Medical Named Entity Recognition

Or Levitas

April 10, 2025

Abstract

This paper presents a Named Entity Recognition (NER) methods designed for the extraction of crucial medical entities specifically conditions, procedures, and medications from clinical texts. I implemented and evaluated three distinct NER approaches: a rule-based system, Conditional Random Fields (CRF), and Iterated Dilated Convolutional Neural Networks (ID-CNNs). The experimental results demonstrate that the optimal model achieves an overall F1 score of 0.9540.

Contents

1 Exploratory data analysis 2

2 Data processing 3

2.1 Span notation . . . 3

2.2 Data cleaning . . . 4

2.3 Dataset Splitting Strategy . . . 5

2.4 Train-Test Statistics . . . 6

3 Models 7

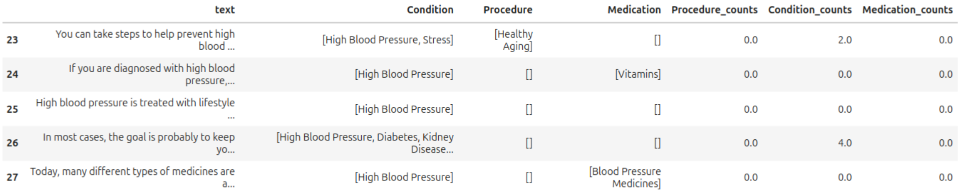
3.1 Rule-Based . . . 7

3.2 Conditional Random Field (CRF) Model . . . 7

4 Results and Conclusions 9

5 Future Directions 10

6 Reference 11



	text	Condition	Procedure	Medication	Procedure_counts	Condition_counts	Medication_counts
23	You can take steps to help prevent high blood ...	[High Blood Pressure, Stress]	[Healthy Aging]	[]	0.0	2.0	0.0
24	If you are diagnosed with high blood pressure,...	[High Blood Pressure]	[]	[Vitamins]	0.0	0.0	0.0
25	High blood pressure is treated with lifestyle ...	[High Blood Pressure]	[]	[]	0.0	0.0	0.0
26	In most cases, the goal is probably to keep yo...	[High Blood Pressure, Diabetes, Kidney Disease...	[]	[]	0.0	4.0	0.0
27	Today, many different types of medicines are a...	[High Blood Pressure]	[]	[Blood Pressure Medicines]	0.0	0.0	0.0

Figure 1: Data example

1 Exploratory data analysis

Example of the data set be shown in Fig. 1.

Tables 1, 2, and 3 summarize key aspects of the dataset. Table1 provides a statistics about the rows. Table 2 details on each entity column, and Table 3 offers statistics on the average text length.

Table 4 delves deeper into the relationships between the extracted entities and the source text as well as among the entities themselves. Specifically:

- **Rows with all entities (Conditions/Procedures/Medication) matching in text:** This metric reports the number of rows where every entity (condition, procedure, and medication) is fully matched within the text.
- **Percentage of entity inside another entity:** This indicates the proportion of unique entities that appear as a substring within another entity.
- **Rows with at least one substring entity:** This represents the number of rows in which at least one entity is found as a substring of another (for example, when "Kallmann syndrome" is a substring of "Kallmann syndrome 6").
- **Overall percentage of substring entities among all entity tokens:** This metric reflects the proportion of all entity occurrences that are substrings of another entity, essentially mirroring the previous metric but calculated across every token occurrence.

