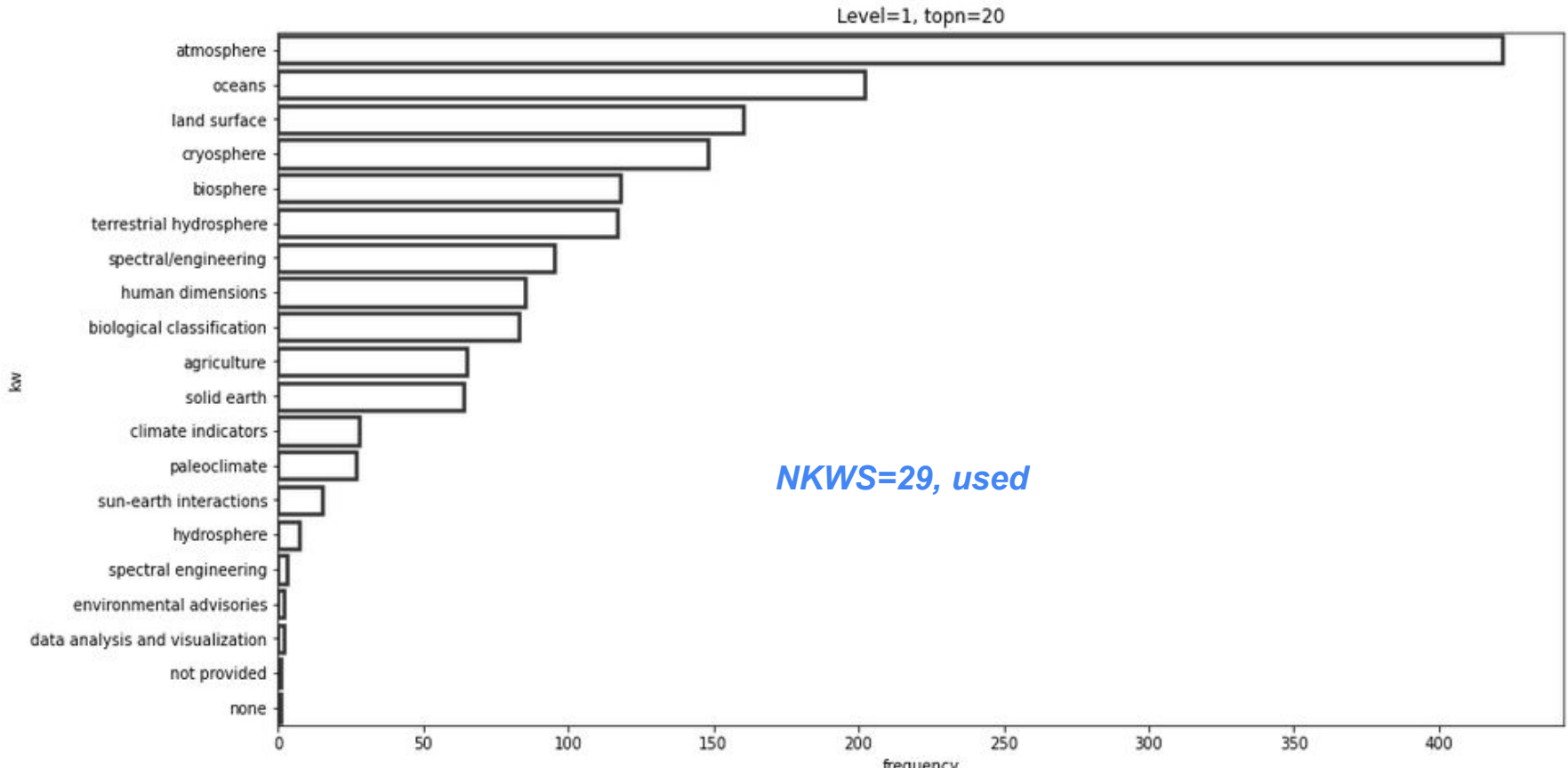


# Tag Analysis (level=0, root)

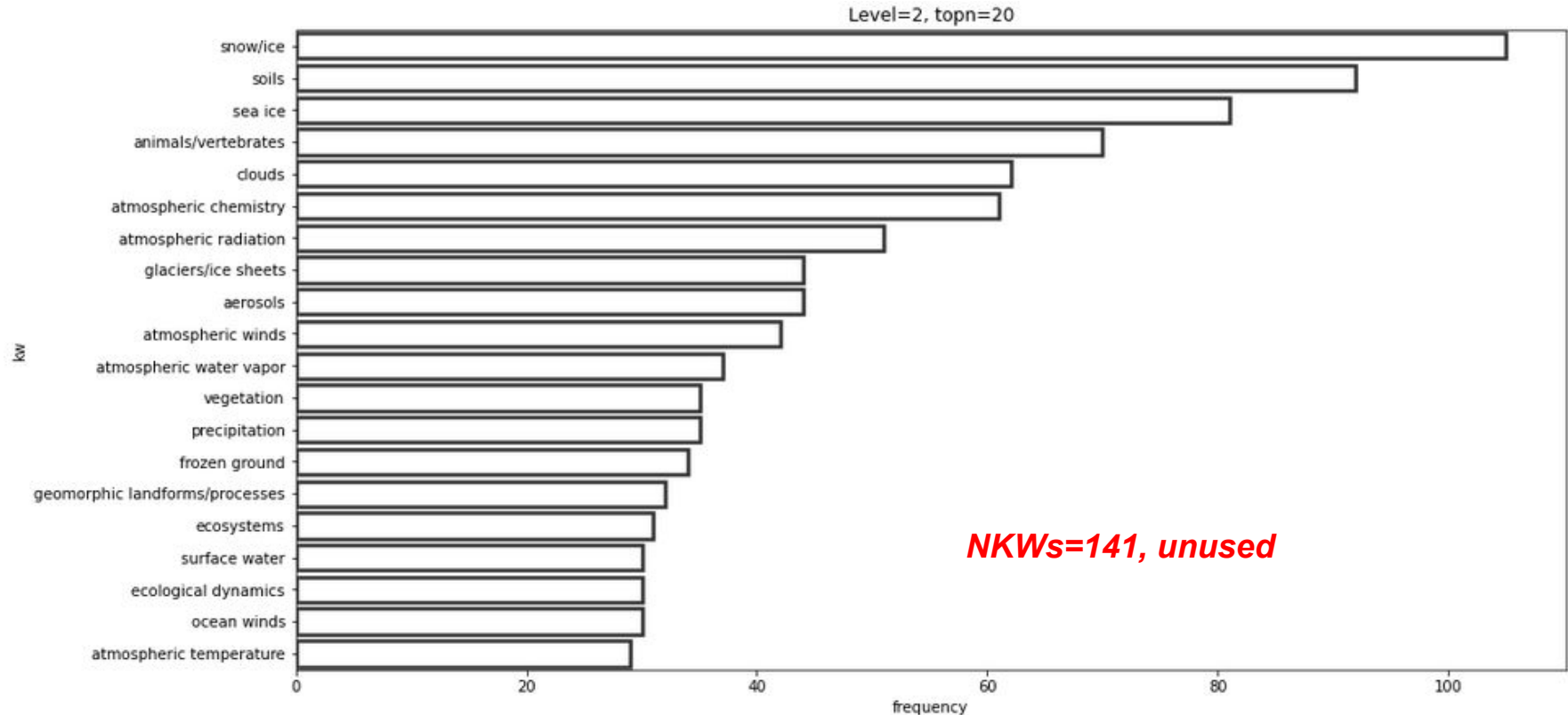




