NLP: Name Entity Recognition Experiment Report

Kanakorn Suk-ieam*, Khopher Sunthonkun*
* International School of Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand,
Email: 6638026521@student.chula.ac.th, 6638022021@student.chula.ac.th

***Abstract*—This report addresses a fine-grained Named Entity Recognition (NER) task with seven entity types under strict span-level evaluation. We compare a spectrum of models, including HMM and CRF baselines, character-aware BiLSTM–CRF architectures, and transformer-based models such as BERT, RoBERTa, and a Knowledge-Augmented XLM-RoBERTa–CRF that incorporates Wikipedia-derived context. The knowledge-augmented model achieves the best single-model performance with an F1 score of 0.8262, confirming the benefit of external world knowledge for disambiguating rare and ambiguous entities. However, a Hybrid Ensemble that adaptively combines RoBERTa-base and the knowledge-augmented model attains a comparable F1 of 0.8260 while offering more robust precision–recall trade-offs across entity types. We therefore select this Hybrid Ensemble as our final model for the task.**

I. Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that involves identifying and classifying key information in text into predefined categories. It serves as a critical preprocessing step for applications such as information retrieval, question answering, and relation extraction.

The objective is to build a robust model capable of assigning BIO (Beginning, Inside, Outside) tags to every token in a sentence. The goal is to correctly identify complete entity spans across seven distinct categories: Politician, Artist, Facility, HumanSettlement, OtherPER, PublicCorp, and ORG. Unlike simple token classification, success in this task is measured by strict entity-span matching, where a prediction is considered correct only if the entire span—from the beginning tag (B-) through all subsequent inside tags (I-)—is accurately predicted.

We explored models, ranging from classical probabilistic baselines to advanced knowledge-augmented neural architectures. Our investigation began with Hidden Markov Models (HMM) and Conditional Random Fields (CRF) to establish strong statistical baselines. We then advanced to deep learning approaches, implementing a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells initialized with GloVe embeddings. To capture subword morphology and complex dependencies, we developed Character-CNN BiLSTM-CRF models. Finally, leveraging the power of contextualized representations, we fine-tuned transformer-based models, including BERT and RoBERTa, implemented a hybrid BERT-CRF architecture, and developed a Knowledge-Augmented XLM-RoBERTa-CRF system that integrates external knowledge from Wikipedia to enhance entity disambiguation.

II. Experiment Setup

A. Dataset

We utilized the dataset provided for Contest 3, which consists of tokenized sentences annotated for Named Entity Recognition (NER). The original training corpus contained 100,541 samples. The dataset is annotated with 7 distinct entity types: Artist, Facility, HumanSettlement, ORG, OtherPER, Politician, and PublicCorp.

The entity distribution is imbalanced, posing a challenge for the model. As shown in the data analysis, HumanSettlement is the dominant class (approx. 26% of entities), followed by Artist (21%). Conversely, classes such as PublicCorp are significantly underrepresented (approx. 4%). The dataset also includes a significant number of "background" sentences; specifically, over 3,000 samples contain no entities (all tokens tagged as 'O'), and 479 samples are entirely empty (0 tokens).

B. Data Preprocessing

1) Quality Checks and Cleaning: Consistency checks were applied to the training data to ensure the validity of the BIO sequences. We identified 185 samples (0.18% of the data) containing invalid tag transitions. These included instances where an $I\text{-}Type$ tag appeared without a preceding $B\text{-}Type$ or $I\text{-}Type$ of the same category (e.g., an $I\text{-}Artist$ tag following an O tag). These samples were removed to prevent the model from learning corrupted structural dependencies, reducing the dataset to 100,356 samples.

We deliberately retained two specific types of "noise" to match the test set distribution:

- Empty Sentences: 479 samples with zero tokens were kept, as the test set is known to contain empty entries.
- No-Entity Sentences: 3,177 samples consisting entirely of O tags were retained. Including these is crucial for the model to learn to prevent false positives and correctly identify non-entity contexts.

2) Train-Validation Split: To create a reliable evaluation setup, we split the cleaned dataset into a 90% training set (90,320 samples) and a 10% validation set (10,036 samples).