Metric	**Count**
Total	16406
None empty (At least 1 entity)	13744

Table 1: Row Summary

Column	**Empty Percentage**	**Unique Entities**
Condition	19.6%	3231
Procedure	76.6%	93
Medication	95.63%	17

Table 2: Entity statistics

Entity Type	**Count**	**Avg Text Length**
Total Rows	16406	1303.82
Condition	13191	1482.78
Procedure	3839	1877.48
Medication	717	2733.72

Table 3: Text Length Statistics

Metric	**Value**
Total rows processed	13744
Rows with all entities (Conditions/Procedures/Medication) match in text	13744 (100%)
Percentage of entity inside another entity	34%
Rows with at least one substring entity	3679 (22.42%)
Overall percentage of substring entities among all entities	8.46%

Table 4: Exact Match and Substring Statistics

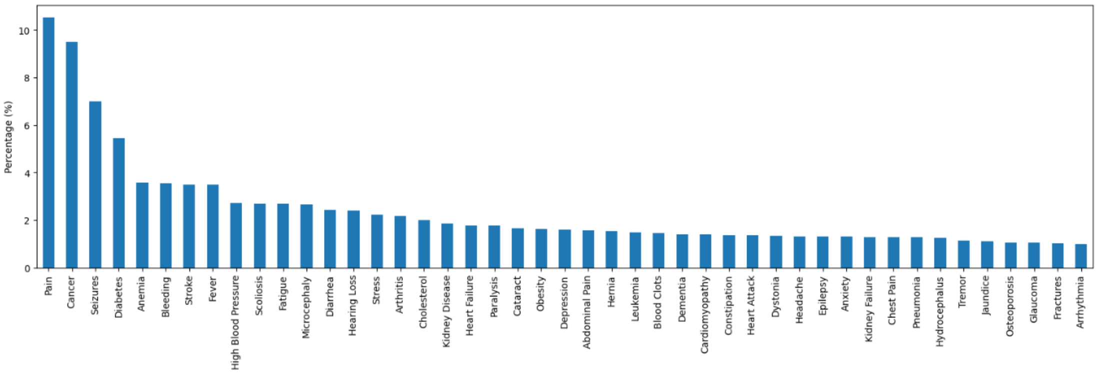


Figure 2: Distribution of condition entities across the dataset's rows.

Figures 2, 3, and 4 display the percentage, relative to the total number of rows, for each entity type (conditions, procedures, and medications, respectively). Figure 5 illustrates the distribution of text lengths, highlighting the variability in the dataset's text content.

2 Data processing

2.1 Span notation

A notable challenge in named entity recognition for this dataset is the occurrence of overlapping entities, where an entity and its substring share intersecting spans. To address this, only the entity with the larger (or higher-order) span is considered. For example, consider the entities **Bone Cancer** and **Cancer**. In the text below:

> "... treating **Cancer** in children ... See Drugs Approved for **Bone Cancer**

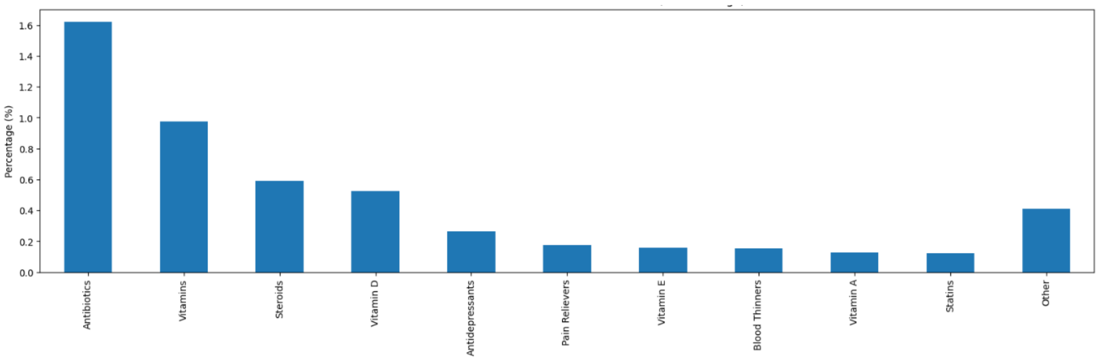


Figure 3: Illustrates the distribution of medication entities across the dataset’s rows

