MarSan at SemEval-2022 Task 11: Multilingual complex named entity recognition using T5 and transformer encoder

Ehsan Tavan[1,*], **Maryam Najafi**[1,*]
[1] NLP Department, Part AI Research Center, Tehran, Iran
{ehsan.tavan, maryam.najafi}@partdp.ai

Abstract

The multilingual complex named entity recognition task of SemEval2020 required participants to detect semantically ambiguous and complex entities in 11 languages. In order to participate in this competition, a deep learning model is being used with the T5 text-to-text language model and its multilingual version, MT5, along with the transformer's encoder module. The subtoken check has also been introduced, resulting in a 4% increase in the model F1-score in English. We also examined the use of the BPEmb model for converting input tokens to representation vectors in this research. A performance evaluation of the proposed entity detection model is presented at the end of this paper. Six different scenarios were defined, and the proposed model was evaluated in each scenario within the English development set. Our model is also evaluated in other languages.

1 Introduction

Named Entity Recognition (NER) is a key component of Natural Language Processing (NLP) assigned to identify regions of text that contain references to entities. It is the process of identifying the informative part of data or applicable labels from unstructured data. In NER, data is gathered from unstructured data such as emails, blogs, newspapers, tweets, etc., to extract meaningful information.

To put it another way, the term NER refers to identifying token spans of entities mentioned in the text and classifying them into a set of predetermined categories. The system finds entities from unstructured data and organizes them into multiple categories. As an extension of NLP, the field of NER can be considered as Information Extraction (IE).

For NLP, IEs are among the trending fields and play an essential role in the following tasks: Find and understand limited relevant parts of texts, Gather information from many pieces of text, and Produce the unified representation of all the relevant information. The NER problem falls into a general class of NLP problems called sequence tagging. Part of Speech (POS) tagging and chunking are sequence tagging NLP tasks in addition to NER. It is possible to detect NER in three ways: flat NER, nested NER, and discontinuous NER. Nested NER has overlapping in the span of text Yan et al. (2021). Most approaches only target flat entities, ignoring nested structures common in many scenarios. It is challenging to identify spans as well as types of named entities in text Lample et al. (2016). So to find the best architecture, we implement a transformer-based language model in this research. NLP has significantly benefited from transfer learning in recent years. Transfer learning gains power and effectiveness from pre-training on large, unlabeled text datasets. The model can then be fine-tuned to a smaller labeled dataset, resulting in better performance. Many models have achieved success in this field, including the Text-To-Text Transfer Transformer (T5) Raffel et al. (2019). The T5 is a pre-trained encoder-decoder language model that employs the "text-to-text" format to accomplish all types of NLP work, including generation, translation, and summarization tasks.

In SemEval-2022 task 11 Malmasi et al. (2022b), a multilingual complex NER is provided. Languages presented in Malmasi et al. (2022a) are Bangla, German, English, Spanish, Farsi, Hindi, Korean, Dutch, Russian, Turkish, and Chinese. Since this dataset contains words from different languages, it is challenging to choose an appropriate word representation for converting them into their corresponding vectors. The multilingual nature of this task necessitated the selection of the multilingual variant of Google's T5 model named MT5 Xue et al. (2020) that had already been trained on a database of more than 101 languages and con-

[*] Equal contribution. Listing order is random.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1639 - 1647

tained up to 13 billion parameters. In this paper, MT5 is used as the main Embedding.

We evaluated the proposed model using the English test set and achieved the F1-score of 71.45% as part of this competition. In our next step, we considered this model in other subtasks, and our rank varied from 9 to 21 depending on which subtask we evaluated. Our code is available at GitHub[1] for researchers.

The contributions of this paper are summarized as follows. Section 2 introduces previous attempts in the NER. In Section 3, information about the task and datasets is presented. We then offer a deep learning framework for recognizing named entities in Section 4. Section 5 details the experimental setup, while Section 6 presents the results of the experiments. Section 7 presents both quantitative and qualitative error analysis. We conclude our paper in Section 8.

