BERT-NER Models for Sensitive Information

Pontus Hedman
Nils Alenäs
Erik Lundqvist
Mathias Ahlgren
Mårten Saltin
Alexander Nyström

Introduction

BERT-NER Models for Sensitive Information

- **Importance of Text Anonymization:**
  - Growing need for privacy protection
  - Compliance with regulations like GDPR
  - Manual anonymization is impractical for large datasets
- **Research Question:**
  - "How can Named Entity Recognition (NER) models be used for text classification to identify confidential information?"
- **Project Approach:**
  - Use BERT-based models trained on NER tasks to identify confidential information in text

What is Named Entity Recognition (NER)?

What is NER?

- Finds and labels important entities in text
- Categories like people, locations, organizations
- Makes unstructured text easier to analyze

Simple Example

- Text: "Sara visited Tokyo"
- NER identifies and tags:
  - Person: Sara
  - Location: Tokyo
- Shows how text gets structured quickly

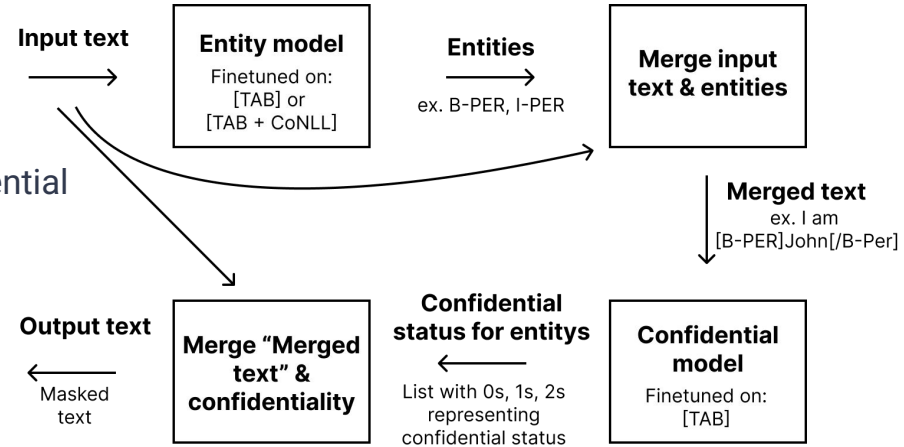
What have we done?

Aim to classify confidential information in a text.

This was done by:
Creating 2 different Near Entity Recognition (NER) models

- **“Entity model”:** Determines entities in a text
  - Finetuned on either 1 or 2 different datasets
- **“Confidential model”**: Determines if a Entity is confidential
  - Finetuned on 1 dataset

Then a pipeline that fuses these 2 different model together



Papers

The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization
Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, Montserrat Batet

- Published in **Computational Linguistics** 2022
  - A level 2 Journal in Norwegian List

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

- Original BERT paper
- Published in **NAACL-HLT** 2019
  - Top-ranked conference

Matching the Blanks: Distributional Similarity for Relation Learning
Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, Tom Kwiatkowski

- Published in **ACL 2019**
  - Top-ranked conferences

Datasets overview

CoNLL-2003:

- Well-established NER dataset with 14,000 entries
- Contains news articles
- Uses IOB (Inside-Outside-Beginning) tagging scheme
- Gold labels
- Entity tags used:
  - PER (Person)
  - ORG (Organization)
  - LOC (Location)
  - MISC (Miscellaneous)

TAB (Text Anonymization Benchmark, 2022):

- 1,300 European Court of Human Rights cases
- Contains both entity types and confidentiality status
- Doesn’t use IOB
- Gold labels
- Same entities as in CoNLL, and additionally it has:
  - QUANTITY
  - Confidentiality classifications (CLASSIFIED, DECLASSIFIED)
  - CODE
  - DATETIME
  - DEM (Demographic)

Methods

Preprocessing of data

- Differences between datasets
- Class explosion
    - Because CoNLL uses IOB system, TAB goes from 9 -> 17 classes

BERT architecture

- A BERT model has understands language and context
- Uses a transformers with attention to achieve this
- Has context of 512 tokens