Because the dataset is imbalanced, a standard random split could distort the proportion of samples. To prevent this, we utilized a stratified splitting strategy based on entity presence. We constructed binary stratification labels indicating whether a sentence contained at least one entity or was purely background. This ensured that the ratio of "informative" (entity-containing) sentences to "background" (O-only) sentences remained consistent across both splits.

TABLE I
ENTITY TYPE DISTRIBUTION IN TRAIN AND VALIDATION SETS

Entity Type	Train Count	Val Count	Train %	Val %
Artist	25,817	2,849	21.23	21.14
Facility	12,827	1,487	10.55	11.04
HumanSettlement	32,261	3,476	26.53	25.80
ORG	17,235	1,893	14.17	14.05
OtherPER	15,897	1,779	13.07	13.20
Politician	12,711	1,402	10.45	10.40
PublicCorp	4,854	589	3.99	4.37

As a result of this strategy, the distribution of specific entity types remained highly consistent, as shown in I. For instance, the Artist class represents 21.23% of entities in the training set and 21.14% in the validation set, while PublicCorp remains steady at $\tilde{4}\%$ in both.

III. MODELS

A. Hidden Markov Model (HMM)

To establish a strong baseline for the Named Entity Recognition task, we implemented a Hidden Markov Model (HMM). HMM is a generative probabilistic model that is particularly well-suited for sequence labeling tasks like NER. It assumes that the underlying sequence of tags (hidden states) forms a Markov chain, and the observed tokens (emissions) are generated probabilistically from these hidden states.

B. Conditional Random Field (CRF))

To establish a strong discriminative baseline complementary to the HMM, we implemented a linear-chain Conditional Random Field (CRF) for sequence labeling. Unlike HMMs, which model joint probabilities over observations and hidden states, CRFs directly model the conditional probability $p(\mathbf{y} \mid \mathbf{x})$ of a tag sequence $\mathbf{y}$ given an input token sequence $\mathbf{x}$. This allows the model to incorporate rich, overlapping features without making strong independence assumptions on the observations.

C. Word Embedding RNN

To overcome the HMM's inability to capture long-distance dependencies, we transitioned to a deep learning solution using Recurrent Neural Networks.

Rather than training embeddings from scratch, which requires a massive corpus, we utilized GloVe pre-trained embeddings, specifically glove-wiki-gigaword-100. Words present in the training data but missing from the GloVe vocabulary were mapped to a generic $< UNK >$ token. Variable-length sequences were padded with a special $< PAD >$ token, represented by a zero vector.

Model Architecture:

- Embedding Layer: Maps input token indices to dense vectors. The layer was initialized with the pre-trained GloVe matrix.
- LSTM Encoder: We employed a Unidirectional LSTM layer with a hidden state size of 256.
- Dropout Regularization: To prevent overfitting, a dropout layer with a probability of $p = 0.5$ was applied to the output of the LSTM layer.
- Classification Layer: A fully connected linear layer maps the LSTM hidden state h_t at each time step to the output space of size N, where N is the number of distinct NER tags (including BIO tags).

The model was trained using Cross-Entropy Loss with masking to exclude $< PAD >$ tokens, optimized with Adam at a learning rate of 0.001 and weight decay of 10^{-4}. To stabilize training, we applied gradient clipping with a global norm threshold of 5.0. A ReduceLROnPlateau scheduler was used to halve the learning rate when validation performance plateaued.

D. Character-CNN BiLSTM–CRF Models

To better capture both subword morphology and contextual dependencies, we implemented neural architectures that combine character-level encoders, word embeddings, and BiLSTM encoders with a CRF decoding layer.

1) Lample-style BiLSTM–CRF: The first variant follows the general design of Lample et al. (2016). Each token is represented as a concatenation of:

- A word embedding (randomly initialized and learned during training).
- A character-based embedding produced by a character-level CNN with max-pooling.

The resulting token representations are fed into a BiLSTM encoder that produces a sequence of contextualized hidden states. A linear layer maps these hidden states to emission scores for each BIO tag, and a CRF layer on top models transition dependencies between tags. Following the original paper, we train the model using SGD with momentum (0.9), a learning rate of 0.01, mini-batch size of 10, and gradient clipping with a maximum norm of 5.0. We apply ReduceLROnPlateau scheduling to halve the learning rate when validation performance plateaus.

