NoNE Found: Explaining the Output of Sequence-to-Sequence Models when No Named Entity is Recognized

Anonymous ACL submission

Abstract

Sequence-to-sequence (seq2seq) models are known to be effective for named-entity recognition (NER). Here we focus on explainability of seq2seq NER models. Contrary to most efforts that focus on explaining why a certain named entity has been recognized, we concentrate on negative cases i.e., sequence-level true negative or false negative, in which no named entity (NoNE) is recognized. Detecting sequence-level false negatives is critical in certain use cases such as location extraction in social media texts for disaster events, to not miss any location mentions. We introduce an approach to feature-relevance explainability for seq2seq models that leverages, a special class-of-input (COIN) token to capture whether or not a named entity was present in the input sequence. We run experiments on a location extraction task using a modified translation model (TANL) and generate NoNE explanation for the sequence-level negatives. We carry out a systematic use case-based validation procedure for our NoNE explanation approach. The experiments demonstrate that our NoNE approach is able to deliver important information about shortcomings of the seq2seq model and to uncover gaps in the formulation and application of the protocol used to annotate the data.

1 Introduction

In this paper, we investigate the explainability of sequence-to-sequence (seq2seq) named entity recognition (NER) models. Most papers discussing explainable methods for NER focus on explaining the positives, presence of the named entities, but not much attention is given to the negatives (Lin et al., 2020; Güngör et al., 2020; Agarwal et al., 2021). In contrast, we introduce an approach that generates explanations for sequences in which no named entity (NoNE) is found. We call the sequences in which no named entity is found as "sequence-level negatives". We run experiments on NER for location entities in crisis-related tweets collected during various different disaster events such as floods, earthquakes, wildfires, hurricane and cyclone. This disaster risk management use case of location extraction in tweets is an example of an NER problem that needs explainability techniques for an area in which decision support is critical and in which explanations for false negatives are especially important. The crisis-related tweets come from Suwaileh et al. (2022)'s paper where they present a large-scale dataset for the NER task of identifying locations in disaster tweets Suwaileh et al. (2022)'s study refers to this NER task as location mention recognition.

With the emergence and rapid progress of various pre-trained language models, we have observed a trend of researchers solving one NLP task by reformulating it as another. This approach is successful for many NLP tasks and provides a promising way of improving NLP models' performance and also has great potential in unifying various NLP tasks, making it possible to use single model for diverse tasks (Sun et al., 2020). Named entity recognition (NER), traditionally a token classification or sequence labeling task, has been recently addressed by researchers as a seq2seq task leading to the current state-of-the-art-performance on various NER datasets (Athiwaratkun et al., 2020; Paolini et al., 2021; Yan et al., 2021; Sun et al., 2020). Paolini et al. (2021)'s Translation between Augmented Natural Languages (TANL) framework solves several structured prediction tasks, including NER, in a unified way with a common architecture and without the need for task-specific modules by framing the task as a translation task between augment natural languages.

Our main contributions are as follows: We propose an approach to feature-relevance explainability for seq2seq NER models by leveraging a class-of-input (COIN) token to capture whether or not there is a named entity present in the input sequence. The sequence-level classification on top of the NER task (using the COIN token label) allows us to categorize an entire sequence into two categories sequence-level positives (sequence contains named entities) and sequence-level negatives (sequence contains no name entities). Secondly, we carry out seq2Seq NER experiments on a location extraction task using the TANL framework and explore the explanation techniques on the model's predictions, particularly the NoNE cases. Thirdly, we carry out use case-based systematic validation procedure for our NoNE explanation approach. Lastly, we demonstrate that our NoNE approach delivers important information about the seq2seq model such as shortcomings and insights into edge cases that can be useful to further improve the model.

2 Related Work

In this section, we discuss how NER has been solved as a seq2seq task and its advantages. Then, we present the current work on explainable NER, where there is a research gap on explaining when no named entities are detected by models. Lastly, we discuss the current work on seq2seq explainability methods and how they present an opportunity to be applied for the NER task.