2 Background

The NER field has undergone enormous changes in recent years Meng et al. (2021) Fetahu et al. (2021). As mentioned before, NER is the process of identifying relevant objects such as persons, products, genes, place, organization, etc., that are mentioned in the string of the text, sentence, or paragraph. NER typically forms the basis of other tasks such as event detection from news, online shopping customer service, knowledge graph construction, and biological analysis Bokharaeian et al. (2017).

Yu et al. (2020) stated that since NER tags are nested, it uses the graph-based dependency parsing method and examines eight separate corpora to achieve State-of-the-art for all. The embedding layer in this article was composed of BERT, fasttext, and character embedding. There is widespread usage of CRF in the field of NER. CRF is used for the first time in Collobert et al. (2011) for the NER. The representation of the sample in this research was obtained by Convolutional Neural Network(CNN). After that, many articles used CRF in various languages and combined it with other methods, such as Part Of Speech Tagging(POS), Long Short-Term Memory(LSTM), Embeddings from Language Models(ELMo), etc.Lample et al. (2016); Alves-Pinto et al. (2022); Rajan and Salgaonkar (2022); Ma and Hovy (2016); Huang et al. (2015). Peters et al. (2018) which is known as ELMo, extends LSTM-CRF and leverages pretrained word-level language models for better context-aware representations. Peters et al. (2018) focuses on introducing and defining a Bidirectional language model that is tested on numerous NLP tasks in 2018. The accuracy of this research of NER with the language model and CRF layer was 93.42%.

To resolve ambiguity in NER tags, Straková et al. (2019) encoded nested entities in a sequence (seq2seq) and prepared an LSTM-CRF-based model. By incorporating pre-trained character-level language models into Flair Akbik et al. (2019), researcher presented contextualized representations. The following year, Akbik et al. extended this model to incorporate dataset-level word embeddings, dynamically aggregating embeddings and implementing pooling to extract a global word representation from all instances Akbik et al. (2018).

Word representations and character representations are used in Ma and Hovy (2016), an end-to-end system designed using LSTM, CNN, and CRF. This idea has been implemented on the Penn Treebank WSJ corpus Marcus et al. (1999) for POS and CoNLL 2003 Sang and De Meulder (2003) for NER datasets. A CNN has been used in this research to extract the character-level representation of words, with its output then being input into LSTMs, followed by a CRF layer. There is another extension to CRF known as hybrid semi-Markov conditional random fields (HSCRFs) is explained in Ye and Ling (2018) by contributing word-level labels in the building of SCRFs Lafferty et al. (2001). The purpose of using word-level tags to derive segment scores is to obtain segment scores.

To solve different sequence tagging models with a CRF inference, Yang and Zhang (2018) implemented NCRF++ with three steps: a character sequence layer, a word sequence layer, and an inference layer. RodrigoAgerri et al. presented a multilingual NER system that combines many features to cluster based on local information Agerri and Rigau (2016). Results from standard task evaluation data such as CoNLL for English, Spanish, and Dutch were reposted.

According to Wang et al. (2020), contextualized language models can produce better results when different embeddings are combined. This paper presents a framework for generating and scoring the output of embedded combinations using rein-

[1] https://github.com/MarSanTeam/Complex_NER_SemEval

forcement learning, which achieves the best accuracy in 6 different fields and 21 large datasets. According to Wang et al. (2020), contextualized language models can produce better results when different embeddings are combined. By rewarding model scores for better concatenations of embeddings, can propose Automated Concatenation of Embeddings to find better concatenations of embeddings for structured prediction tasks.