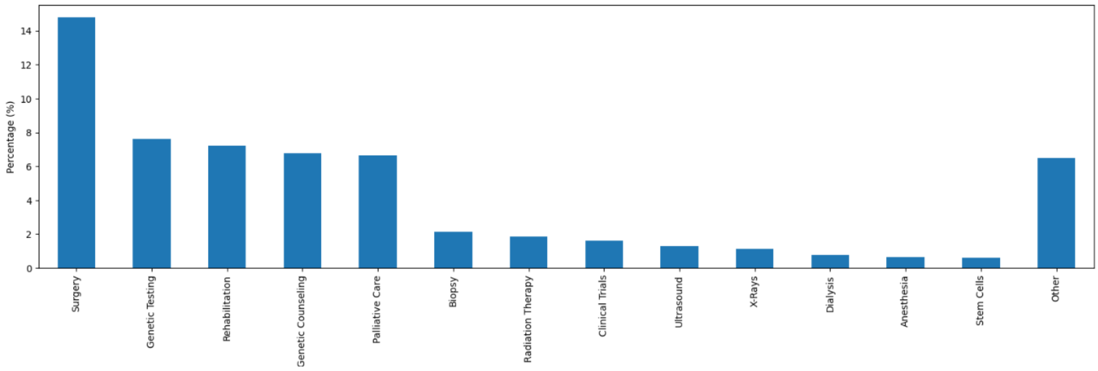


Figure 4: Illustrates the distribution of procedure entities across the dataset’s rows

> for more information...”

only the instance corresponding to **Bone Cancer** is retained, while the substring entity **Cancer** is disregarded when it appears as part of a larger entity.

Entities are annotated with the following labels:

B - Beginning of a multi-token entity.

I - Inside a multi-token entity.

O - Outside any entity.

2.2 Data cleaning

The text is preprocessed with the following steps:

1. Convert all characters to lowercase.
2. Remove special characters and excessive whitespace.

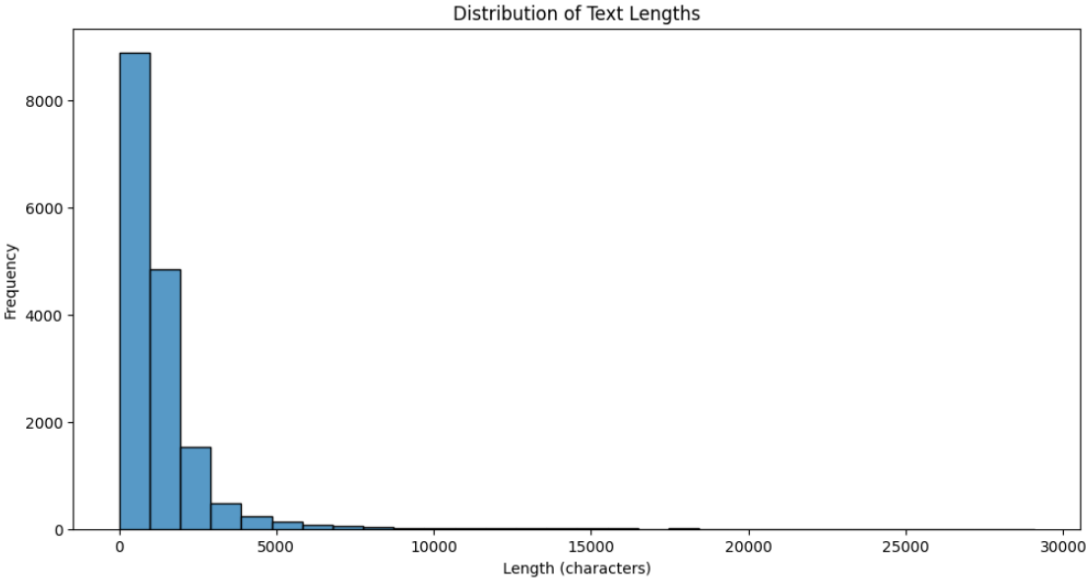


Figure 5: Illustrates the distribution of test length across the dataset's rows

2.3 Dataset Splitting Strategy

The dataset is split into 13333 Train examples and 3073 Test examples using the following approach:

- **Rare Entities:** Entities that appear fewer than three times are treated as rare. These cases are divided into 90% for the training set and 10% for the test set.
- **Common Entities:** A TF-IDF classifier is trained on the common entities dataset using 1,000 features, followed by K-means clustering to form approximately 150 clusters. The common entities dataset is split into 80% for training and 20% for testing.

