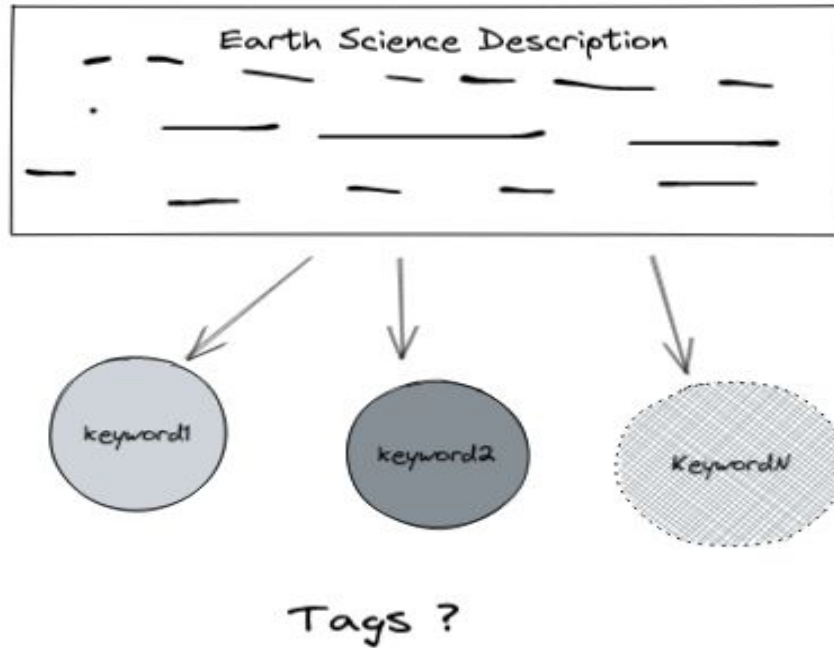




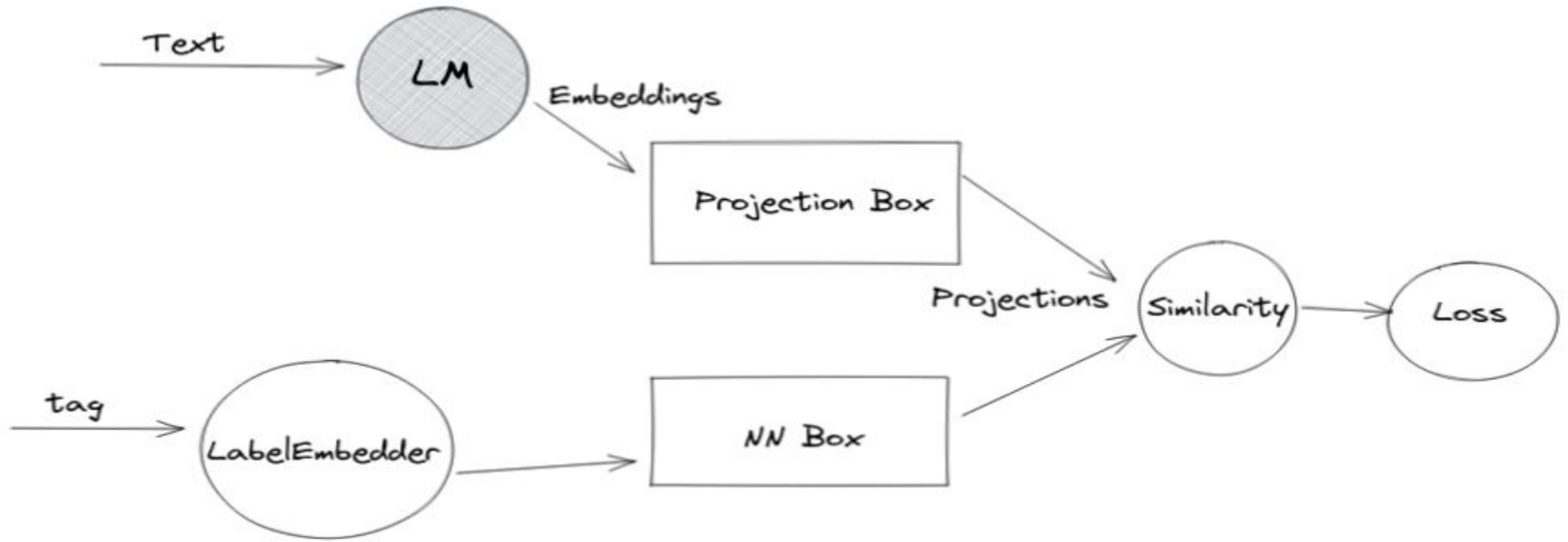
Earth Science

Where Nish meets
NLP meets UAH