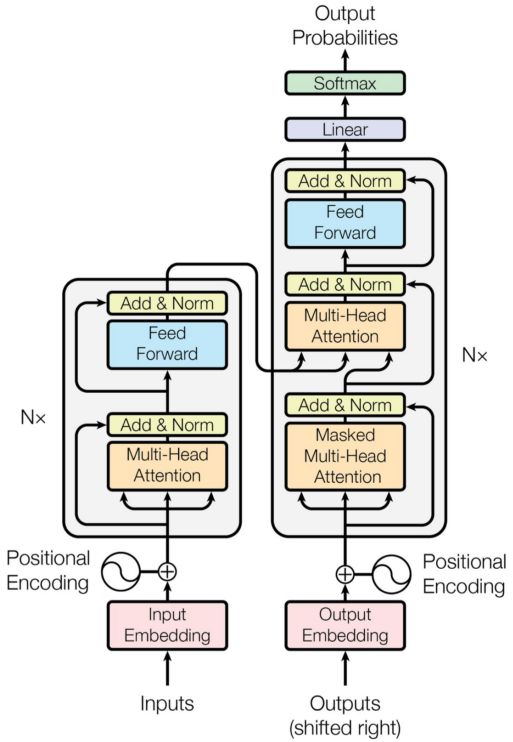
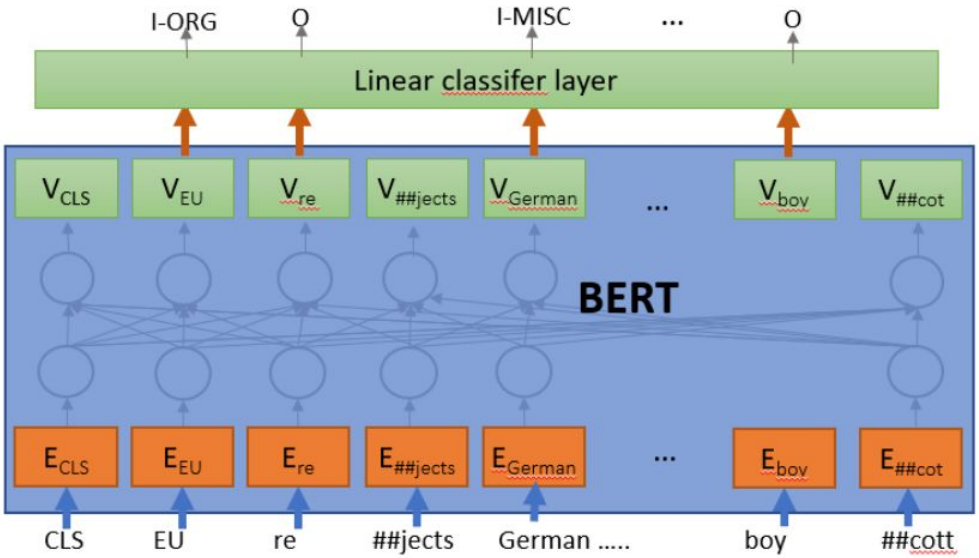


Figure 1: The Transformer - model architecture.

https://cameronrwolfe.substack.com/p/many-languages-one-deep-learning

NER models

- Based on BERT
- Named-entity classification by using linear classifier layer
- Two types of models:
  - NER “Entity model”, fine-tuned on:
    - TAB
    - TAB and CoNLL
  - NER “Confidentiality model”