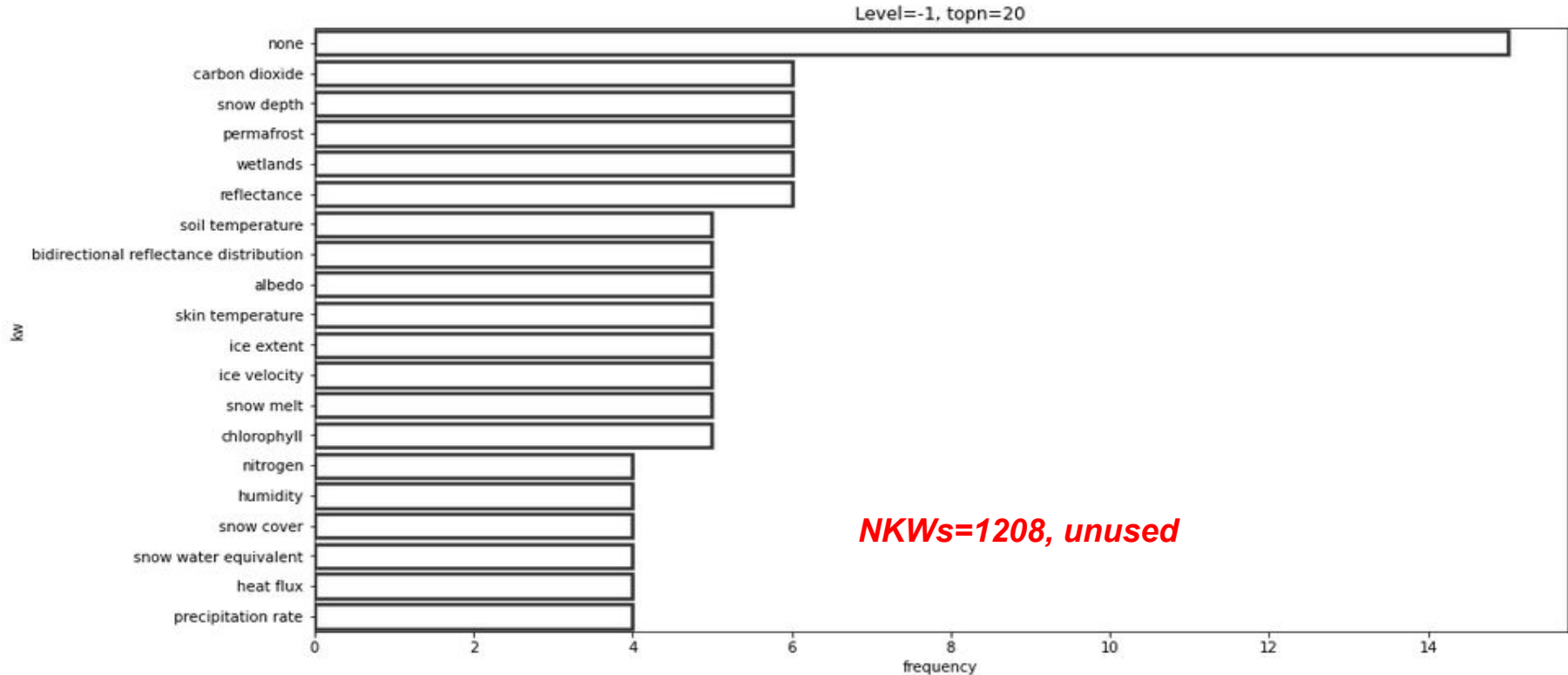
# Tag Analysis (level=1)