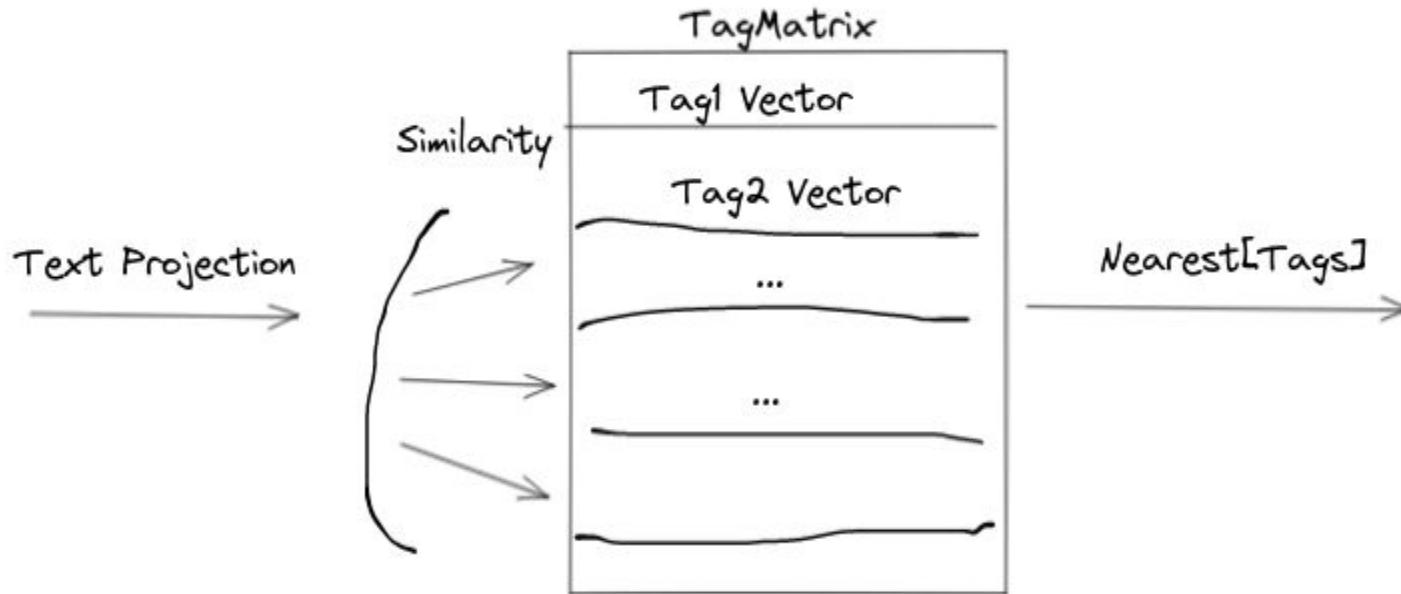
Problem Statement



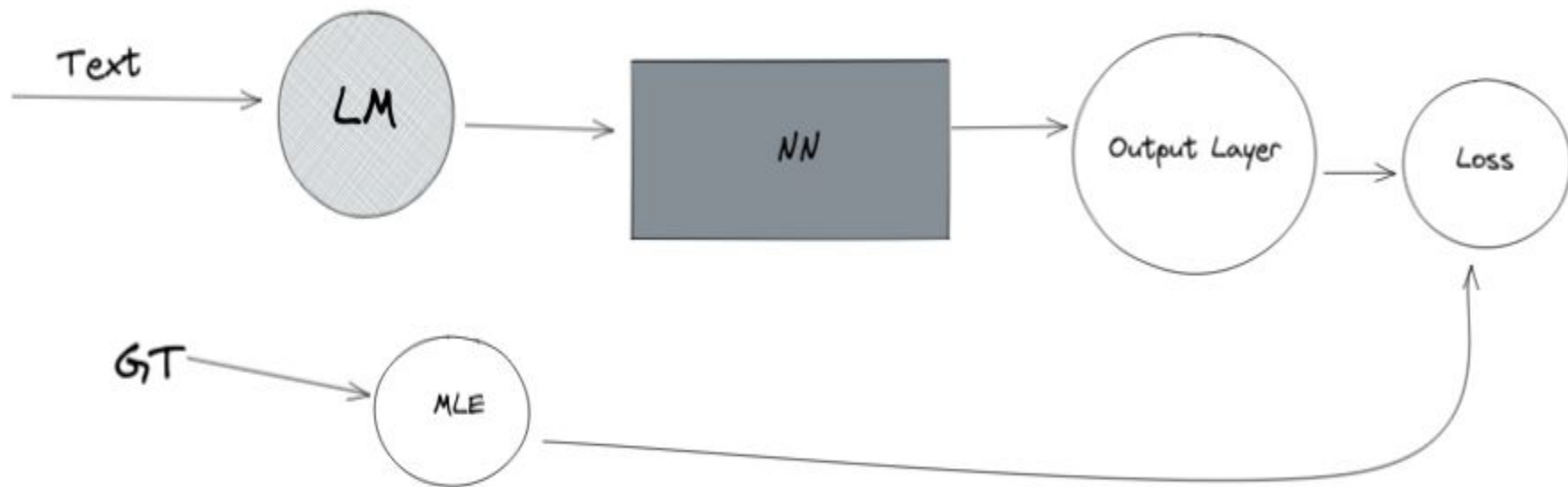
Approach 1



Approach 1 (Inference)