Subsequently, the test set is formed by consolidating 10% of the rare entities with 20% of the common entities, while the training set comprises the remaining 90% of the rare entities and 80% of the common entities. in Table. 5 statistic on the rare and common text is presented.

Metric	**Value**
Number of rare entities (appearing less than 3 times)	1730
Texts with rare entities	2089 (12.7%)
Texts with common entities only	14317 (87.3%)

Table 5: Rare and Common Entity Statistics

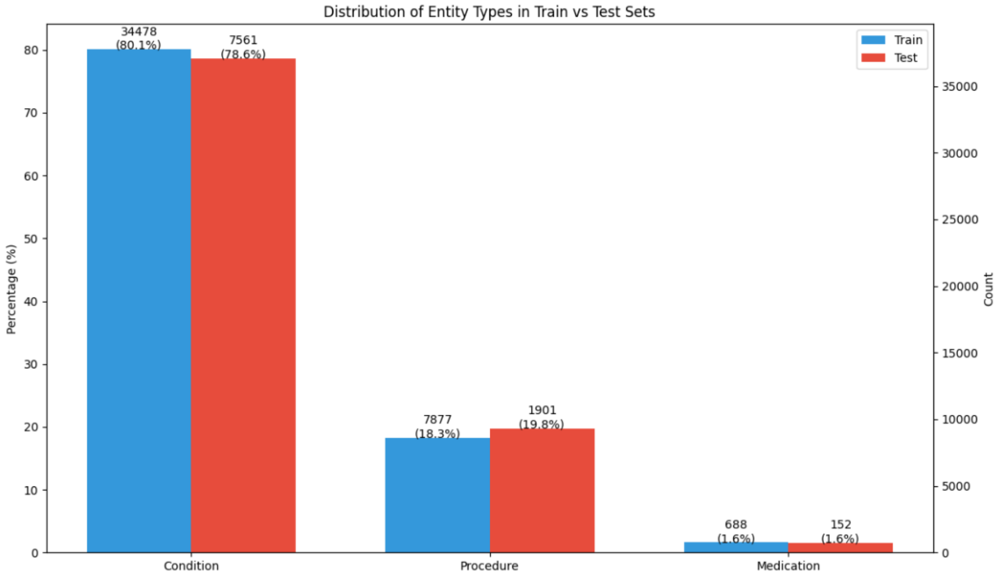


Figure 6: Train-Test Entity Distribution

2.4 Train-Test Statistics

For evaluating the quality of our train-test split (As shown in Fig.6 with per entity label distribution) in medical entity recognition, I developed a comprehensive scoring methodology that assesses both the distribution and overlap of entity types (*Condition*, *Procedure*, and *Medication*) between partitions. These four key metrics were incorporated to calculate a composite balance score for each entity type:

Entity Overlap Ratio (40% weight) The entity overlap ratio measures the proportion of unique entities that appear in both training and test sets:

$$\text{Overlap Ratio} = \frac{|\text{Entities}_{\text{train}} \cap \text{Entities}_{\text{test}}|}{|\text{Entities}_{\text{train}} \cup \text{Entities}_{\text{test}}|} \tag{1}$$

A high overlap ratio ensures the model is evaluated on entity types it has encountered during training, providing a fair assessment of its learning capability. A substantial 40% weight was assigned this metric due to its direct relevance to evaluating the model's generalization to known entities. It score on the different entities label can be seen in Fig.7.