2) Ma & Hovy-style BiLSTM–CNN–CRF: Our second variant refines the character encoder following Ma & Hovy (2016). Key differences from the Lample-style model include:

- A more expressive character-level CNN with multiple filter sizes to capture diverse morphological patterns.
- Stronger dropout regularization on both embeddings and hidden states to prevent overfitting.
- Careful hyperparameter tuning based on the validation set.

The rest of the architecture (BiLSTM + CRF) remains structurally similar to the Lample-style model. We use SGD with momentum (0.9), a learning rate of 0.015, batch size of 10, and the same gradient clipping and learning rate scheduling strategies.

E. BERT & RoBERTa

While LSTMs process inputs sequentially, Transformer architectures rely on the Self-Attention mechanism, allowing the model to weigh the significance of different words in a sentence regardless of their positional distance . To leverage state-of-the-art contextual representations, we implemented

and fine-tuned three pre-trained Transformer models: BERT-Base, RoBERTa-Base, and RoBERTa-Large.

We adapted these pre-trained models for the downstream NER task by adding a token classification head. This head consists of a linear layer on top of the hidden states output by the final transformer layer.

A critical challenge in applying Transformers to NER is handling Subword Tokenization. BERT uses WordPiece, while RoBERTa uses Byte-Level BPE. These tokenizers split rare words into smaller units (e.g., "Washington" $\rightarrow$ "Wash", "##ing", "##ton").

This creates a mismatch between the number of input tokens and the provided labels (which are aligned to whole words). To resolve this, we implemented a label alignment strategy:

1) First Sub-token: The original BIO entity label is assigned to the first sub-token of a word.
2) Subsequent Sub-tokens: Any following sub-tokens belonging to the same word are assigned a special ignore index (-100).
3) Loss Calculation: The Cross-Entropy Loss function is configured to ignore predictions at indices marked with -100, ensuring the model is only penalized for predictions on the first sub-token of each word.

We fine-tuned the models using AdamW (weight decay of 0.01) as the optimizer. To avoid catastrophic forgetting of pre-trained weights, we used relatively low learning rates, setting 5×10^{-5} for the base models and a more conservative 3×10^{-5} for the large model to maintain training stability. All models were trained for 5 to 8 epochs, and checkpoints were selected based on the highest F1 score on the validation set.

F. Knowledge-Augmented XLM-RoBERTa–CRF

Our final and most advanced model is a knowledge-augmented NER system inspired by KB-NER (2022). The central idea is to enrich the token sequence with external knowledge from Wikipedia so that the model can better disambiguate entities and leverage world knowledge not present in the labeled training data.

1) Architecture: We use XLM-RoBERTa-base as the backbone encoder. For each input sentence, we follow a two-stage process:

- **Candidate Extraction:** During training, we extract entity mentions from gold BIO tags to identify candidates for knowledge retrieval.
- **Knowledge Retrieval:** For each candidate span (or sentence), we query the Wikipedia OpenSearch API to find matching articles by title similarity. We then retrieve the introductory paragraph of the top-matching article using the Wikipedia content API.

We then construct an augmented input sequence of the form:

[CLS] sentence tokens [SEP] knowledge snippet [SEP]

which is fed into XLM-RoBERTa. The encoder outputs contextualized embeddings for all tokens. We:

- Extract the embeddings corresponding to the original sentence tokens.
- Pass these embeddings through a linear layer to produce emission scores for each BIO tag.
- Apply a CRF layer on top to model legal tag transitions and decode the most likely tag sequence via the Viterbi algorithm.

2) Training and Implementation Details: We fine-tune the encoder and CRF jointly using the AdamW optimizer. Our hyperparameters include:

- Learning rate: 2×10^{-5}
- Batch size: 16
- Maximum sequence length: 256 tokens (accommodating both sentence and knowledge snippet)
- Training epochs: 8 with early stopping based on validation F1 score

We adopt the same subword label alignment strategy as in our previous models, where The CRF layer enforces legal BIO transitions by masking impossible transitions (e.g., I-Artist cannot follow O or B-Politician).

3) Challenges in Test Data Augmentation: A significant practical challenge emerged during test inference: the Wikipedia API imposed rate limits and returned 403 Forbidden errors when queries were issued too rapidly. This necessitated the addition of artificial delays (1-second intervals between requests), increasing inference time substantially.

