Named Entity Recognition in Noisy Social Media Domain

Jishnu Ray Chowdhury
jraych2@uic.edu

Usman Shahid
hshahi6@uic.edu

Tuhin Kundu
tkundu2@uic.edu

Zhiming Zou
zzou6@uic.edu

Abstract

We address the challenges posed by noise and emerging/rare entities in Named Entity Recognition task for social media domain. Following the recent advances [4], we employ Contextualized Word Embeddings from Language Models pre-trained on large corpora [10] along with some normalization techniques to reduce noise. Our best model achieves state-of-the-art results (F1 52.47%) on WNUT 2017 dataset. Additionally, we adapt a modular approach to systematically evaluate different contextual embeddings and downstream labeling mechanism using Sequence Labeling and a Question Answering framework.

1 Introduction

A Named Entity is often described as a phrase that clearly identifies one item from a set of other items that have similar attributes e.g. names of people, locations, specific dates etc. Named Entity Recognition (NER) is the task of identifying named entities in a given text [38]. Named entities can span beyond a single word or token which makes NER a structured prediction task requiring the prediction of both the span and type of named entities present in the text [38]. For example in the sentence “Michael Jordan lives in Chicago”, we need to identify that there is a Named Entity “Michael Jordan” of type *Person* present in the text spanning over the first two words of the sentence; additionally, we need to identify the other Named Entity “Chicago” of type *Location* as the last word of the sentence. Factual information (who, where, what, when) is often captured in the form of named entities, hence, making NER an important step for downstream applications; specifically, NER can be used for Information Retrieval [39], Question Answering [28], Content Recommendation [14], Knowledge-Base Construction [39] etc.

While Named Entity Recognition (NER) has been quite successful in the formal domains (e.g. News and Publications) [21] and most state-of-the-art models [6] achieve near perfect results (over 90% F1 Score), it is particularly challenging in informal domains (e.g. Social Media) (49.59% F1 [6]) The challenges in informal domains arise from the unusual and noisy surface forms [37, 5, 12, 2], additionally, most entities are quite rare (following a long-tail distribution) and new entities keep emerging all the time [5, 12] which makes it harder to design generalizable methods for NER.

For example, Figure 1 shows texts from CoNLL 2003 [38] (News) and WNUT 2017 [12] (Social Media) datasets, representing formal and informal domains respectively. You can see regularities in the formal domain, such as capitalization of the Named Entities which are otherwise absent in the informal domain. The misspelling of the words e.g. ”luv“ and ”yeeuuuuppp“ exhibit the noisy surface forms and “trey” is potentially a rare entity.

Thus, in this work we focus particularly on this more challenging task of extracting Named Entities from a noisy domain which has much more room for improvement.

CoNLL 2003
[***Spanish***]$_{\text{MISC}}$ Farm Minister [***Loyola de Palacio***]$_{\text{PER}}$ had earlier accused [***Fischler***]$_{\text{PER}}$ at an [***EU***]$_{\text{ORG}}$ farm ministers ' meeting of causing unjustified alarm through " dangerous generalisation . "

(a) Formal domain (CoNLL, News).

WNUT 2017, Twitter domain
been listenin to [***trey***]$_{\text{PER}}$ alllll week ... can u luv someone u never met ?? bcuz i think im in luv yeeuuuuppp !!!

(b) Informal domain (WNUT 2017, Twitter)

Figure 1: Examples with NER tags fron CoNLL 2003 [38] and WNUT 2017 [12] tasks

Problem Formulation: Named Entity Recognition can be formulated as a **sequence labeling** task. Given a sequence $S = [w_1, w_2, w_3, ..., w_n]$ of size n, a set of words (or vocabulary) V such that $\forall_i, w_i \in V$ and a set of labels L, the objective is to learn a function $f_{SL} : (S, i) \rightarrow o_i$, $o_i \in L$ which maps each individual word in the sequence to a corresponding label. This is typically a structured prediction task with some syntactical and structural constraints existing on the sequence of labels. Practically, a standard implementation of a sequence-labeling function f_{SL} will include an embedding layer to map the words into dense vectors followed by a Bi-directional Long Short Term Memory (Bi-LSTM) model [16] to contextualize the word representaions. In the last layer a Conditional Random Field [19] is sometimes used [17] to more explicitly model the structural elements of the sequence labels.