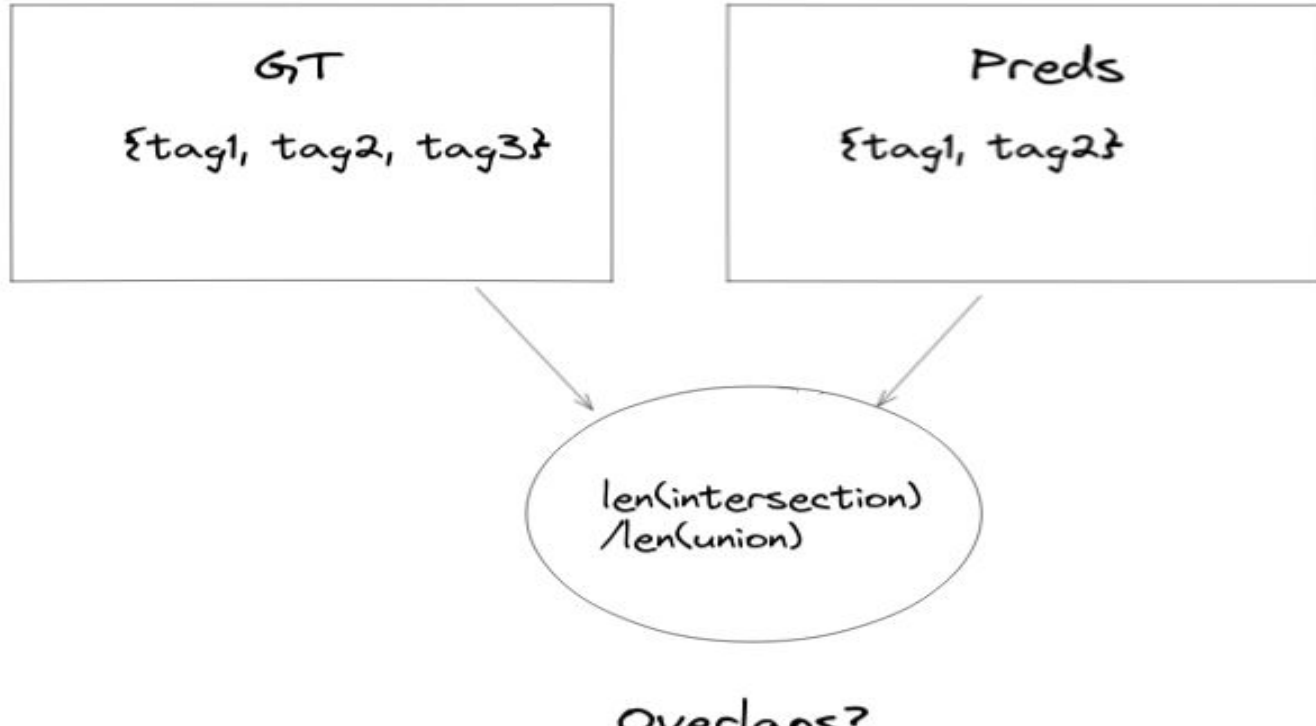


Approach 2