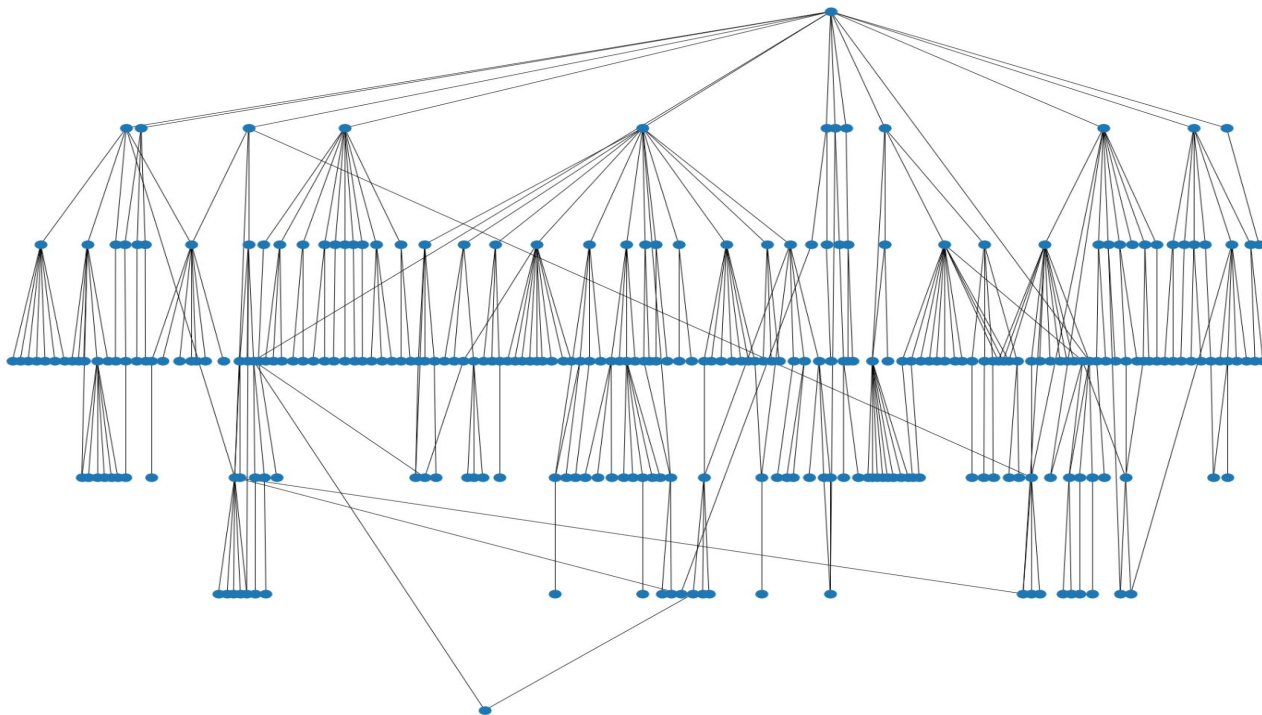
# Tag Analysis (Level=2)