We also explored the potential of **Question Answering (QA)** framework which was previously proposed by Li et al. [22]. But the main idea is older. As argued by Kumar et al. [18], almost any task in NLP, can be framed as a QA problem. NER is no exception. More precisely, here, we consider an extractive QA framework where the task is to usually predict the starting and ending positions of the answer spans in a given passage. In NER the query can be about a specific type of named entity. For example, “find me the names of people” can be a query about person type named entities. The passage would be the input text and the named entity spans of queried type can be the answers. Formally, we learn two functions $f_{QAstart} : (S, i, q) \rightarrow \{0, 1\}$ and $f_{QAend} : (S, i, q) \rightarrow \{0, 1\}$ where q is the query, S is the passage (the input text) where the answer(s) (the named entities) exists. For every word w_i in S, $f_{QAstart}$ maps it to (1) if i is the starting index of a named entity whose type is queried else it is mapped to (0). Similarly f_{QAend} maps every word w_i to (1) if i is the ending index of a named entity span else to (0).

In this work, we achieved state-of-the-art results on the WNUT-17 [12] task with the recently proposed ELECTRA [10] embeddings and our major contributions are summarized as follows:

1. We evaluated and compared three different Contextualized Word Embeddings (BERT [13], ELECTRA [10], Contextualized String Embeddings [3]) to represent context for downstream NER task.
2. We verified that normalization of the unusual surface forms (using character-level speech features and syntactic information) is helpful for NER regardless of other methods.
3. We tested the QA framework with different configuration for NER in the noisy domain and compared it with the sequence-labeling approach under similar architectural settings.

The following section (Section 2) provides a summary of existing work. In Section 3, we provide further details of the WNUT-17 [12] dataset. In Section 4 we explain different configurations of our methods followed by Section 5 containing experimental details. In Section 6 we provide our results along with the insights gained from this work. Finally, we discuss future directions for research in Section 7 and conclude in Section 8 respectively.

2 Background

Traditionally Named Entity Recognition (NER) have been looked upon as a sequence labeling task. Hidden Markov Models (HMM), and Maximum Entropy classifiers were some of the initial statistical approaches to the task [8, 9]. Collobert et al. [11] showed the possibility of using word-embedding based approaches for NER and other sequence labeling tasks. Huang et al. [17] introduced the Bi-LSTM-CRF framework for NER. Lample et al. [20] used character-level embeddings along with word embeddings for further improvement. Ma and Hovy [25] introduced the

Train Data		Dev Data		Test Data	
Posts	Tokens	Posts	Tokens	Posts	Tokens
3,395	62,729	1,009	15,733	1,287	23,394

Table 1: WNUT-2017 Statistics

CNN-BiLSTM-CRF stack where a Convolutional Neural Network (CNN) is used to produce word representations from character-level embeddings. The word-level character-feature representation is used alongside standard word-embeddings in a Bi-LSTM-CRF stack. Additionally, Ma and Hovy [25] demonstrated that it is possible to do very well in NER without gazetteers or other handcrafted features which many of the previous approaches [33] used. Bharadwaj et al. [7] used phonologically-aware character level features for NER in low-resource and cross-lingual settings.