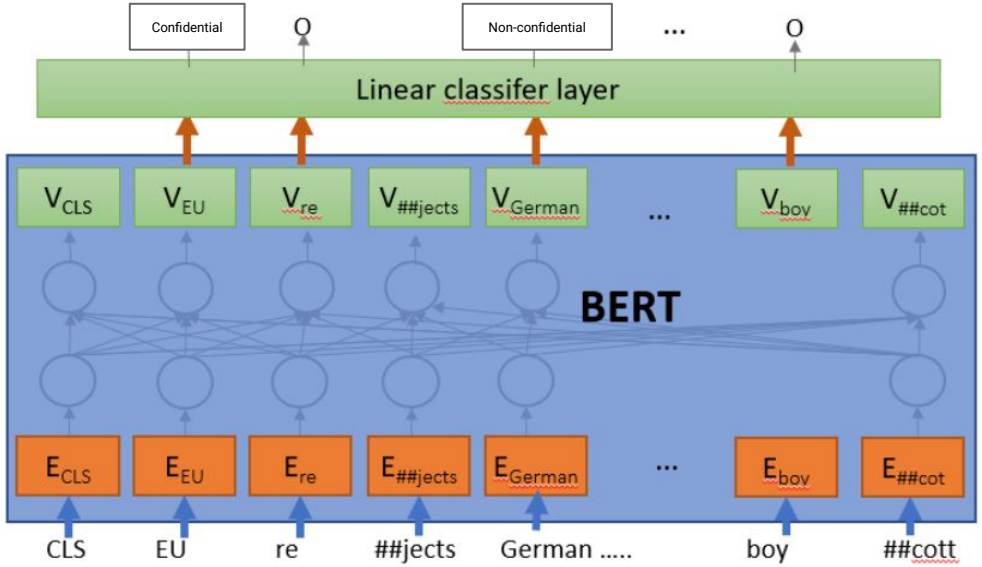


https://cameronrwolfe.substack.com/p/many-languages-one-deep-learning
https://www.analyticsvidhya.com/blog/2021/06/why-and-how-to-use-bert-for-nlp-text-classification/

Confidential classification model

- Identify confidential NER tags
- Comparison methods
  - Gold standard
  - NER output
- Classes
  - [Non-NER (0)]
  - [Non-confidential (1)]
  - [Confidential (2)]
- Merging strategies
  - Token
  - [Entity] Token
  - [Entity] Token [\Entity]
  - [Entity]

Ex. I am [B-per]John[\B-per]



Pipeline

Why Two Models?

Our idea was to use two models:

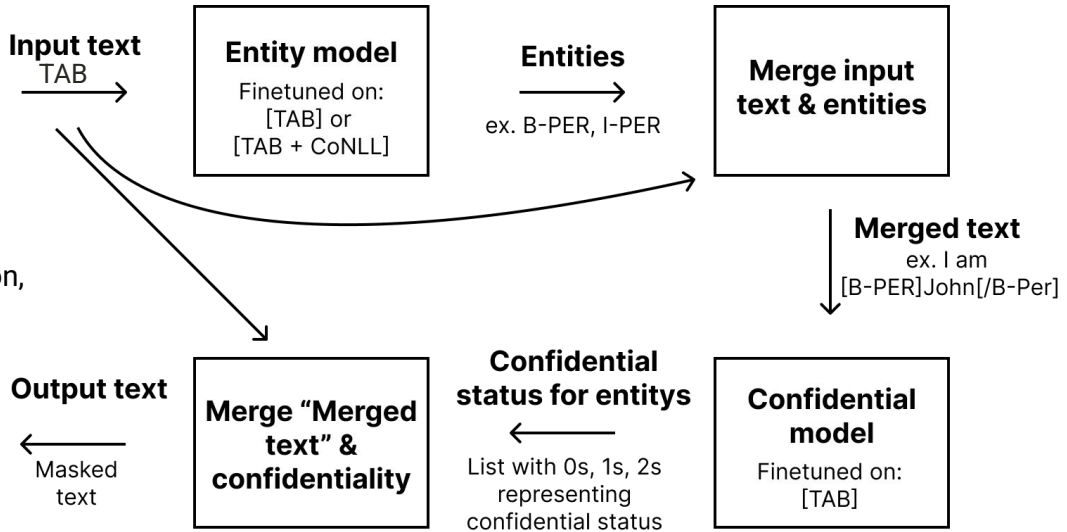
“Entity model”: Identifies Named Entities (NER).

“**Confidential model”**: Classifies confidentiality.