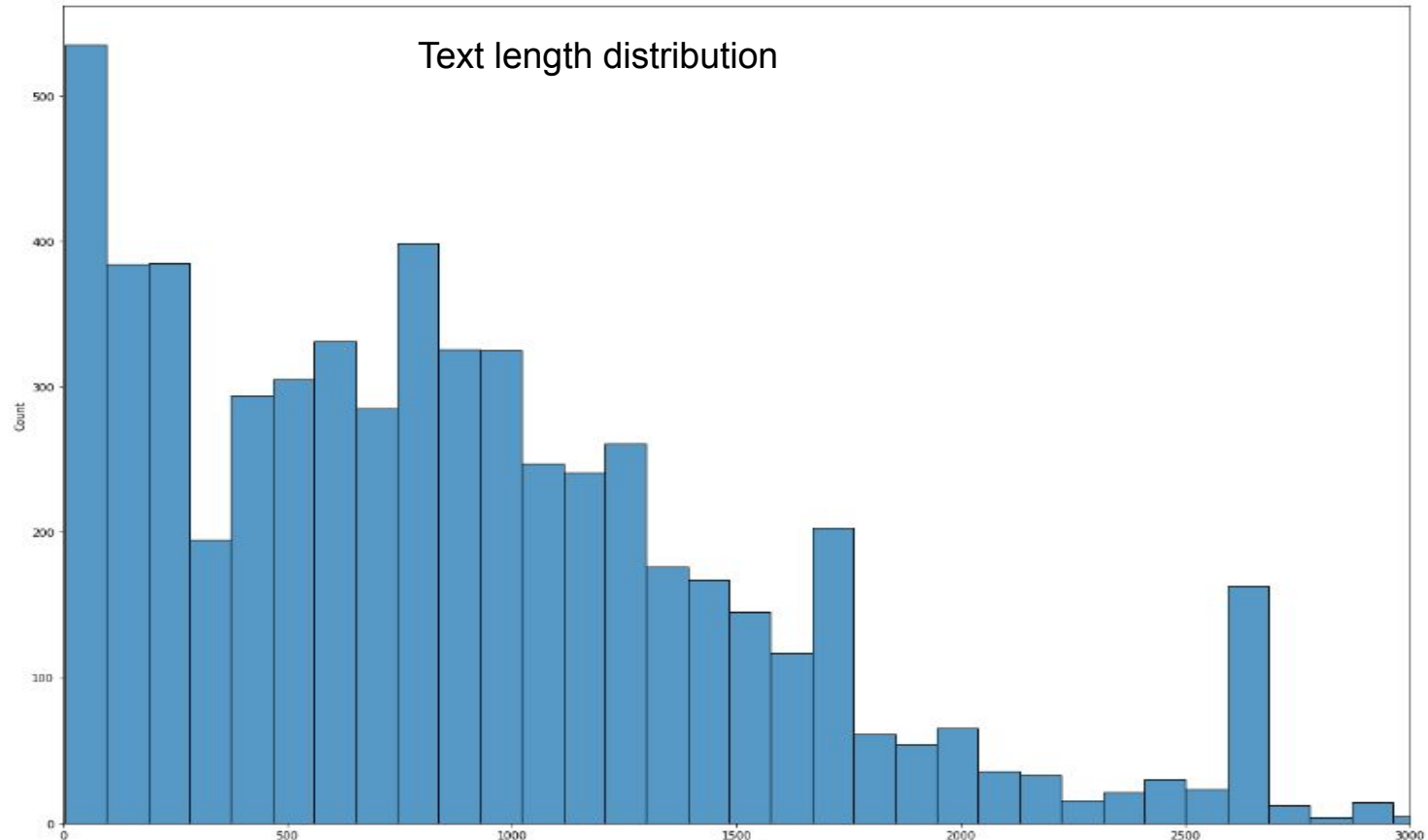
# Tag Analysis (Level=-1, leaves)