Majumder et al. (2022) considers the issue of informal data, whose unstructured and incomplete nature makes the process more challenging. To solve the mentioned challenge, Bi-LSTM based architecture for informal tweets in Hindi and English was implemented. There is more than one way to do it in this field. Finding the part of the text containing entity information, classifying and identifying the right entity, and applying that entity to the appropriate part of the text are just some ways. Generating tags is one other way to implement them. As mentioned in Yan et al. (2021), Hang Yan et al. propose that they use a novel and simple Bidirectional Auto-Regressive Transformer(BART) sequence-to-sequence (Seq2Seq) framework that uses a pointer mechanism Vinyals et al. (2015) to generate the entity sequence directly.

Another approach in NER, which is mentioned in Islam et al. (2022), consists of using an attention mechanism to minimize the problem of detecting redundant and inessential data and ignoring them entirely. Combining semantic, glyph, and phonetic features to improve the expression ability of Chinese character embedding, Li and Meng (2021) proposes an architecture based on Fusion Embedding for the Chinese language.

There is also a paper for the Chinese language entitled Jia et al. (2020) that identifies entities from Chinese social media texts, using uncertain information from word segmentation. Researchers have proposed that interactions between spans of tokens can help determine discontinuous mentions and have developed a transition-based model with a generic neural encoding to be able to detect discontinuous mentions Khan et al. (2020).

A model of bidirectional transformers is presented in Yamada et al. (2020), which produces a contextualized representation of words and tokens. BERT's masked language model is used as pretrained word embeddings. As a result, Yamada et al. present advancement in attention known as entity-aware self-attention mechanisms and achieve state-of-the-art in five benchmarks that include: Open Entity (entity typing), TACRED (relation classification), CoNLL-2003 (NER), ReCoRD (cloze-style question answering), and SQuAD 1.1 (extractive question answering). Since carelessness or a lack of background knowledge of annotators might lead to model performance errors, some research has been conducted to identify and solve these issues. Wang et al. (2019) provides a framework for finding human errors in NER annotations. Following the correction of labels in the test set, they re-evaluated state models in NER, claiming that the results were more accurate than the original test set. This paper's main idea is cross weighs, which accommodates label mistakes during training and then trains a more robust NER model.

Wang et al. (2021) finds the external context of input sentences by retrieving relevant sentences. The process of selecting the top similar text involves re-ranking retrieved samples according to their semantic relevance to the input sentence. As a result, the inputs are the concatenation of input sentences and external contexts. Both input types are used to implement Cooperative Learning (CL), and different representations are encouraged to produce similar contextual representations or output label distributions. Results on eight other NER datasets achieve state-of-the-art results. But one drawback of this method is that there are no document-level contexts in practice.

3 Task Description

Our investigation aims to comparatively study MultiCoNER-2022 datasets that have been considered individually for complex NER systems in 11 languages. Under short and low-context settings, the task detects semantically ambiguous and complex entities.

Languages presented in MultiCoNER-2022 are: English, Spanish, Dutch, Russian, Turkish, Korean, Farsi, German, Chinese, Hindi, and Bangla. The number of instances for each language varies between 150k to 500k. There are 15300 train data and 800 validation data for each language.

This dataset is labeled with the following tags: PER: Person, LOC: Location, GRP: Group, CORP: Corporation, PROD: Product, CW: Creative Work. There is detailed information of datasets illustrated in Table 1.

The datasets used in this task include sentences labeled with the IOB format. Using this format,