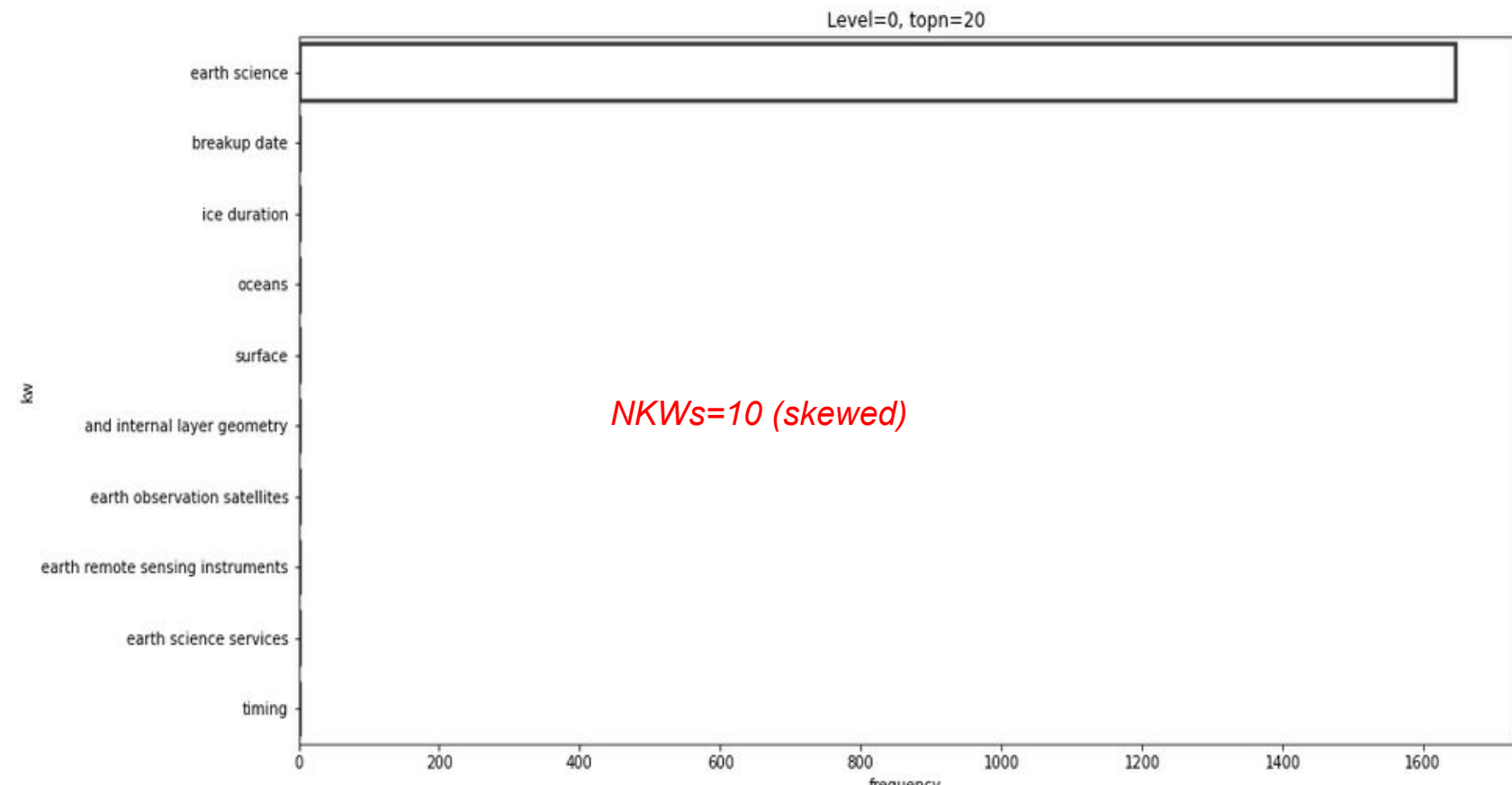


Approach 2 (Inference Metric)