Recently, following Peters et al. [35] most State-of-the-Art (SOTA) approaches to NER use some form of contextualized embeddings. Peters et al. [35] motivated the use of contextualized word embeddings (ELMo) generated from pre-trained Bi-LSTM language models to model word polysemy. Devlin et al. [13] introduced a Transformer Masked-Language-model based contextualized embeddings called BERT which was shown to achieve near SOTA performance on NER without even using a BiLSTM-CRF. Li et al. [22] introduced a new framework for NER. Instead, of taking the conventional sequence-labeling approach, they formulated NER as a machine comprehension or question answering task where the spans of the answers (named entities) are to be predicted. Li et al. [23] motivates the use of a variant of DICE loss instead of cross entropy to better handle the data imbalance that is inherently present in NER.

In the social media domain, Ritter et al. [37] conducted some initial experimental studies on NER in Tweets. They showed that off-the-shelf NER models trained on news domains are weak in the less formal domain of Twitter. Aguilar et al. [1] proposed a multi-tasking approach to NER in social media based on BiLSTM-CRF. Aguilar et al. [2] improved the above-mentioned approach by making their model phonologically aware to better handle the noisiness inherent in Tweets. Similar to the formal domain, the current SOTA in social media domain, is held by approaches that use contextualized embeddings. Akbik et al. [3] presented Contextualized String Embeddings to use on NER. Akbik et al. [4] motivated the use of a pooling operation to aggregate multiple instances of contextualized word embeddings occurring in different contexts in prior examples to further improve NER and handle new emerging named entities. Mayhew et al. [27] explored data augmentation strategy combining both true-cased and uncased versions of the samples in different manners to enhance NER in different domains including on a Twitter dataset. Another interesting approach used a true-case prediction model to enhance NER in social media dataset where capitalizations are often noisy and uninformative for NER [31, 26].

3 Dataset

In WNUT-2017 workshop [1], the Emerging and rare entity recognition [2] task was proposed and a dataset, WNUT 2017 [12] was created to serve as a benchmark for the task. This dataset is a compilation of texts from different informal domains such as Twitter[3], Youtube[4], StackExchange[5] etc. In addition to the noise, what makes this task challenging is that the training, test and dev sets [12] are extracted from different platforms: the training data is from Twitter, the validation data is from YouTube comments, and the test data is from Reddit and StackExchange. The dataset statistics are shown in Table 1. There are six classes of primary entities in the dataset: Person, Location, Corporation, Product, Creative Work and Group.

[1] https://noisy-text.github.io/2017/index.html

[2] https://noisy-text.github.io/2017/emerging-rare-entities.html

[3] https://twitter.com/

[4] https://www.youtube.com/

[5] https://stackexchange.com/

4 Methodology

We investigate two broad labeling mechanisms for NER: Sequence-Labeling and Question-Answering. In this Section, we describe the two mechanisms along with the specific models used.

4.1 Sequence-Labeling

The general sequence-labeling approach to NER is described in Section 1. We use BIO tagging scheme for the labels where 'B' indicates the beginning of a named entity, 'I' indicates inside of a named entity, and 'O' indicates outside of a named entity. We consider the following contextualized word embeddings:

1. **Contextualized String Embeddings (CSE)** [3], based on the hidden state representations of pre-trained character-level LSTM language models.
2. **BERT** [13], a Transformer-based bi-directional language model which is pre-trained using masked language modelling objective and next sentence prediction task. We used cased BERT pre-trained with whole-word masking. We used scalar-weighted aggregation of each layers of BERT to retrieve to embeddings. The scalar-weights corresponding to each layers are normalized using softmax and the weights are treated as parameters which are learned during training. Unlike CSE, BERT produces subword level embeddings which do not align with the word level sequence labels. To get word level embeddings we use mean pooling over the subwords associated to a particular word.
3. **ELECTRA** [10], another Transformer-based model. Unlike BERT it was pre-trained as a discriminator to detect words that were replaced by a generator. We used the same layer aggregation and pooling strategy as in BERT to extract its embeddings.