Pipeline Overview

1. **Input Text → Model 1**
   - Outputs entities (e.g., Person, Organization, Location). Total 17 classes
2. **Merge Entities with Text**
3. **Merged Text → Model 2**

   Outputs **three classifications**:
   - **0 =** Non-Confidential Non-Entity
   - **1** = Confidential Non-Entity
   - **2** = Confidential Entity
4. **Final Merge → Masked Output Text**



Results

Evaluation Metrics

Recall:

- Of the actual positive ones: which did we predict
- *Example: We found 8 ripe apples out of 20 total ripe apples = 40% recall*

Precision:

- Of the positives predictions: which ones are correct
- *Example: 8 of 10 apples we picked were actually ripe = 80% precision*

F1 Score:

- Harmonic mean between precision and recall
- *Example: High precision but low recall gives mediocre F1 score*
- *Perfect detector: F1 = 1.0*
- *Useless detector: F1 = 0*

$$\frac{TP}{TP + FN}$$

$$\frac{TP}{TP + FP}$$

$$\frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Results of NER “Entity model”

The results of how well “Entity model” predicted entities:

- Fine-tuning on CoNLL and TAB gives better results than only on TAB
- The evaluation is done on the TAB dataset.

Model	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Weighted Precision**	**Weighted Recall**	**Weighted F1-score**
CoNLL + TAB	0.68	0.48	0.50	0.86	0.84	0.86	0.83
TAB	0.46	0.40	0.41	0.84	0.79	0.84	0.81

Entity metrics for “Entity models”

See how the models have performed on individual 17 classes

What is interesting?

- Large class imbalance contributes to the low average metrics, that is why weighted average is very different.
- The smaller classes metrics have improved in ConLL + TAB, which impacts the non-weighted metrics

Trained on conLLU and TAB

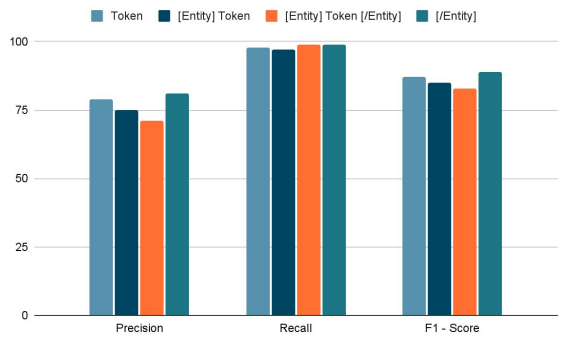
labels	precision	recall	f1-score	support
0	0.88	0.97	0.92	44335
1	0.93	0.92	0.92	781
2	0.90	0.91	0.91	2113
3	0.78	0.36	0.49	1807
4	0.82	0.79	0.81	6354
5	0.52	0.74	0.61	318
6	1.00	0.04	0.08	70
7	0.31	0.01	0.02	518
8	0.26	0.01	0.02	2098
9	0.47	0.48	0.47	155
10	0.53	0.24	0.33	327
11	0.95	0.53	0.68	80
12	0.00	0.00	0.00	5
13	0.77	0.88	0.82	1452
14	0.79	0.81	0.80	1802
15	0.86	0.42	0.56	519
16	0.75	0.00	0.01	790
accuracy			0.86	63524
macro avg	0.68	0.48	0.50	63524
weighted avg	0.84	0.86	0.83	63524

Only trained on TAB

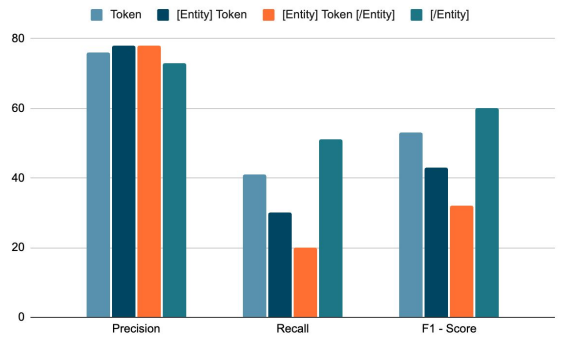
labels	precision	recall	f1-score	support
0	0.87	0.97	0.91	44335
1	0.96	0.80	0.87	781
2	0.83	0.91	0.87	2113
3	0.71	0.27	0.39	1807
4	0.78	0.72	0.75	6354
5	0.40	0.65	0.50	318
6	0.00	0.00	0.00	70
7	0.00	0.00	0.00	518
8	0.00	0.00	0.00	2098
9	0.49	0.34	0.40	155
10	0.47	0.10	0.17	327
11	0.00	0.00	0.00	80
12	0.00	0.00	0.00	5
13	0.74	0.87	0.80	1452
14	0.68	0.78	0.73	1802
15	0.87	0.40	0.55	519
16	0.00	0.00	0.00	790
accuracy			0.84	63524
macro avg	0.46	0.40	0.41	63524
weighted avg	0.79	0.84	0.81	63524