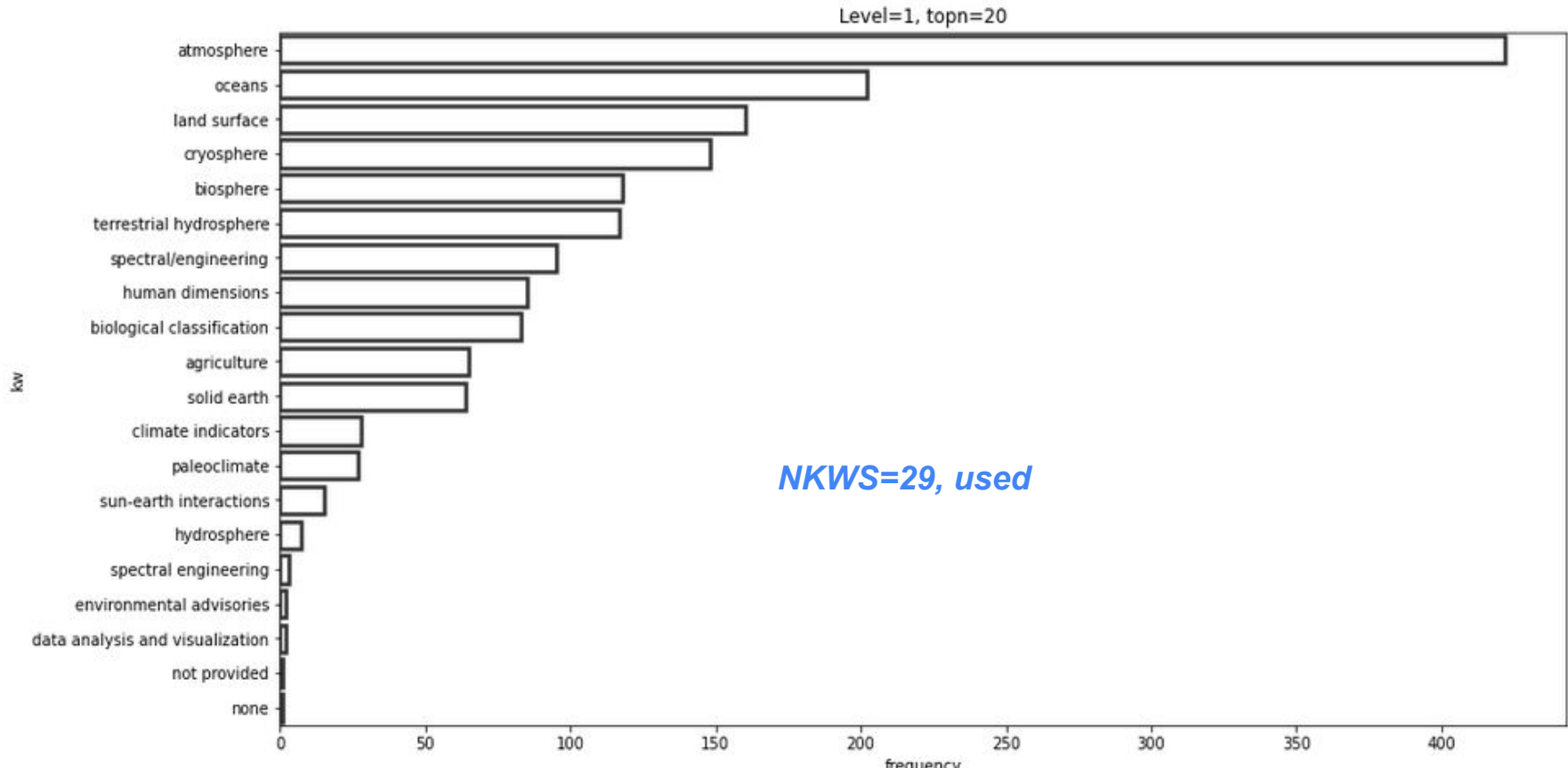
Jaccard Similarity



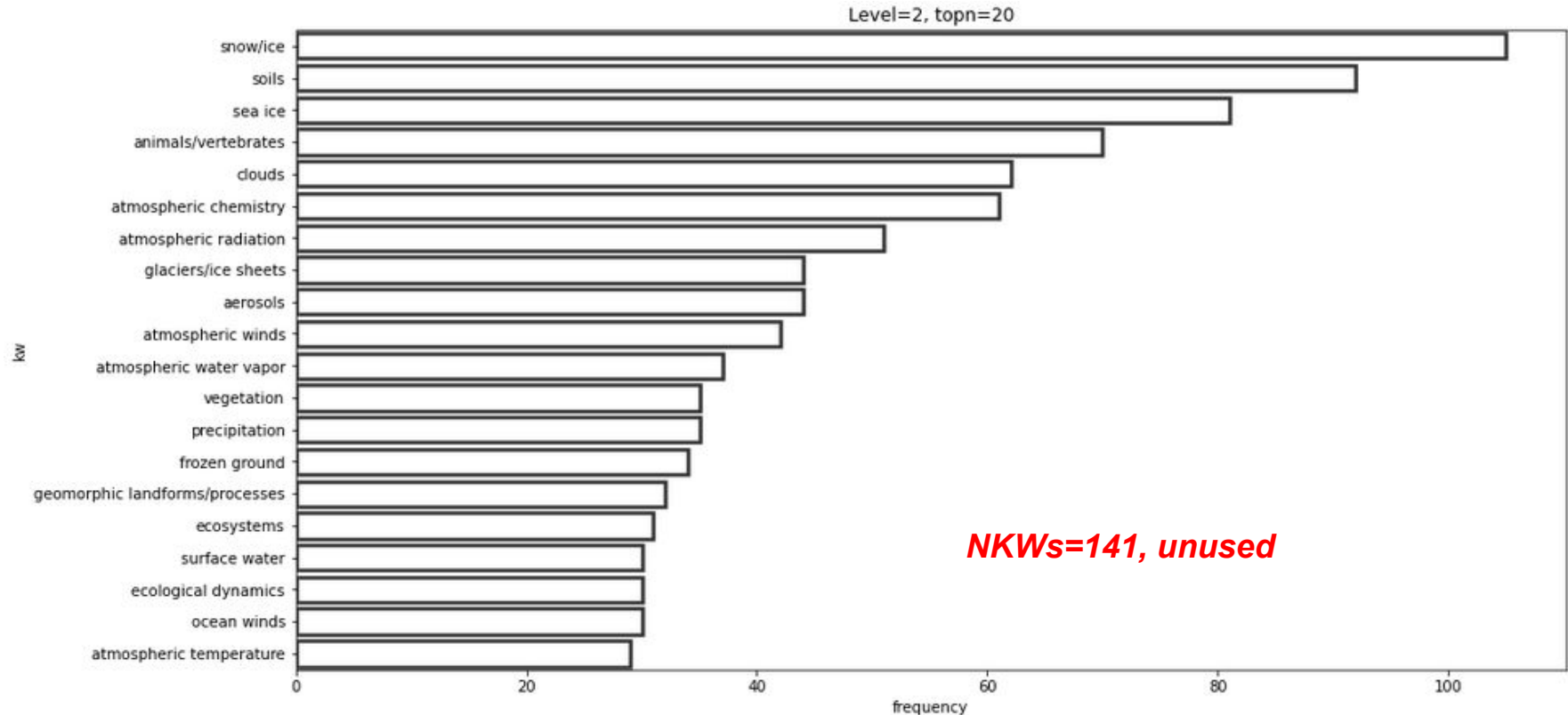