We concatenated contextualized embeddings with extra features like POS tags, static embeddings, or character-level speech features. The concatenated feature representations are then fed to a BiLSTM-CRF stack [17] which is often used as a standard for sequence-tagging tasks. We also ran various ablation tests where we ablate the BiLSTM-CRF component or some of the extra features.

4.1.1 Extra Features

During sequence-labeling we consider three types of features:

1. **Static Embeddings:** Prior research [4] shows a substantial boost in performance when using a static (non-contextualized) embedding along with contextualized embeddings in noisy settings. Thus, we also used static embeddings as extra features. We used recently released word2vec embeddings by Godin [15] which were trained on a large Twitter corpus. We expected it to be particularly appropriate for the noisy social media domain given its training domain.
2. **Syntactic Features:** We used POS-tags as syntactic features. We trained the POS embeddings from scratch during NER training and used a Twitter-optimized POS-tagger [32].
3. **Speech features:** Instead of using plain character level features as specified in prior research, we considered using character-level *speech* features similar to [2]. As Aguilar et al. [2] argued, in social media domains people often use informal word forms (which do not usually exist in formal domains: for example, writing "u" instead of "you"), however the informal word forms still tend to share similar articulatory features ("u" and "you" have similar pronunciation). Thus, the speech-features can be potentially used to *normalize* the variations in different wordforms that tend to occur in noisy domains. Motivated by this reason, we used *epitran* [30] to generate character-level phonetic (IPA) representations for each word, and we used *panphon* [29] to generate character-level phonological feature representations (represented as vectors). We trained the IPA-embeddings from scratch during NER training, and we directly use the phonological feature vector representation as embeddings.

While the static embeddings and the syntactic features are at word-level, the speech feature vectors are at the character level. To integrate them, we first had to create word-level representations of the character level features. To that end, first, we concatenated the character level embeddings and then used three parallel 1-dimensional character-level CNNs with different kernel sizes (to model different n-grams character features) followed by the Swish activation function [36]. For each CNN, we applied global max pooling over the word length to get a single vector representation for a word. We then concatenated the output of the three CNNs to create the final world-level representation of the character level features. After that, they were concatenated with the rest of the word level features (contextualized embeddings, POS-tags) before being fed to the Bi-LSTM-CRF.

4.2 Question Answering

The general QA approach to NER is described in Section 1. There are several motivations for taking the QA approach: (1) A natural language query can be modeled using pre-trained embeddings which can incorporate prior knowledge about entity types. (2) The span index based prediction can be used to handle overlapping and hierarchical named entities which cannot be addressed using standard sequence labeling. (3) The framework can be used for zero-shot generalization to unseen entity types. Below, we discuss different specific settings for this approach that we investigated:

Prediction Type: Similar to Li et al. [22], one prediction type that we considered is the span-index based prediction style as described in Section 1. Precisely, we use two binary classifiers (a single non-linear layer is used for each) on the final hidden state representations of each token in the sequence to model $f_{QAstart}$ and f_{QAend}. Unlike Li et al. [22], we do not use the additional start-end index alignment task which is necessary to handle overlapping spans. Since our dataset does not have any overlapping entities to consider, we used a simple heuristic to align each start index to its nearest end index. Another alternate prediction style that we considered is to use BIO tagging scheme (similar to sequence labeling mechanism) which can encourage the model to explicitly model the beginning and the insides of multiple spans. There are two main differences in this from the pure sequence labeling approach: (1) we are not predicting the types here (the type is already part of the query). (2) In this case, the sequence-tagger is conditioned on the given query and thus can enjoy the prior-knowledge encoded in a query.

Query Type: While there are some motivations to use a **natural language query** for a named entity type (since it can bring some prior knowledge), it is not prima facie too obvious why that is necessary when there are only six types of named entities. It is not clear if the full-blown modeling of sentence level natural language queries is necessary given the limited variety of queries. Out of this concern, we explore another approach to representing queries using **id-based queries**. In this approach, we map each entity type to an id which is then mapped to a randomly initialized trainable "query-embedding" which we treat as the "encoded query". We compare both types of queries.