Merging strategy results for “Confidentiality model”

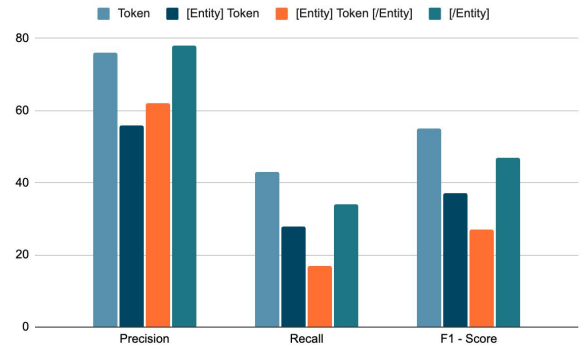
Non Confidential
Ex: ‘was’, ‘sued’, ‘by’, ‘the’



Non Confidential With Entity
Ex: ‘Danish’, ‘government’



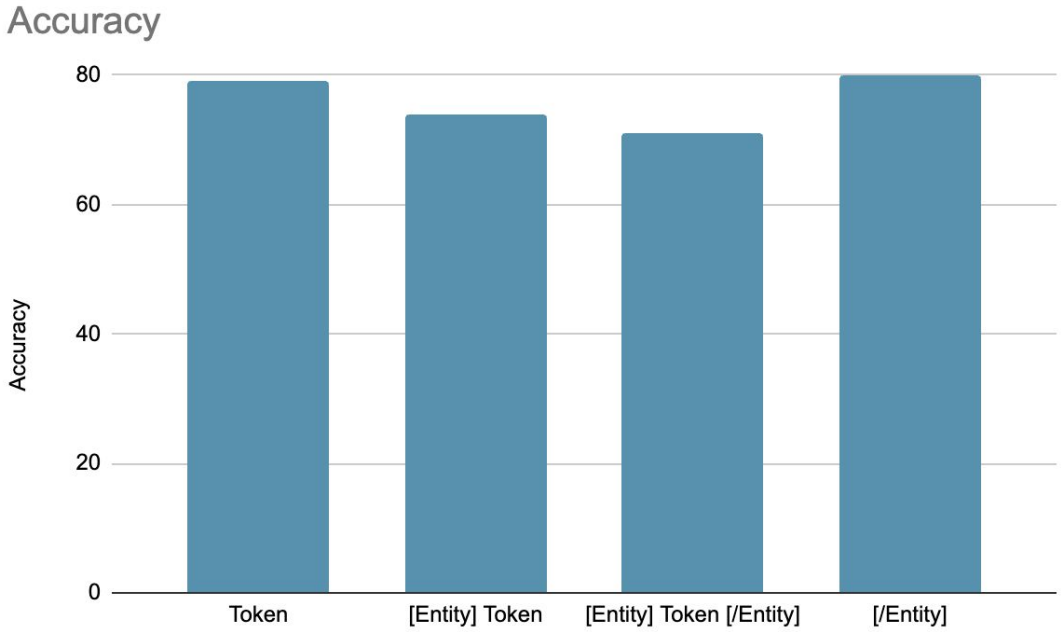
Confidential With Entity
Ex: ‘John’, ‘Doe’



Example sentence: **“John Doe was sued by the Danish government.”**
Key takeaways:

- Generally not good at classifying confidential tags
- Very bad recall, can’t identify entities very well.