Earth Mover's Distance (20% weight) Also known as the Wasserstein distance, EMD quantifies the dissimilarity between two probability distributions by measuring the minimum "work" required to transform one distribution into another:

$$\text{EMD}(P_{\text{train}}, P_{\text{test}}) = \inf_{\gamma \in \Gamma(P_{\text{train}}, P_{\text{test}})} \text{E}_{(x,y)\sim\gamma}[d(x,y)] \tag{2}$$

Where P_{train} and P_{test} are the entity frequency distributions in training and test sets, respectively. Lower EMD values indicate more similar distributions. It was weighted at

20% because it effectively captures differences in entity frequency distributions, which is crucial for ensuring that frequent and rare entities are proportionally represented in both splits.

Kolmogorov-Smirnov Statistic (10% weight) The KS statistic measures the maximum absolute difference between the cumulative distribution functions of the two samples:

$$\text{KS} = \sup_x |F_{\text{train}}(x) - F_{\text{test}}(x)| \tag{3}$$

Lower values indicate more similar distributions. This metric received a 10% weight as it effectively identifies distribution shifts and complements EMD by focusing on the maximum deviation rather than average distance.

Jensen-Shannon Divergence (10% weight) JS divergence measures the similarity between two probability distributions using a symmetrized version of the Kullback-Leibler divergence:

$$\text{JS}(P_{\text{train}}||P_{\text{test}}) = \frac{1}{2}D_{KL}(P_{\text{train}}||M) + \frac{1}{2}D_{KL}(P_{\text{test}}||M) \tag{4}$$

Where $M = \frac{1}{2}(P_{\text{train}} + P_{\text{test}})$. Lower JS divergence indicates more similar distributions. This metric was assigned a 10% weight as it provides a robust measure of distributional similarity that accounts for both common and rare entities.

To get a s single score, named "Composite Balance Score": each metric was normalized to the [0,1] range and combined them into a composite score:

$$\text{Balance Score} = \frac{(40 \times (1 - \min(\text{EMD}, 1)) + 10 \times (1 - \text{KS}) \times 10 \times (1 - \min(\text{JS}, 1)) + 20 \times \text{Overlap})}{100}$$

This score ranges from 0 to 100%, Fig.8, with higher values indicating better balance between training and test sets. The weights were selected to prioritize both distributional similarity (EMD) and entity coverage (overlap ratio), Fig.7 , with complementary information from KS statistic and JS divergence.

3 Models

3.1 Rule-Based

A baseline model, examines all entities present in the training dataset and searches for corresponding matches in the test dataset. This model is employed solely as a benchmark to evaluate the performance of more sophisticated approaches.

3.2 Conditional Random Field (CRF) Model

The CRF model was trained using a comprehensive set of features designed to capture various aspects of the tokens within the text. These hand-crafted features include:

- The lowercased form of the word (serving as a unique word identifier).
- The part-of-speech tag of the current word.