Architecture: We consider two types of architecture. One is the standard setup used for "BERT-like" models, where both the question and passage are joined together with a separator token in between and then fed to the "BERT-like" model which is fine-tuned on the task. In our case, we primarily used ELECTRA for the QA experiments due ot its higher performance. While using the BIO type prediction, we can also employ a CRF on top of ELECTRA to model structural constraints. We used the same layer aggregation strategy as before. This setup can allow deep interaction between the question and the passage. The other setup that we consider is intended to be as close to the standard sequence-labeling framework as possible for the sake of comparison. In this setup we encode the query and concatenate the encoded query feature with every tokens in the passage. The concatenated features are then fed to a BiLSTM-CRF. We take a sequence-labeling approach for this case. In the case of a natural language query, we used ELECTRA to encode it (we sum up the hidden state embeddings and divide by the square root of the sequence length. The result is linearly transformed to the query dimension). In case of a id-based query, we used the randomly initialized query embeddings as query-encodings (as discussed before).

5 Experimental Details

For all our experiments, we used the following hyperparameters whenever applicable: a learning rate of $1e{-}3$, a fine tuning rate of $2e{-}5$ (only used for contextualized word embedding model parameters when applicable), AdamW optimizer [24] with a weight decay of $1e{-}3$, a maximum gradient norm cutoff of 5, a batch size of 32, a word-level dropout of 0.05 (just before Bi-LSTM when using one), a

Bi-LSTM input dropout of 0.5, 256 hidden units for forward and backward LSTM, IPA-embeddings of 22 dimensions, POS-tag embeddings of 16 dimensions, character-CNN output channels of 32 (all three), character-CNN kernel sizes of 3, 5 and 7, and an early stopping patience of 3.

Embedding	Metric	Embedding	Embedding	Embedding	Embedding	Embedding	Embedding
		—	—	—	+word2vec	—	+GloVe
		—	—	—	—	+POS & speech	—
		—	—	—	—	—	+Pooling
		—	—	+BiLSTM	+BiLSTM	+BiLSTM	+BiLSTM
		—	+CRF	+CRF	+CRF	+CRF	+CRF
CSE* [4]	F1 score	—	—	—	—	—	49.59
CSE	Precision	59.18	**67.39**	60.40	54.66	57.62	—
CSE	Recall	20.02	23.17	27.71	15.76	30.49	—
CSE	F1 score	29.92	34.48	37.99	24.46	39.88	—
BERT	Precision	49.83	58.37	53.65	56.70	56.44	—
BERT	Recall	26.97	28.45	34.75	16.87	36.98	—
BERT	F1 score	35.00	38.26	42.18	26.00	44.68	—
ELECTRA	Precision	54.29	63.92	61.46	60.36	58.80	—
ELECTRA	Recall	35.77	37.44	43.00	21.87	**46.43**	—
ELECTRA	F1 score	43.13	47.22	50.60	32.11	**51.89**	—

Table 2: Results from different embeddings and settings using the sequence-labeling mechanism. The best of every metric is in bold. * denotes that the result is reported verbatim from prior work. CSE* indicates the previous SOTA.

6 Results & Discussion

Comparing Contextualized Embeddings: From the experiments we find that under comparable settings, ELECTRA consistently outperform BERT and CSE by a large margin. This is expected because ELECTRA was pre-trained on a larger dataset. This is also consistent with previous research [10] showing that the discriminative pre-training over all tokens is better than generative pre-training on masked token for the downstream natural language understanding tasks [10]. As we would expect, adding a BiLSTM-CRF stack on top of the contextualized embeddings enables better utilization of the embedding features and further improves the performance.