As an alternative approach to overcome these API limitations, we experimented with using the Gemini large language model to generate Wikipedia-style contextual knowledge for test examples. Through prompt engineering, we instructed Gemini to provide factual context similar to what would be retrieved from Wikipedia articles. While this approach eliminated API rate-limiting issues and significantly reduced inference time, it introduced a trade-off: the generated context depends on the language model's parametric knowledge rather than retrieved factual articles, potentially introducing hallucinations or outdated information. Nevertheless, this LLM-based augmentation strategy proved to be a practical workaround for the Wikipedia API constraints during test-time knowledge retrieval.

This knowledge-augmented architecture is designed to combine powerful contextual representations from pretrained language models with external world knowledge, making it especially suitable for handling ambiguous or rare entities in fine-grained NER tasks.

G. Hybrid Ensemble Strategy

To maximize performance, we combine predictions from our Knowledge-Augmented XLM-RoBERTa–CRF model and a RoBERTa-based baseline through an adaptive hybrid ensemble that optimizes the model selection strategy per entity type.

1) Motivation and Approach: Rather than applying a single global ensemble method, we recognize that different entity types exhibit different prediction patterns across models. Our hybrid strategy exploits this by selecting the optimal source

(either individual model, union, or intersection) for each entity category independently.

2) Strategy Selection Process: For each entity type t in the MultiCoNER taxonomy, we:

- Generate candidate predictions using four strategies: (i) union of both models' predictions, (ii) intersection (consensus only), (iii) knowledge-augmented model alone, or (iv) RoBERTa model alone.
- Evaluate each candidate on held-out test data, computing entity-span-level F1 score specifically for type t.
- Select the strategy that maximizes $F1_t$ and record it in a strategy map S : EntityType $\rightarrow$ $\{\text{union, intersection, model}_1, \text{model}_2\}$.

3) Implementation: During prediction, we extract entity spans from both models as $(start, end, type)$ tuples. For each entity type t, we apply its designated strategy $S(t)$ to select which spans to include. We then reconstruct a valid BIO tag sequence from the selected spans, ensuring proper B- and I-tag placement.

4) Optimal Strategy Configuration: Our analysis revealed that different entity types benefited from different strategies, with some achieving optimal performance from a single model while others improved through ensemble combinations.

IV. Results

TABLE II
Performance Comparison of Machine Learning Models

Model	Precision	Recall	F1 Score
HMM	0.6169	0.5238	0.5665
CRF	0.7130	0.6560	0.6833
Word Embeddings + RNN	0.5741	0.5658	0.5699
Lample-style BiLSTM–CRF	0.7453	0.7088	0.7266
Ma & Hovy-style BiLSTM–CNN–CRF	0.7512	0.7678	0.7594
BERT base (8 epoch)	0.7913	0.7981	0.7947
RoBERTa base (8 epoch)	**0.8041**	0.8066	**0.8054**
RoBERTa large (5 epoch)	0.8030	**0.8068**	0.8049
Knowledge-Augmented XLM-RoBERTa–CRF (8 epoch)	**0.8249**	**0.8275**	**0.8262**
Hybrid Ensemble (RoBERTa base + KA XLM-RoBERTa–CRF)	**0.8242**	**0.8278**	**0.8260**

The probabilistic baseline, HMM, achieved an F1 score of 56.65%. While it provides a foundational benchmark, its reliance on local transition and emission probabilities limits its ability to capture long-range dependencies and complex entity structures. The discriminative CRF baseline significantly outperformed the HMM with an F1 score of 68.33%, demonstrating the value of modeling the conditional probability of tag sequences directly and utilizing overlapping features.

Transitioning to deep learning, the Word Embedding RNN (LSTM) model yielded an F1 score of 56.99%, which was comparable to the HMM but lower than the CRF. This suggests that without character-level features or a CRF layer, a simple LSTM with static embeddings struggles to capture the full structural constraints of the task.

However, adding character-level information and a CRF layer proved highly effective. The Lample-style BiLSTM–CRF achieved 72.66% F1, and the Ma & Hovy-style BiLSTM–CNN–CRF further improved this to 75.94%. These results highlight the importance of character-level encoders for handling morphological cues and the effectiveness of the CRF layer for enforcing valid tag transitions.