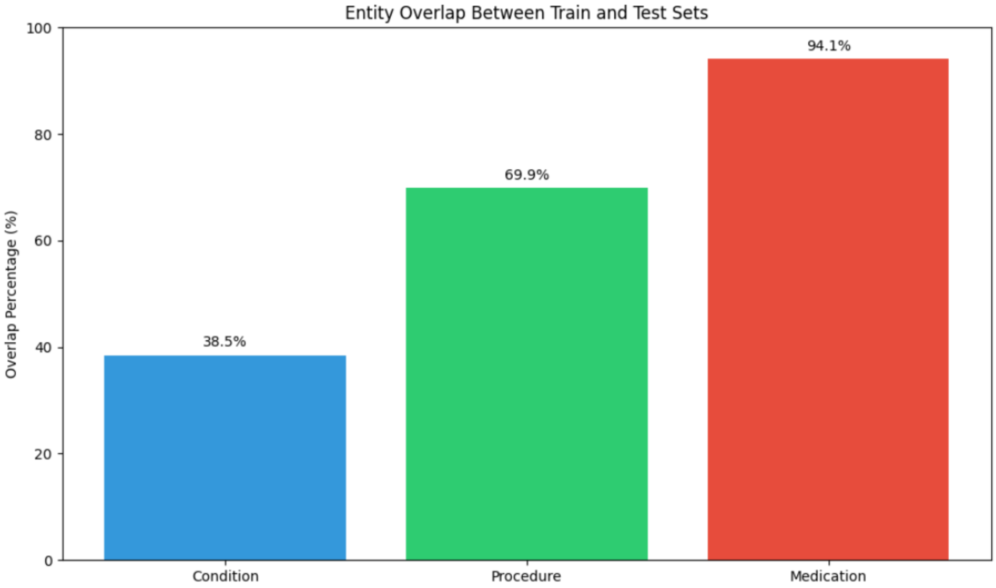


Figure 7: Train-Test Entity Overlap Ratio

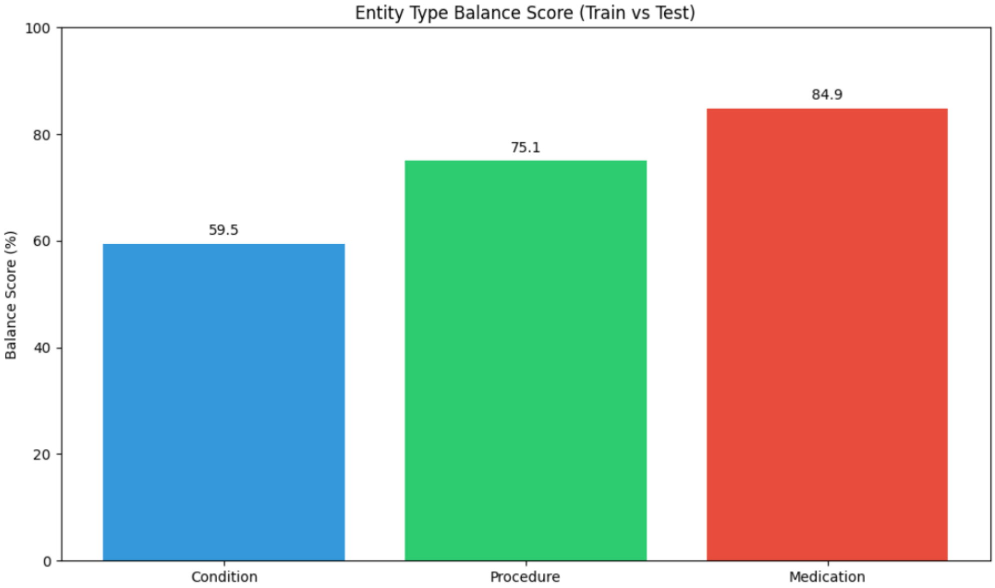


Figure 8: Train-Test balance score across entities

- The last three characters of the word.
- The dependency relation of the word.
- The head word of the dependency relation.

- A boolean flag indicating whether the word is entirely uppercase.
- A boolean flag indicating whether the word represents a digit.
- A boolean flag indicating whether the word starts with a capital letter.

A representative snippet of the feature extraction code is provided below:

```
f'word{0}.lower=' + word_text.lower(),            # serves as word ID
f'word{0}.postag=' + word_pos,                    # PoS tag of current word
f'word{0}[-3:]=' + last_three,                    # last three characters
f'word{0}.dep=' + word_dep,                       # dependency relation
f'word{0}.head=' + head_text,                     # dependency head
f'word{0}.isupper={word_text.isupper()}',         # is the word all uppercase
f'word{0}.isdigit={word_text.isdigit()}',         # is the word a digit
f'word{0}.startsWithCapital={capital}'            # starts with a capital letter
```

ID-CNNs Model

The iterated Dilated Convolutional Neural Networks (ID-CNNs) model, was introduced in *Fast and Accurate Entity Recognition with Iterated Dilated Convolutions* by Strubell et al.[3], was trained for 15 epochs. This model employs repeated dilated convolutions to effectively aggregate context over long sequences, offering a significant speed advantage while achieving accuracy comparable to traditional Bi-LSTM models.