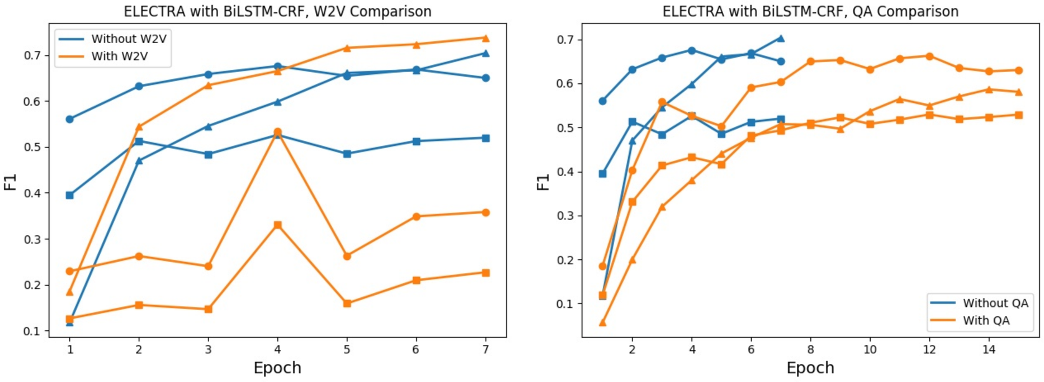
	ELECTRA	ELECTRA	ELECTRA	ELECTRA	ELECTRA	ELECTRA
	+Fine Tune	+Fine Tune	+Fine Tune	+Fine Tune	—	—
	—	—	+ CRF	+ CRF	+ CRF	—
QA framework	Yes	Yes	Yes	No	No	No
Query Type	Natural Lang.	Natural Lang.	Natural Lang.	—	—	—
Prediction Type	Span-index	BIO	BIO	BIO	BIO	BIO
Precision	46.27	43.59	49.31	**56.46**	63.92	54.29
Recall	43.74	55.14	**56.07**	47.36	37.44	35.77
F1 score	44.97	48.69	**52.47**	51.51	47.22	43.13

Table 3: Comparison of QA mechanism and Sequence-labeling mechanism using fine-tuned embeddings. We also compare different prediction types for QA framework.

Effect of extra features: There is a noticeable improvement in the performance when we use syntactic features (POS tags) and character-level speech features (phonetics and phonological features). We show that contextualized word embeddings do not compensate for these extra features and explicit normalization of the surface forms is necessary for better performance. Surprisingly, however, adding the word2vec embeddings substantially decreased the performance. This is quite unexpected

	ELECTRA +BiLSTM +CRF	ELECTRA +BiLSTM + CRF	ELECTRA +BiLSTM + CRF
QA framework	Yes	Yes	No
Query Type	Natural Lang.	Id-based	—
Prediction Type	BIO	BIO	BIO
Precision	56.21	51.10	**61.46**
Recall	**47.82**	45.32	43.00
F1 score	**51.68**	48.04	50.60

Table 4: Comparison of QA-mechanism and Sequence-labeling mechanism in the Bi-LSTM-CRF settings. We also compare different query types for QA framework.



(a) Word2Vec Comparison

(b) QA framework (MRC) Comparison

Figure 2: F1 scores for training (▲), validation (●) and test (■) data over multiple epochs.

because adding a non-contextualized embeddings had been previously shown to increase the performance of CSE [4] significantly. We suspect that our poor result is due to over-fitting in the training data (see Figure 2a). Both the word2vec embedding and the training data are from the same domain (Twitter) whereas the development and test set are from different domains. Possibly because of this, the model can learn to over-rely on word2vec during training which becomes counter-productive at test-time. Another reason can be Out-of-vocabulary (OOV) words. Word2vec requires a closed vocabulary unlike the contextualized embedding models. We built a closed vocabulary based on the training samples. Many of the significant words, especially, named entities in the development and test sets are potentially absent from the vocabulary. In such cases, word2vec embeddings would not be of much help. Again, in such cases, learned over-reliance on word2vec features can have highly detrimental effects during testing.