# Tag Analysis (Tree)



# Data Analysis



# Model Architecture

## ***BERT + Linear Layer***

```
(dense): Linear(in_features=768, out_features=3072, bias=True)
)
(output): BertOutput(
  (dense): Linear(in_features=3072, out_features=768, bias=True)
  (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
  (dropout): Dropout(p=0.1, inplace=False)
)
)
)
)
(pooler): BertPooler(
  (dense): Linear(in_features=768, out_features=768, bias=True)
  (activation): Tanh()
)
)
(classifier): Linear(in_features=768, out_features=22, bias=True)
(criterion): BCEWithLogitsLoss()
)
```

---

- After 10 epochs 402/402 [05:34<00:00, 1.20it/s, loss=0.0512, v\_num=7, train\_loss=0.0128, val\_loss=0.0812]
- `[{'test_loss': 0.07667940109968185}]`

*Sorry, couldn't get the proper log graph for training. :(*

# Jaccard Metrics

```
14     return metrics
```

```
In [22]: 1 res = eval_jaccard_json("outputs/inference.json")
```

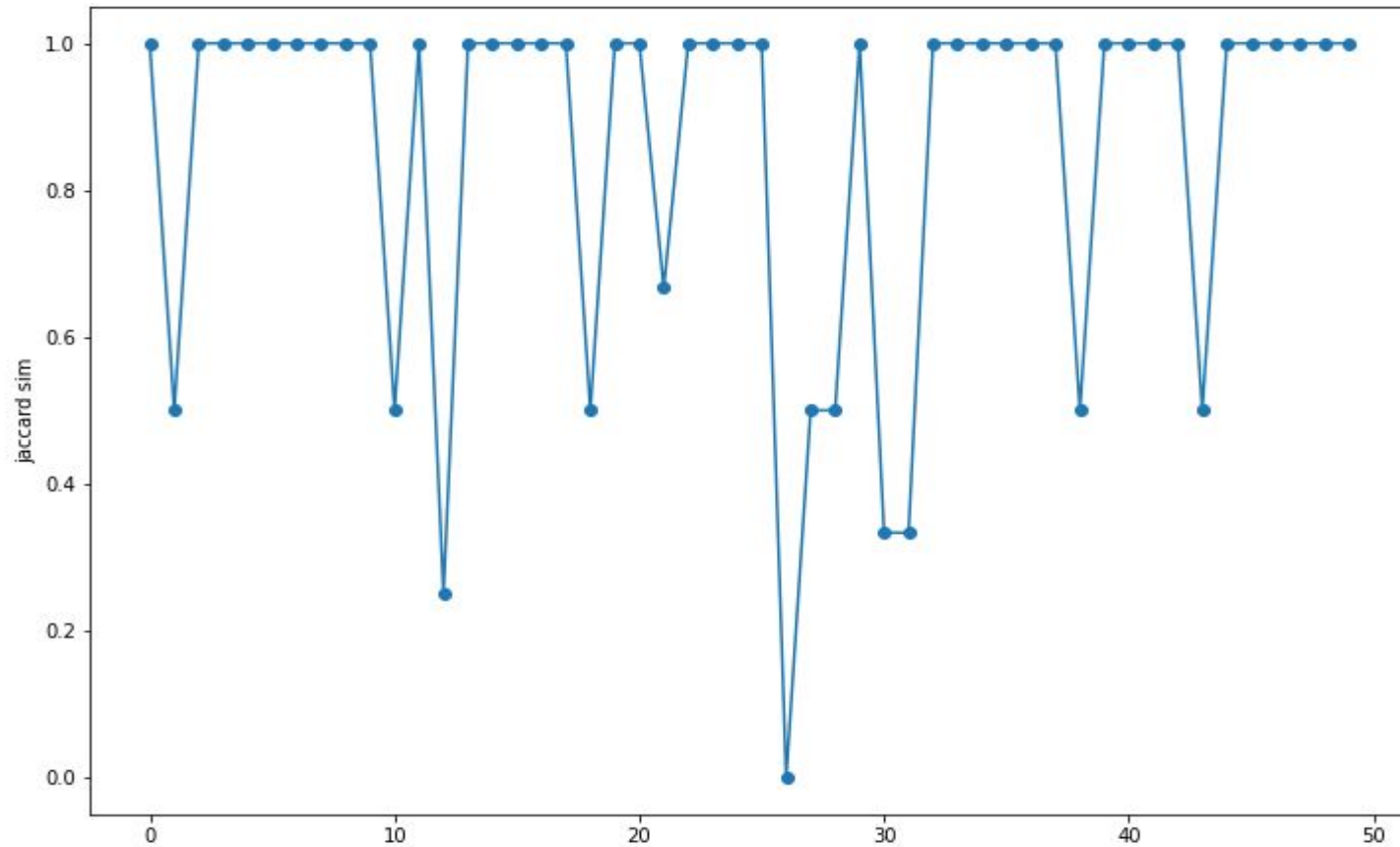
```
In [23]: 1 np.mean([p[-1] for p in res])
```

```
Out[23]: 0.8616666666666667
```