Languages	LOC		PER		PROD		GRP		CW		CORP	
	Span	Token	Span	Token	Span	Token	Span	Token	Span	Token	Span	Token
English	4799	7550	5397	11538	2923	4723	3571	10038	3752	9782	3111	6222
Spanish	4968	7204	4706	9999	3040	4404	3226	7993	3690	8734	2898	5208
Dutch	5529	6767	4408	9369	2935	3572	3306	7204	3340	7142	2813	4544
Russian	5529	6767	4408	9369	2935	3572	3306	7204	3340	7142	2813	1731
Turkish	5804	6862	4414	8446	3184	4392	3568	6649	3574	7715	2761	4420
Korean	6299	6837	4536	7171	3082	4165	3530	5525	3883	3665	3313	1370
Farsi	5683	8720	4272	8613	2955	4496	3199	7676	3694	7528	2991	5382
German	4778	6566	5288	11230	2961	3898	3509	5878	3507	9054	3083	6210
Chinese	6986	28762	2225	14048	4854	16084	713	3200	5248	18817	3805	18069
Hindi	2614	4218	2418	5254	3077	2295	2843	8664	2304	5896	2700	5617
Bangla	2351	3804	2606	5738	3188	5152	2405	6653	2157	5001	2598	5299
Code-Mixed	325	493	296	680	316	560	248	677	298	755	294	653

Table 1: Distribution of spans and tokens for each entity

tokens that are not a part of an entity are tagged as 'O', the first token of an entity is represented by the 'B' tag, while the rest of the entity's tokens are represented by an 'I' tag. The entity category precedes both a hyphen and the "B" and "I" tags. Therefore, NER is a task that labels tokens according to their text, which is multi-class token classification.

4 System overview

In this section, we will introduce our proposed NER framework. The proposed framework consists of three parts:

1. Word Representation Module
2. Feature extraction Module
3. Prediction Module

The proposed architecture takes the token sequence $S = \{s_1, s_2, s_3, ..., s_n\}$, and predicts entity sequence $O = \{o_1, o_2, o_3, ..., o_n\}$ as output. Figure 1 is an illustration of the proposed architecture.

4.1 Word Representation Module

In light of the multilingual nature of the SemEval NER task, the T5 language model was applied to convert tokens into their representation vector. In the word representation module, the last hidden state of the T5-large encoder is chosen to learn the k-dimensional (1024 here) representation for input tokens. The T5 uses SentencePiece encoding and assigns named entity tags to its extracted tokens. Tokens extracted from T5 are always equal to or greater than main tokens. Each token thus becomes one or more subtokens. The label of the first subtoken corresponds to the label of the first token, while the other subtokens have the label X.

Subtoken Check A key consideration is that each token becomes one or more subtokens. To provide better training, a feature called subtoken check is used. This feature checks whether the input token is tokenized into the subtoken or not. After tokenizing the tokens, the first subtoken of each token has a value of 1, and the rest have a value of 0. Hence, the T5 encoder takes two input sequences of the same length; one is the subtoken index, and the other is the subtoken check index. After adding this feature and improving the results, it was found that in token-based tasks such as NER, the existence of this feature is extremely helpful for managing subtokens.

Byte-pair Embeddings The Byte-pair Embeddings(BPEmb) Heinzerling and Strube (2017) consists of pre-trained subword embeddings in 275 languages. BPEmb is a variable-length encoding that views the text as a sequence of symbols, iteratively merging the pair with the highest frequency into a new symbol. It provides a mechanism for properly tokenizing input sequences so that unknown tokens can prepare appropriate representations by using subtokens. To predict the named entity tag for the input sequences, we concatenate the output vector of the BPEmb model with the output vector of the feature extraction layer since fine-tuning of the model is not possible during training.

4.2 Feature Extraction Layer

Since the goal of NER is to predict the entity label of each token, an awareness of the semantic dependencies between tokens can be extremely helpful. The which uses the multi-encoder architecture Vaswani et al. (2017), which uses the multi-head self-attention mechanism, is one of the most suitable deep learning architectures for extracting relation between tokens. There are two sublayers in this encoder module. The first sublayer is a multi-head self-attention mechanism, while the second is