Effects of Prediction Types in QA: We investigate the effects of different prediction types in Table 3. We find that BIO-based tagging scheme works better than the span-index based predictions. Unlike the span-index prediction methodology, BIO-tagging scheme may encourage the model to align the start index and end index more naturally. The explicit prediction of tokens in-between (associated to the "I" tag) the start and end indices of a span can also potentially provide better loss signals. Moreover, using a structured prediction in the sequence-labeling style allows us to also exploit a CRF which further improves the performance.

Effects of Query Types in QA: In Table 4, we compare different types of query. We use the Bi-LSTM CRF setup for this set of experiments because we suspected that the id-based query is not suitable for the fine-tuning method since it would introduce a different unfamiliar kind of embedding (the embedding of the query id) into ELECTRA. To keep things comparable, we performed all the experiments in Table 4 on a similar Bi-LSTM-CRF setup. Here, we note that the natural language query works better on paper. However, on deeper analysis, it fails to fit very well in the training domain (see Figure 2b). We notice that the testing and validation domains might be closer to each other whereas the training domain is quite different. Training domain being from Tweets can generally be noisier than Reddit, stack-exchange or YouTube. This might result in lower performance on training set but relatively higher performance on the test and validation sets. We have observed this phenomenon in our experiments with Bi-LSTM-CRF with natural queries. We suspect that the under-performance of Bi-LSTM-CRF with natural language query in the training set is an artifact of using such an unorthodox QA architecture (there is no issue with natural language queries in Table 3) which we developed for better comparison with a pure sequence-labeling setting.

Effects of Framework: In both Table 3 and Table 4, we also compare the QA frameworks against the pure sequence labeling framework under comparable settings. In Table 3, the fine-tuned ELCTRA-CRF QA approach is slightly better than the same fine-tuned model in a sequence-labeling setting. Overall, ELECTRA-CRF QA acheives a new SOTA on WNUT-2017 followed closely by ELECTRA-Bi-LSTM-CRF with extra features in the sequence-labeling settings. Similarly, the best QA model in Table 4 seems to be slighly better than the similar sequence-labeling approach, however that model also have issues with fitting on the training data as discussed before. It may be possible to further improve the QA performance, if we do not omit the span start-end-index alignment objective. Although there is no apparent motivation for span-alignment objective when there are no span overlaps, it may be still possible that the span-alignment objective can provide a better (and more explicit) inductive bias to improve the span prediction task. Testing this hypothesis is the beyond the current scope of study.

Effects of DICE Loss: We also experimented with a variation of DICE loss as proposed by Li et al. [23] which is supposed to better handle the class-imbalance in NER (most tokens are not named entities), especially the dominance of negative examples. However, we were unable to get any better results from DICE loss. This may require further exploration in the future.

7 Future Work

One limitation of the current study is that the QA framework was mainly used for comparison. The current study does not fully explore the possible synergies between the best techniques used in the sequence-labeling experiments and in the QA experiments. For example, one possible combination can be ELECTRA-BiLSTM-CRF with fine tuning, natural language query, and extra features for normalization. Even though the Twitter trained word2vec performed poorly, there is still a good potential for getting a further improvement by using some form of non-contextualized embeddings like GloVe [34] (previous work does get a substantial boost [4]). Another direction of research can be to incorporate the case (capitalized or not) information in texts. It is an important feature for NER in formal domain but not particularly useful in noisy domains like Twitter [37]. However, instead of relying on the natural casing (which can be noisy and informal) in noisy domains, we can use the casings predicted by a statistical true-casing model [31, 26] trained on formal domains. The statistical model can annotate a given text with its cases in a more formal style which enables capitalization to better correlate with named entities. We may also enhance NER by means of data-augmentation. Data Augmentation can be used to make sure that our model does not rely heavily on the surface form of named entities and reliably encodes the context (which is crucial to deal with emerging Named Entities). Precisely, we can augment the data by replacing the original named entities with different entities of the same type in every epoch. This enables the model to derive information from the context instead of the surface form.