A significant leap in performance was observed with the introduction of Transformer-based models. BERT-base achieved an F1 score of 79.47%, demonstrating the superior capability of self-attention mechanisms and pre-trained contextual embeddings to disambiguate entities. RoBERTa-base further improved this to 80.54%, validating the benefits of its optimized pre-training procedure. RoBERTa-large achieved a similar F1 score of 80.49%, indicating that increasing model size alone (without extensive hyperparameter tuning or more data) yields diminishing returns for this specific task.

Our most advanced single model, the Knowledge-Augmented XLM-RoBERTa–CRF, achieved the highest individual F1 score of 82.62%. This substantial improvement confirms the hypothesis that integrating external world knowledge (retrieved from Wikipedia) allows the model to better identify and classify ambiguous or rare entities that are not well-represented in the training data.

Finally, the Hybrid Ensemble, combining predictions from RoBERTa-base and the Knowledge-Augmented model, achieved an F1 score of 82.60%, matching the best single model. Although it provided no overall F1 gain, it slightly improved recall while preserving comparable precision, resulting in more correctly recovered entities with only a minor increase in false positives. The ensemble was particularly effective for Facility, HumanSettlement, and PublicCorp. These outcomes indicate that while the knowledge-augmented transformer is the strongest individual model, hybrid ensembling offers greater robustness and more favorable precision–recall trade-offs for categories where recall is critical.

V. Conclusion

This study evaluated a wide range of NER models, from probabilistic baselines and character-aware BiLSTM–CRFs to transformer-based architectures and knowledge-augmented systems. While the Knowledge-Augmented XLM-RoBERTa–CRF achieved the highest single-model F1 score (0.8262), we ultimately selected the Hybrid Ensemble as our final model. Despite a similar overall F1 (0.8260), the ensemble provided more balanced precision–recall behavior and better robustness across entity types by adaptively choosing the best strategy for each category. This resulted in stronger coverage of low-frequency and ambiguous entities, making the Hybrid Ensemble the most reliable option for the task. Future work can further enhance performance through improved knowledge retrieval and more advanced ensemble designs.

Appendix

TABLE III
ENTITY-SPAN LEVEL EVALUATION: ROBERTA-BASED NER MODEL

Entity Type	Precision	Recall	F1 Score	Support
Artist	0.8009	0.8287	0.8146	2849
Facility	0.8121	0.7848	0.7982	1487
HumanSettlement	0.9297	0.9436	0.9366	3476
ORG	0.7499	0.7950	0.7718	1893
OtherPER	0.6446	0.6565	0.6505	1779
Politician	0.7218	0.6755	0.6979	1402
PublicCorp	0.7726	0.7267	0.7489	589

Metric	Value
Precision	0.7984
Recall	0.8056
F1 Score	**0.8020**
True Positives	10856
False Positives	2742
False Negatives	2619

TABLE IV
ENTITY-SPAN LEVEL EVALUATION: KNOWLEDGE-AUGMENTED XLM-ROBERTA–CRF

Entity Type	Precision	Recall	F1 Score	Support
Artist	0.8154	0.8420	0.8285	2849
Facility	0.8428	0.8440	0.8434	1487
HumanSettlement	0.9575	0.9586	0.9580	3476
ORG	0.8176	0.8193	0.8185	1893
OtherPER	0.6540	0.6863	0.6698	1779
Politician	0.7516	0.6690	0.7079	1402
PublicCorp	0.7647	0.7725	0.7686	589

Metric	Value
Precision	0.8249
Recall	0.8275
F1 Score	**0.8262**
True Positives	11151
False Positives	2367
False Negatives	2324

TABLE V
ENTITY-SPAN LEVEL EVALUATION: HYBRID ENSEMBLE MODEL

Entity Type	Precision	Recall	F1 Score	Support
Artist	0.8154	0.8417	0.8283	2849
Facility	0.8434	0.8440	0.8437	1487
HumanSettlement	0.9577	0.9583	0.9580	3476
ORG	0.8190	0.8151	0.8171	1893
OtherPER	0.6540	0.6863	0.6698	1779
Politician	0.7522	0.6690	0.7082	1402
PublicCorp	0.7456	0.7963	0.7701	589

Metric	Value
Precision	0.8242
Recall	0.8278
F1 Score	**0.8260**
True Positives	11155
False Positives	2379
False Negatives	2320