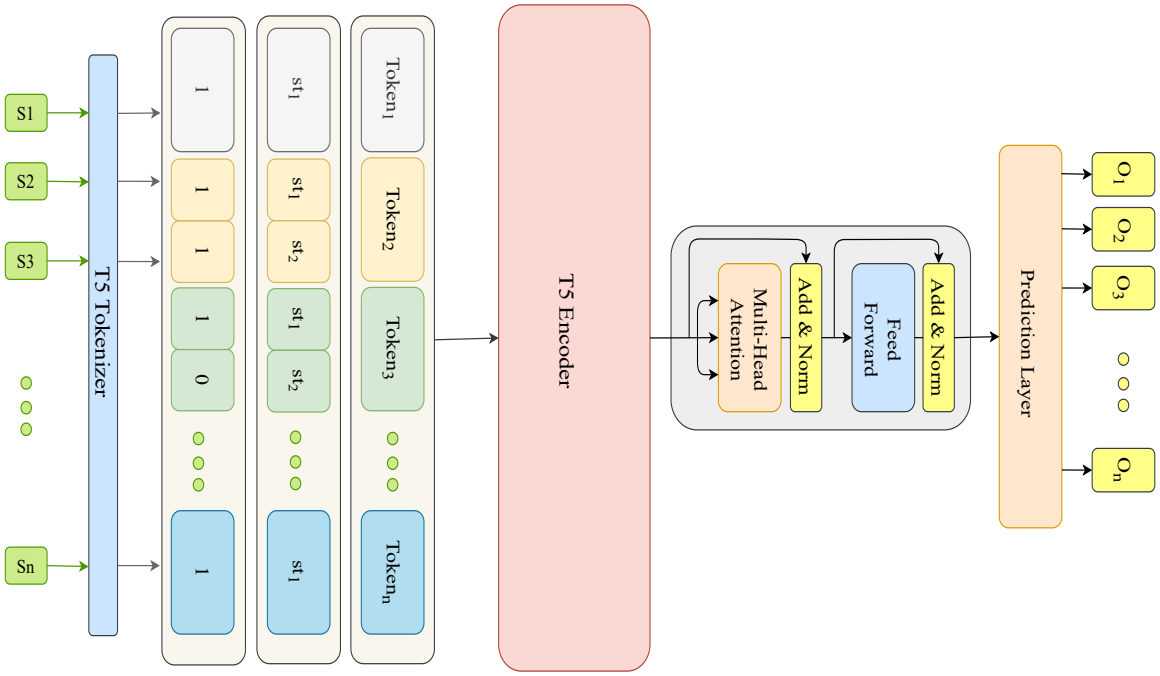

Figure 1: Proposed multilingual NER architecture

a position-wise fully connected feed-forward network. This architecture uses residual connections around each of the two sublayers followed by layer normalization. A multi-head attention module comprises several scaled dot-product attention used in parallel. In scaled dot-product attention, the input consists of three matrices Q, K, and V. The scaled dot-product attention is calculated using the following formula.

$$W_i^Q, W_i^K, W_i^V \in R^{d_{model} \times d_k}$$

$$Q = XW_Q, K = XW_K, V = XW_V \quad (1)$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The attention module has three trainable parameters, W_Q, W_K, and W_V. The three matrices Q, K, V are constructed by multiplying the input vector X by the corresponding matrices W_Q, W_K, W_V. Consequently, the dot product between Q and K is divided by $\sqrt{d_k}$ in order to prevent the dot product from becoming too large.

4.3 Prediction Layer

The vector obtained in the feature extraction Layer is given to a fully connected layer to predict the named entity label of the input sequence. For the input sequence $S = s_1, s_2, ..., s_n$ the output sequence $O = o_1, o_2 \ldots, o_n$ is predicted.

5 Experimental setup

We implemented the model in PyTorch and trained it on Nvidia V100 GPUs. The AdamW optimizer with a learning rate of 2e-5 is used to train the network. Our training method includes early stopping, which ensures the validation loss reduction with patience of 10 epochs. The training batch size is set to 32, and the dropout rate is 0.2. Transformer encoders have eight attention heads, and position-wise feed-forward layers have 2048 hidden sizes. In both T5 and MT5 tokenizers, the max length varies between 100 and 250 characters according to the evaluated language. The hyper-parameters of each subtask were tuned with the dev set. All other parameters are initialized randomly.