---



# Jaccard Metrics



# Hierarchical Multi-label Classification

NeuralNLP-NeuralClassifier framework

## NeuralNLP-NeuralClassifier framework

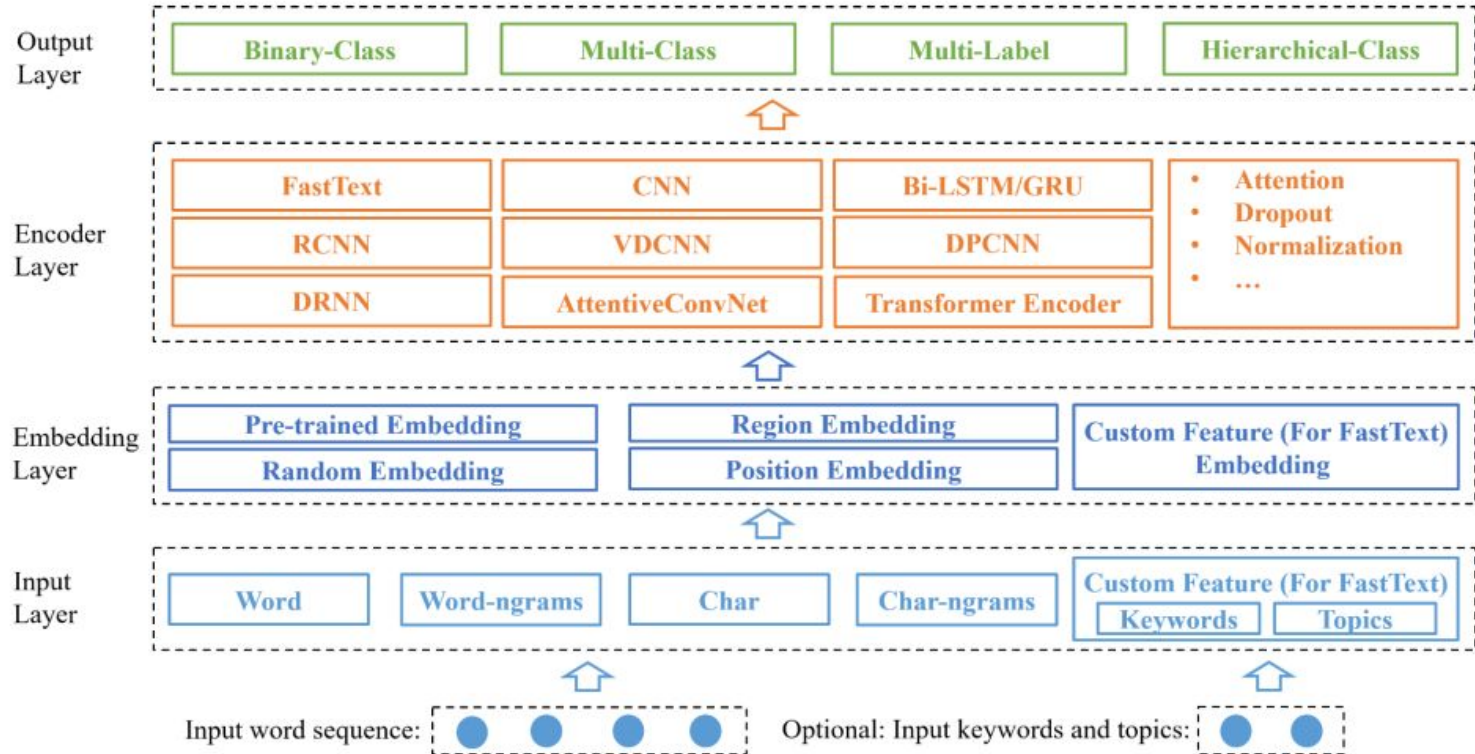


Figure 3: Architecture of NeuralClassifier. There are four layers: an input layer, an embedding layer, an encoder layer and output layer.

# Hierarchical Multi-label Classification

Evaluation after 25 epoch (best model at 23 epoch)

- Performance is precision: 0.607306, recall: 0.441860, fscore: 0.511538, right: 266, predict: 438, standard: 602.

# Hierarchical Multi-label Classification

## Train/Val Performance

- Train performance at epoch 25 is precision: 0.830596, recall: 0.889321, fscore: 0.858956, macro-fscore: 0.751426, right: 4805, predict: 5785, standard: 5403.  
Loss is: 0.003939
- ▼ • Validate performance at epoch 25 is precision: 0.505576, recall: 0.451827, fscore: 0.477193, macro-fscore: 0.047105, right: 272, predict: 538, standard: 602.  
Loss is: 0.007875.

# Hierarchical Multi-label Classification

Jaccard Similarity between Ground Truth and Predictions

(at different levels)

2021-07-30	11:11:44.999		DEBUG		__main__:<module>:4	-	[Level=0]		0.6161111111111111
2021-07-30	11:11:45.007		DEBUG		__main__:<module>:4	-	[Level=1]		0.4543849206349207
2021-07-30	11:11:45.013		DEBUG		__main__:<module>:4	-	[Level=2]		0.32370591676841676
2021-07-30	11:11:45.020		DEBUG		__main__:<module>:4	-	[Level=3]		0.19290151515151516
2021-07-30	11:11:45.024		DEBUG		__main__:<module>:4	-	[Level=4]		0.18427083333333333

# Improvements

- Representation Learning?
- Longformer?
- Improve training loop in NeuralNLP-NeuralClassifier?

# Reference

- loss function for multi-label classification
- multi-label classification using BERT
- [Longformer](#)
- Google's paper on representation learning for 2d documents
- [NeuralClassifier: An Open-source Neural Hierarchical Multi-label Text Classification Toolkit](#)
- [NeuraClassifier GitHub](#)



# Final Thoughts

- This was tedious (yet fun)
- It was difficult (challenging) because of “**hierarchies**”. Interesting problem nevertheless.