Tag Analysis (level=0, root)



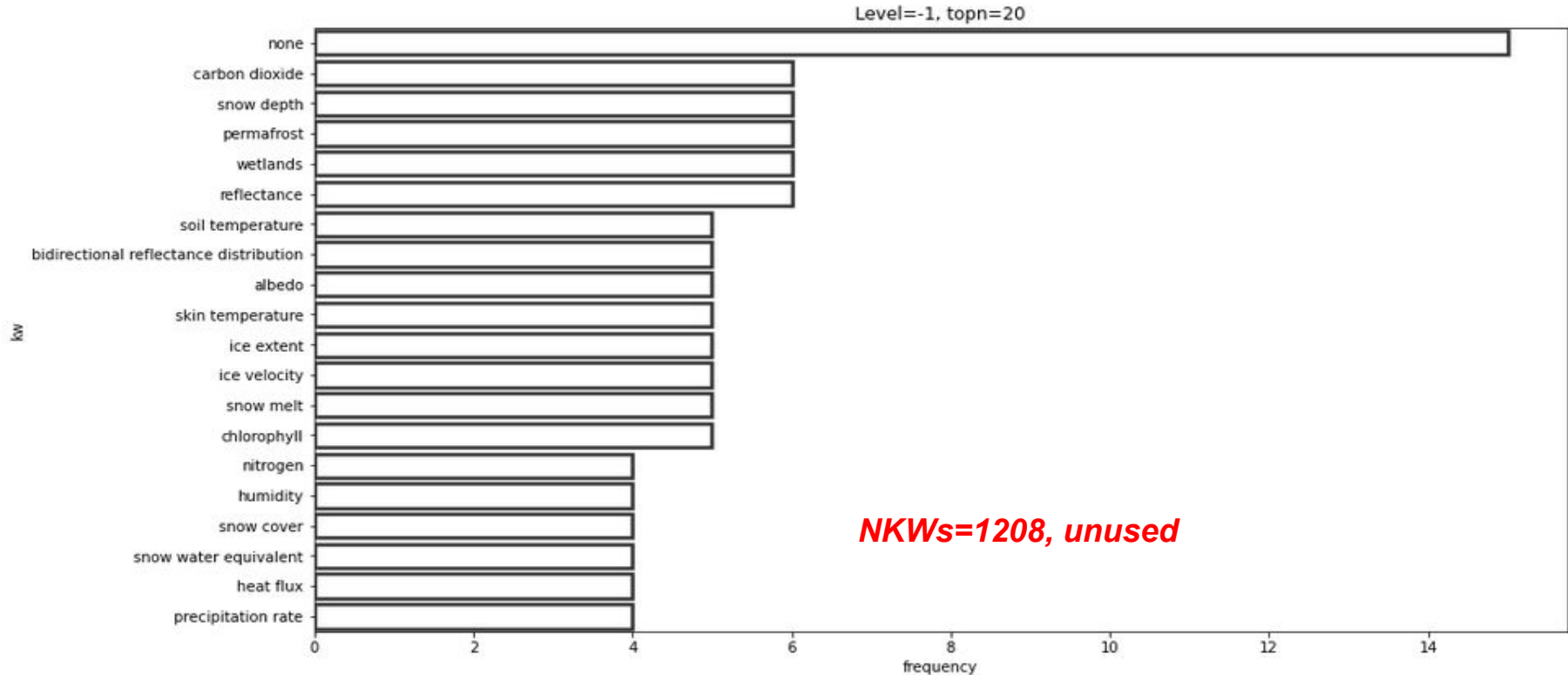
Tag Analysis (level=1)