6 Results

Several experiments have been conducted to develop the most appropriate model for NER. Experiments with the English dataset can be found in Table 2. Due to the success of T5 in this study, this language model has been used to compute the word representation vectors. According to Table 2, using the T5 can improve F1-score by 4% compared to MultiCoNER Baseline that uses XLM-RoBERTa.

We have attempted to improve the language model results by adding deep learning architectures and textual features Tavan et al. (2021). We evaluated LSTM and Transformer architecture on

top of the T5 and found that using the transformer improved the F1-score further than LSTM. After various experiments, it was found that the use of BPEmb could not improve the F1-score of the model.

Models	Train	Dev	Test
MultiCoNER Baseline	-	77.60	-
T5	94.40	81.92	65.99
T5 + LSTM	96.40	82.20	67.54
T5 + Transformer	89.09	82.91	67.63
T5 + Transformer + subtoken + bpemb	90.12	82.36	67.22
T5 + subtoken + Transformer (ours)	**97.32**	**86.73**	**71.45**

Table 2: Experiments with English dataset

According to Table 2, the proposed model has achieved the F1-score of 86.73% and 71.45%, on the dev and test dataset for English, respectively, which is the highest F1-score among the other experiments. Figure 2 Compares the F1-score of different deep learning architecture.

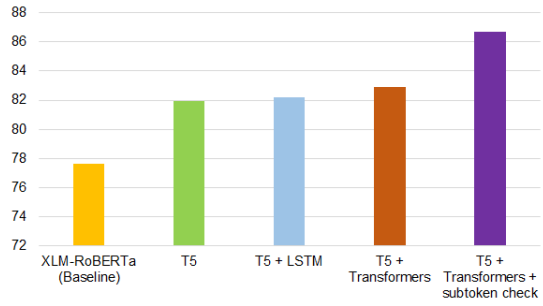

Figure 2: Compares the F1-score of deep learning architectures.

The results of the proposed model on the test dataset, as well as its ranking in the competition, are shown in Table 3. According to these results, Chinese, Hindi, and Bangla have lower F1-scores than other languages like German, Dutch, and English due to their language complexity.

The results of proposed model for the different entities are shown in Table 4. According to Table 4, the "PER" entity has reached the highest F1-score among other entities in all languages. Except for Russian, Turkish and Chinese, "CW" has the lowest F1-score in all other languages. As a result, the proposed model is weaker in identifying the "CW" entity than other entities.

7 Error Analysis

Several scenarios could occur when comparing the golden standard annotation with the output of a NER system Nejadgholi et al. (2020):

Language	Precision	Recall	F1-score
English (17)	71.11	71.91	71.45
Spanish (11)	68.65	68.71	68.30
Dutch (10)	71.18	71.98	71.13
Russian (10)	66.83	68.44	67.49
Turkish (10)	60.22	62.70	61.09
Korean (14)	61.13	63.92	62.26
Farsi (9)	61.80	63.06	62.14
German (12)	73.10	73.60	72.12
Chinese (19)	60.15	57.10	56.64
Hindi (12)	56.39	57.01	56.31
Bangla (11)	56.48	53.77	54.22
Multilingual (14)	69.38	70.47	69.28
Code-Mixed (21)	67.36	67.41	67.03

Table 3: Precision, Recall and F1-score on test dataset in all languages. The rank of the proposed model in each language is shown in parentheses.

Scenario 1, Complete True Positive:
An entity is predicted by the NER model correctly.

Scenario 2, Complete False Positive:
An entity is predicted by the NER model but is not annotated in the hand-labeled text.

Scenario 3, Complete False Negative:
The model does not predict a hand-labeled entity.

Scenario 4, Wrong label, Right Span:
A hand-labeled entity and a predicted one have the same span but different tags.