8 Conclusion

To conclude, we investigate different contextualized embedding and labeling mechanisms in NER for noisy social media domain. We found contextualized fine-tuned ELECTRA [10] embeddings and surface form normalization (using speech and syntacitc features) to be most effective. Moreover, the QA framework also provides a slight benefit in general. Using the QA approach, we achieved state-of-the-art F1 score of 52.47% on WNUT 2017 dataset. This is closely followed by the best F1 score of 51.89% which is achieved by a sequence-labeling framework. Both results outperform the previous SOTA (49.59%) by a considerable margin.

References

[1] Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. A multitask approach for named entity recognition in social media data. In *In 3rd Workshop on Noisy User-generated Text*. ACL, 2017.

[2] Gustavo Aguilar, Adrian Pastor López-Monroy, Fabio González, and Thamar Solorio. Modeling noisiness to recognize named entities using multitask neural networks on social media. In *In NAACL HLT, Volume 1 (Long Papers)*. ACL, 2018.

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *In ICLR*. ACL, 2018.

[4] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *In NAACL HLT, Volume 1 (Long and Short Papers)*, 2019.

[5] Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. Generalisation in named entity recognition. *Comput. Speech Lang.*, 44:61–83, 2017.

[6] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*, 2019.

[7] Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime Carbonell. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *In EMNLP*, 2016.

[8] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Applied Natural Language Processing*, pages 194–201. ACL, 1997.

[9] Andrew Eliot Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University, USA, 1999. AAI9945252.

[10] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[11] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2010.

[12] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proc. of 3rd Workshop on Noisy User-generated Text*, pages 140–147, 2017.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *In NAACL HLT, Volume 1 (Long and Short Papers)*. ACL, 2019.

[14] Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6), 2017.

[15] Fréderic Godin. *Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing*. PhD thesis, Ghent University, Belgium, 2019.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.

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

[18] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, pages 1378–1387, 2016.

[19] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In ICML*, 2001.

[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *In NAACL: HLT*, pages 260–270. ACL, 2016.

[21] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[22] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.

[23] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*, 2019.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *In ICLR*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[25] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *In ACL (Volume 1: Long Papers)*, pages 1064–1074. ACL, 2016.

[26] Stephen Mayhew, Nitish Gupta, and Dan Roth. Robust named entity recognition with truecasing pretraining. *arXiv preprint arXiv:1912.07095*, 2019.

[27] Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. ner and pos when nothing is capitalized. In *In EMNLP-IJCNLP*, 2019.

[28] Diego Mollá, Menno van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, November 2006.

[29] David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *COLING*, pages 3475–3484. ACL, 2016.

[30] David R. Mortensen, Siddharth Dalmia, and Patrick Littell. Epitran: Precision G2P for many languages. In *In LREC*, 2018.

[31] Kamel Nebhi, Kalina Bontcheva, and Genevieve Gorrell. Restoring capitalization in #tweets. In *In WWW*, WWW '15 Companion, page 1111–1115. ACM, 2015. ISBN 9781450334730.

[32] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *In NAACL: HLT*, pages 380–390. ACL, 2013.

[33] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. In *In CoNLL*, pages 78–86. ACL, 2014.

[34] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *In EMNLP*, pages 1532–1543, 2014.

[35] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[36] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2018. URL `https://openreview.net/forum?id=SkBYYyZRZ`.

[37] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *In EMNLP*, pages 1524–1534. ACL, 2011.

[38] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *In NAACL*, 2003.

[39] Daisy Zhe Wang, Yang Chen, Sean Goldberg, Christan Grant, and Kun Li. Automatic knowledge base construction using probabilistic extraction, deductive reasoning, and human feedback. In *Proc. of Joint Workshop on AKBC-WEKEX, NAACL HLT*, pages 106–110. ACL, 2012.