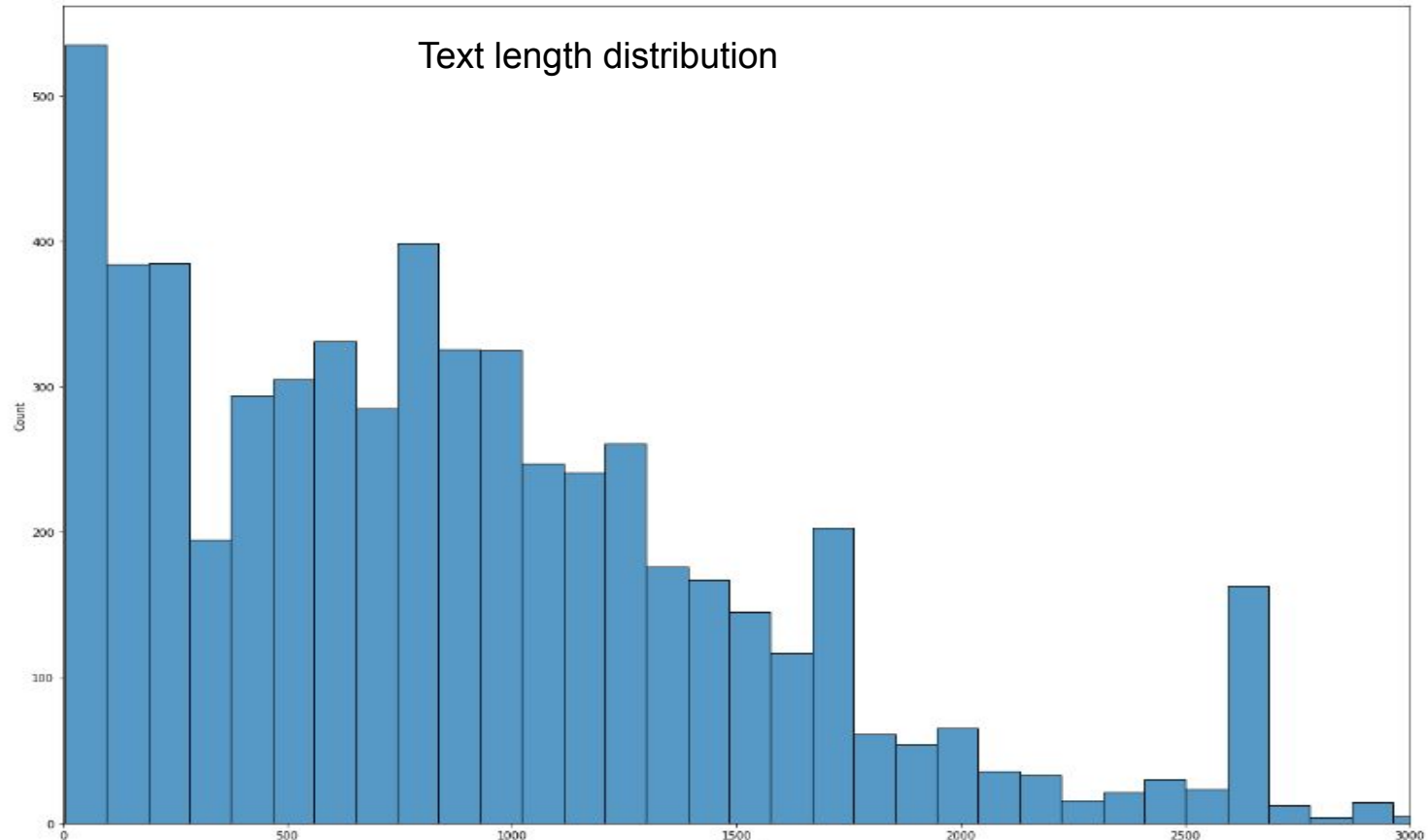
Tag Analysis (Level=2)