4 Results and Conclusions

Table 6: Performance Comparison of NER Models

Model	**Entity Type**	**Precision**	**Recall**	**F1 Score**
Rule-Based	Condition	0.8585	0.9401	0.8974
	Procedure	0.9841	0.9774	0.9807
	Medication	0.8216	1.0000	0.9021
	Overall	*0.8807*	*0.9484*	*0.9133*
CRF	Condition	0.9555	0.9356	0.9455
	Procedure	0.9944	0.9866	0.9905
	Medication	1.0000	0.9732	0.9864
	Overall	*0.9591*	*0.9418*	*0.9503*
ID-CNNs	Condition	0.9530	0.9356	0.9442
	Procedure	0.9898	0.9827	0.9862
	Medication	0.9957	0.9669	0.9811
	Overall	*0.9617*	*0.9464*	*0.9540*

Comparative analysis of the three Named Entity Recognition (NER) models is shown in Table. 6 and reveals several significant findings:

- **Performance Progression:** We observe a clear progression in model performance from the simple rule-based approach to more sophisticated methods. The overall F1 scores increase from 0.9133 (Rule-Based) to 0.9503 (CRF) and finally to 0.9540 (ID-CNNs), demonstrating the benefits of advanced modeling techniques to learn OOV medical entities.

- **Procedure Entity Performance:** All models demonstrated exceptional performance on the "Procedure" entities, achieving F1 scores ranging from 0.9807 to 0.9905. Since no significant differences were observed in the train-test split or in the EDA statistics, it is plausible that Procedure entities are inherently more detectable than other entity types. This is particularly noteworthy considering that even the Naive baseline model performed remarkably well on these entities.

- **Precision vs. Recall Trade-offs:** The rule-based model demonstrated high recall but lower precision, particularly for Medication entities (recall of 1.0000 but precision of only 0.8216). In contrast, the more advanced models achieved better balance between precision and recall.

- **ID-CNNs Advantages:** The ID-CNNs model demonstrated superior performance by achieving a marginally higher overall F1 score (0.9540) compared to the CRF model. This slight improvement suggests potential benefits of the ID-CNNs architecture for this particular classification task.

In conclusion, while all three models demonstrate viable approaches to medical named entity recognition, the ID-CNNs model offers the best overall performance. The CRF model presents a strong alternative with comparable results.

5 Future Directions

Although the models demonstrate promising overall results, several avenues for further improvement are proposed:

- Integration of an encoder model, such as BERT, with the OOV tokens followed by retraining could enhance the contextual embedding of the full text. Augmenting this approach with a Conditional Random Field (CRF) layer would likely yield additional performance gains. (Initial attempts to implement this model, even with frozen layers and training limited to the few additional OOV tokens, proved computationally demanding to my laptop.)

- Exploration of domain-specific models trained on medical data, such as "medspaCy" [2] or "BioBERT: a pre-trained biomedical language representation model for biomedical text mining" by Jinhyuk Lee et al. [1]. However, thorough investigation would be necessary to ensure these models were not trained on datasets overlapping with the current task.

- Reconsideration of the strict evaluation criteria applied to this medical data. Currently, only exact matches of entity spans that encompass any sub-entities (Only considering "Bone Cancer" when it present along side "Cancer") are considered valid. Implementing more flexible matching constraints might provide a more nuanced evaluation of model performance.

6 Reference

References

[1] Jinhyuk Lee et al. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *CoRR* abs/1901.08746 (2019). arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746.

[2] *medspacy - Library for clinical NLP with spaCy.* URL: https://github.com/medspacy/medspacy.

[3] Emma Strubell et al. “Fast and Accurate Sequence Labeling with Iterated Dilated Convolutions”. In: *CoRR* abs/1702.02098 (2017). arXiv: 1702.02098. URL: http://arxiv.org/abs/1702.02098.