Token merging strategies

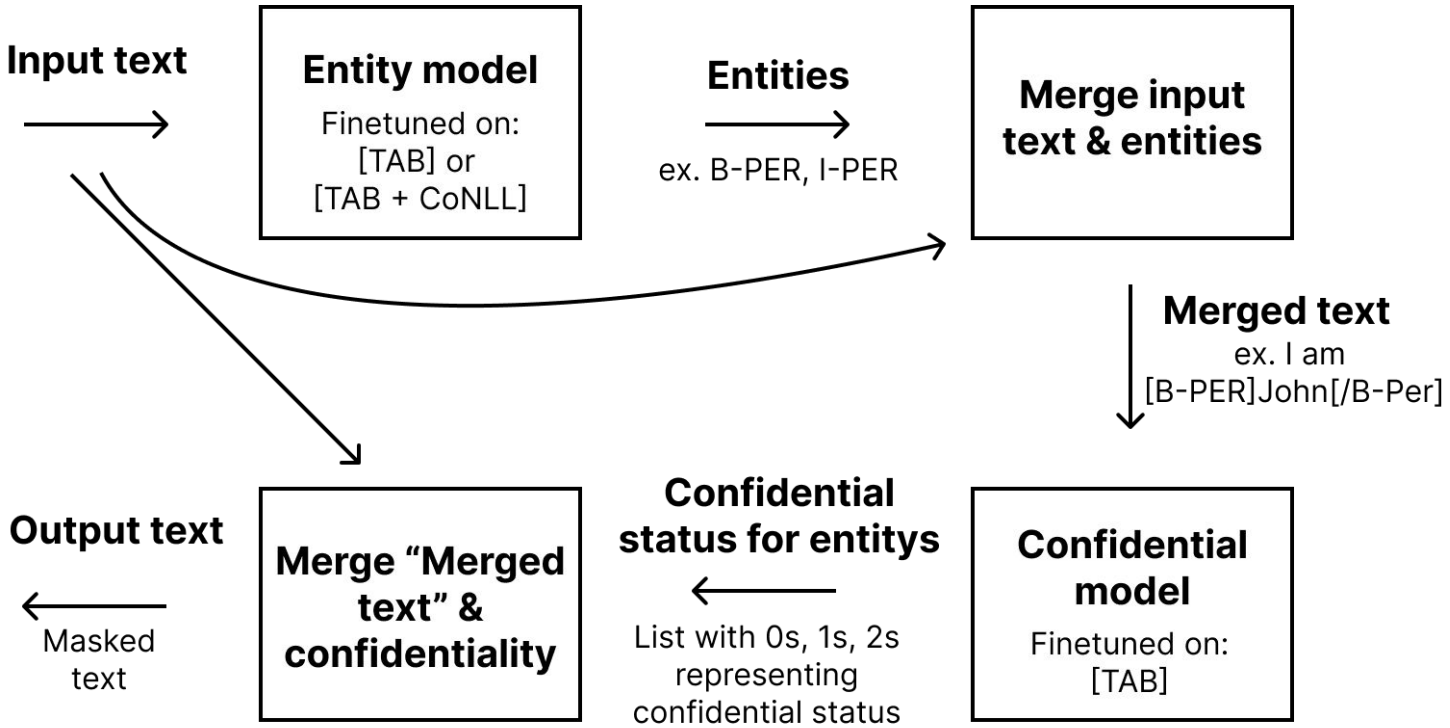


It makes a tiny difference

Graph shows us the accuracy of the different tagging strategies on the confidentiality prediction

- [Entity] [/Entity] is the worst, perhaps affect BERT context length (only 512 tokens fit)
- Other unknown differences
- First model is almost irrelevant