Scenario 5, Right label, overlapping spans:
A hand-labeled entity and a predicted one have overlapping spans and the same tags.

Scenario 6, Wrong label, overlapping spans:
A hand-labeled entity and a predicted one have overlapping spans but different tags.

The output of the proposed model on the English dev dataset has been evaluated on six scenarios in Table 5. From Table 5, 96.20% of "PER" entities are in Scenario 1, and the most significant proportion of Complete True Positives are related to this entity. Approximately 17% of "PROD" entities are in Scenario 2, higher than other entities. "PROD" actually has the highest Complete False Positive value among all entities. The "CW" has the greatest number of entities in Scenario 3 and the highest proportion of Complete False Negatives among other entities.

Scenario 4 includes 6.21% of "CORP" entities. Scenario 5 has the most significant number of entities compared to other scenarios, having 6.12 percent of the "PROD" entities. The "CORP" also has the highest number of entities in Scenario 6. Table 6 shows examples of the English test samples that are categorized in different scenarios.

Language	LOC	PER	PROD	GRP	CW	CORP
English	72.23	86.71	70.09	67.62	62.32	69.70
Spanish	66.95	84.77	63.50	63.89	61.31	69.40
Dutch	68.64	86.77	69.32	67.44	65.18	69.44
Russian	69.67	74.88	66.50	61.11	62.56	70.24
Turkish	64.29	70.76	63.83	52.31	54.93	60.44
Korean	71.48	68.32	59.81	60.13	50.12	63.72
Farsi	68.46	71.29	62.86	64.75	46.13	59.33
German	74.61	87.13	71.97	69.70	63.82	71.51
Chinese	67.93	57.59	64.38	34.08	51.95	63.93
Hindi	60.81	62.85	54.19	57.72	41.74	60.54
Bangla	57.04	67.26	51.12	64.63	33.03	52.22
Multilingual	74.19	81.11	63.96	63.61	63.33	69.49
Code-Mixed	72.98	78.99	68.81	59.50	58.52	63.39

Table 4: F1-score in English test dataset for each entity

Entity	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
LOC	215 (91.88%)	17 (7.26%)	8 (3.41%)	2 (0.85%)	5 (2.13%)	4 (1.7%)
PER	**279 (96.20%)**	9 (3.1%)	1 (0.34%)	4 (1.37%)	1 (0.34%)	5 (1.72%)
PROD	117 (79.59%)	**25 (17.0%)**	18 (12.24%)	1 (0.68%)	**9 (6.12%)**	2 (1.36%)
GRP	168 (88.42%)	5 (2.63%)	3 (1.57%)	9 (4.73%)	2 (1.05%)	9 (4.73%)
CW	134 (76.13%)	22 (12.5%)	**22 (12.5%)**	7 (3.97%)	5 (2.84%)	8 (4.54%)
CORP	155 (80.31%)	7 (3.62%)	3 (1.55%)	**12 (6.21%)**	8 (4.14%)	**15 (7.77%)**

Table 5: Results of different scenario in English dev dataset.

Scenario 1	========== HUMMAN ANOTATION ========== in 1841, he established a production of whale oil . ========== MODEL PREDICTION ========== in 1841, he established a production of whale oil .
Scenario 2	========== HUMMAN ANOTATION ========== these desktop application launchers work with microsoft windows operating systems only. ========== MODEL PREDICTION ========== these desktop application launchers work with microsoft windows operating systems only.
Scenario 3	========== HUMMAN ANOTATION ========== upper head lug joins the head tube and top tube ========== MODEL PREDICTION ========== upper head lug joins the head tube and top tube
Scenario 4	========== HUMMAN ANOTATION ========== the caps were jointly designed by major league baseball and the new era cap company . ========== MODEL PREDICTION ========== the caps were jointly designed by major league baseball and the new era cap company .
Scenario 5	========== HUMMAN ANOTATION ========== molten chocolate and a piece of a chocolate bar ========== MODEL PREDICTION ========== molten chocolate and a piece of a chocolate bar
Scenario 6	========== HUMMAN ANOTATION ========== he was also the inventor of the nerf football. ========== MODEL PREDICTION ========== he was also the inventor of the nerf football .