Tag Analysis (Level=-1, leaves)



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

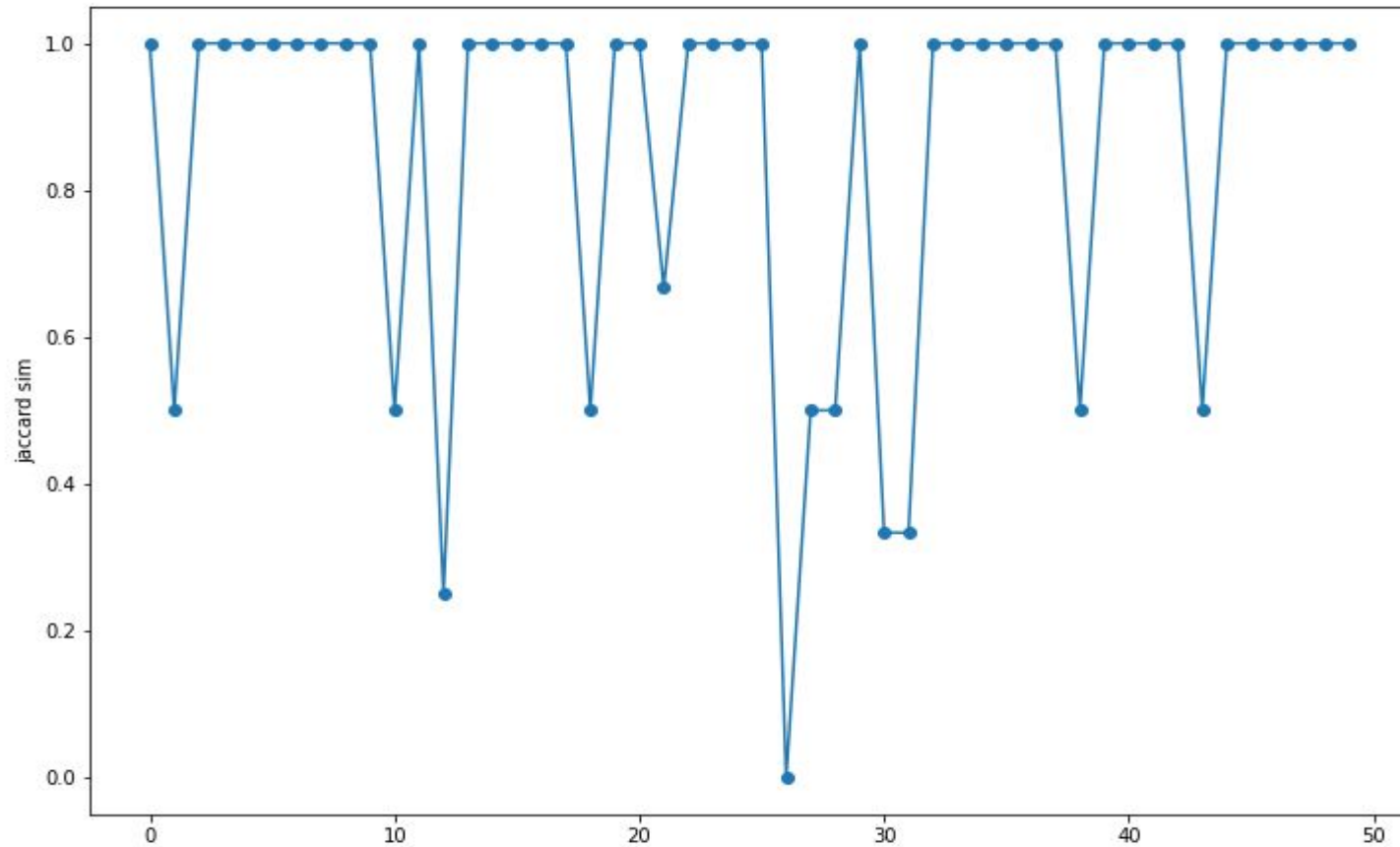
```
14     return metrics
```

```
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



Improvements

- Representation Learning?
- Longformer?
- ...

Reference

- loss function for multi-label classification
- multi-label classification using BERT
- [Longformer](#)
- Google's paper on representation learning for 2d documents

Final Thoughts

- This was tedious (yet fun)
- I overworked on this by halting all the other tasks.
- It was difficult (challenging) because of “**hierarchies**”. Interesting problem nevertheless.