Putting it all together



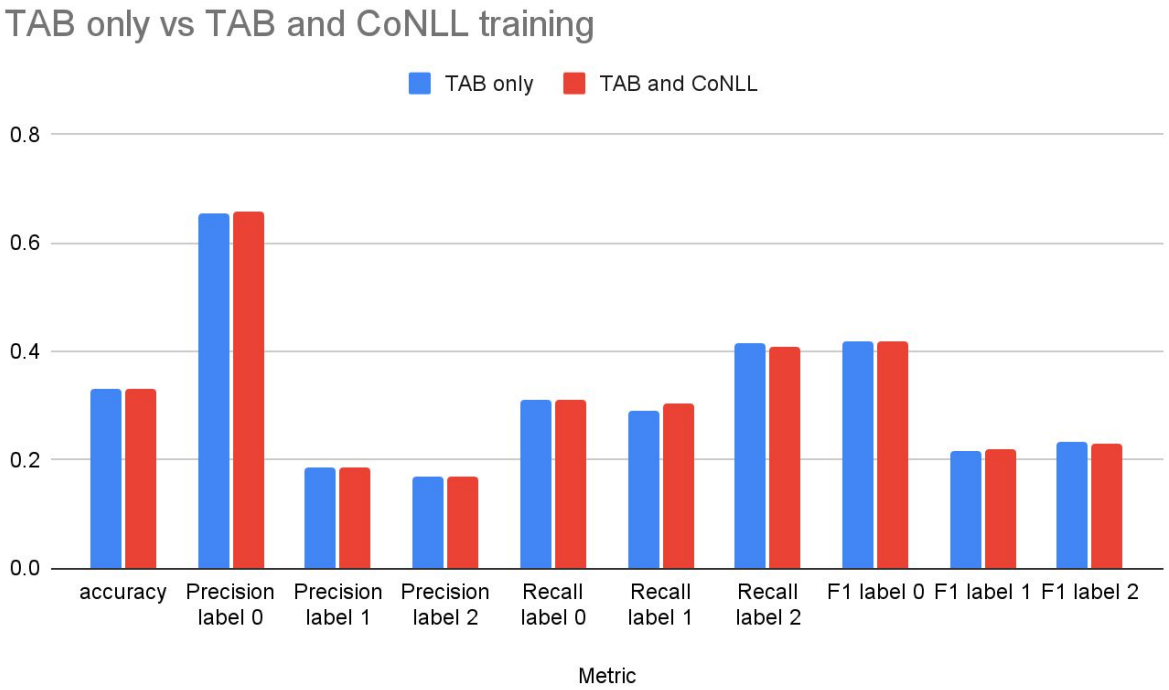
Results of Pipeline

Generally poor results

Precision of 15% on classification

Every “Entity model” miss accumulates in the “Confidentiality model”

No real difference between Entity BERT models



Relating Results to Other Work

System	Set	R_{di+qi}	ER_{di}	ER_{qi}	P_{di+qi}	WP_{di+qi}
Neural NER (RoBERTa fine-tuned on Ontonotes v5)	Dev	0.910	0.970	0.874	0.447	0.531
	Test	0.906	0.940	0.874	0.441	0.515
Presidio (default)	Dev	0.696	0.452	0.739	0.771	0.795
	Test	0.707	0.460	0.758	0.761	0.790
Presidio (+ORG)	Dev	0.767	0.465	0.779	0.549	0.622
	Test	0.782	0.463	0.802	0.542	0.609

The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization:

Tried different models - Only Neural NER that is comparable to ours.

- R_{di+qi} (Recall) and P_{di+qi} (Precision)
- Other evaluations are self proposed and hard to compare with

They did everything in a single model, not a pipeline

Conclusion & Future research

Conclusion

Confidential model

The “Confidential model” performs slightly better when entities are replaced with [Entity]. This suggests that it doesn’t understand the context. It also has a bias toward non-entities.

Full pipeline

Both recall and precision became worse when running the full pipeline. The most notable was precision for label 1 and label 2 went from around 80% to 20%.

Accumulating errors

Without a perfect “Entity model”, the pipeline performs worse than not annotating entities at all.

Improvements and future research

Not anonymized

We have just classified data that is confidential.

- Anonymizing requires removing all instances of confidential information, we have not look at this.
  - For example, if ‘John’ appears multiple times, all ‘Johns’ must be masked.
- Use custom loss functions

Better BERT-model

- RoBERTa, ALBERT or BERT-base-large
- Or use longer context-length

Class imbalance

TAB wasn’t annotated with IOB. Changing it caused class imbalance in TAB

- Use Conditional Random Fields (CRF).
  - Helps understand what comes after, ex. B-PER → I-PER
- class weighting.

Thank you for listening!