Table 6: Example of different scenario in English dev dataset.
PROD CW CORP GRP

8 Conclusion

This paper proposes a model that uses an encoder module of transformers in the top of the hidden state of the T5 to process the extracted features to obtain the most important feature representations. Many experiments were conducted to evaluate the model's performance and find the best language model. The experiments prove that our architecture is most compatible with the T5 language model and does cover a reasonable range of results. Since this dataset is multilingual, the MT5 embedding

module was selected.

The model's accuracy is confirmed by an in-depth analysis of the provided datasets. Accordingly, the same architecture was used in all other sub-tasks. Error analysis enabled us to identify specific NER challenges, creating immediate future tasks. We plan to apply more robust deep architectures to multilingual datasets as part of our future work.

References

Rodrigo Agerri and German Rigau. 2016. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, 238:63–82.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Ana Alves-Pinto, Christoph Demus, Michael Spranger, Dirk Labudde, and Eleanor Hobley. 2022. Iterative named entity recognition with conditional random fields. *Applied Sciences*, 12(1):330.

Behrouz Bokharaeian, Alberto Diaz, Nasrin Taghizadeh, Hamidreza Chitsaz, and Ramyar Chavoshinejad. 2017. Snpphena: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *Journal of biomedical semantics*, 8(1):1–13.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

Benjamin Heinzerling and Michael Strube. 2017. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. *arXiv preprint arXiv:1710.02187*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Tanvir Islam, Sakila Mahbin Zinat, Shamima Sukhi, and MF Mridha. 2022. A comprehensive study on attention-based ner. In *International Conference on Innovative Computing and Communications*, pages 665–681. Springer.

Shengbin Jia, Ling Ding, Xiaojun Chen, Yang Xiang, et al. 2020. Incorporating uncertain segmentation information into chinese ner for social media text. *arXiv preprint arXiv:2004.06384*.

Muhammad Raza Khan, Morteza Ziyadi, and Mohamed AbdelHady. 2020. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *arXiv preprint arXiv:2001.08904*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Jiatong Li and Kui Meng. 2021. Mfe-ner: Multi-feature fusion embedding for chinese named entity recognition. *arXiv preprint arXiv:2109.07877*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Amit Majumder, Apurba Paul, and Abhishek Banerjee. 2022. Deep learning-based approach using word and character embedding for named entity recognition from hindi-english tweets. In *Applications of Networks, Sensors and Autonomous Systems Analytics*, pages 237–243. Springer.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: a Large-scale Multilingual dataset for Complex Named Entity Recognition.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Mitchell P Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3. *Linguistic Data Consortium, Philadelphia*, 14.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512.

Isar Nejadgholi, Kathleen C Fraser, and Berry De Bruijn. 2020. Extensive error analysis and a learning-based evaluation of medical entity recognition systems to approximate user experience. *arXiv preprint arXiv:2006.05281*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Annie Rajan and Ambuja Salgaonkar. 2022. Named entity recognizer for konkani text. In *ICT with Intelligent Applications*, pages 687–702. Springer.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*.

Ehsan Tavan, Ali Rahmati, Maryam Najafi, and Saeed Bibak. 2021. Bert-dre: Bert with deep recursive encoder for natural language sentence matching. *arXiv preprint arXiv:2111.02188*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. *arXiv preprint arXiv:2105.03654*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv preprint arXiv:1909.01441*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. *arXiv preprint arXiv:2106.01223*.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. *arXiv preprint arXiv:1806.05626*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-markov crf for neural sequence labeling. *arXiv preprint arXiv:1805.03838*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. *arXiv preprint arXiv:2005.07150*.