2.1 Seq2seq NER Approach

The seq2seq approach has been commonly used in machine translation, language modelling, summarization, and question-answering tasks but there is a ongoing trend towards seq2seq for NER especially with the emergence of large language models like T5 and BART (Raffel et al., 2020; Lewis et al., 2020). Wang et al. (2019) propose for the first time the seq2seq model to be used for NER, called SC-NER, that has a classifier (which can be trained jointly with the encoder and decoder), added to determine whether sentences have entities. Athiwaratkun et al. (2020) proposed a seq2seq framework that combines sequence labeling and sentence-level (sequence-level) classification in an augmented natural language format. For the sequence-level classification, Athiwaratkun et al. (2020) used the pattern `(( sentence-level label ))` in the beginning of the sentence. Our COIN token is closely related to this approach, however is clearly novel since Athiwaratkun et al. (2020) do not consider explainability.

Paolini et al. (2021)'s TANL framework solves several structured prediction tasks, including NER, in a unified way with a common architecture and without the need for task-specific modules. We adopt the TANL framework in our work. We chose the framework because it can handle multitask learning, achieves at performance comparable to the current state-of-the-art for NER and have showed potential in generalizability (Paolini et al., 2021). Table 1 shows an example of TANL in use for NER for location entities on a crisis-related tweet. Furthermore, with the improved transfer of knowledge about label semantics, TANL can significantly improve the performance in the few-shot regime (Paolini et al., 2021) or when there is limited training data, which is the common case for disaster events.

2.2 NER Explanations

To the best of our knowledge, there has been no explainable NER papers that produce explanations for when models predict no named entities. Güngör et al. (2020) proposed a method of explaining NER predictions by assigning importance values to the morphological features of the detected entities and were only interested in the explanation of the named entity regions. Zugarini and Rigutini (2023) investigated XAI techniques for NER by involving semantic knowledge in generating global explanations for each named entity. Lin et al. (2020)'s paper introduced 'entity triggers', proxy of human explanations, group of words in a sentence that helps to explain why humans would recognize an entity in a sentence. They argued that a combination of entity triggers (explanations) and standard entity annotations can enhance the generalization power of NER models (Lin et al., 2020).

2.3 Seq2seq Explanations

Feature attribution methods, also referred to as saliency methods, are widely used to interpret model decisions. These methods assign distributions of importance scores over input tokens to represent their impact on model predictions (Simonyan et al., 2014; Murdoch et al., 2018; Madsen et al., 2022). Work on feature attribution for seq2seq has been mainly focused on machine translation, highlighting word alignments, coreference resolutions capabilities and model training dynamics (Ding et al., 2019; He et al., 2019; Voita et al., 2021). As seq2seq models are not traditionally used for NER, they present an opportunity to research explainability methods. Framing the NER task as a seq2seq task lets us utilize sequential

Input: `# PakArmy rescuing the injured in # Earthquake without a rest , Whereas , Young volunteers from neighbouring cities like # SaraiAlamgir ,# Jhelum , # Kharian ,# Gujrat are helping people in hospitals . # Mirpur is awake , # Pakistan is awake !` **`#0`**

Output: `# PakArmy rescuing the injured in # Earthquake without a rest , Whereas , Young volunteers from neighbouring cities like # [SaraiAlamgir | location] , # [Jhelum | location] ,# [Kharian | location] , # [Gujrat | location] are helping people in hospitals . # [Mirpur | location] is awake , # [Pakistan | location] is awake ! [` **`#0`** `| wloc]`

Table 1: Example of TANL-based NER with COIN token prediction output. The **Input** sequence is concatenated with **`#0`**, our COIN token at the end. The **Output** contains location entities being enclosed in the **[]** tags and labeled as *"location"*. The COIN token makes it possible to generate a sequence-level classification label: either **`nloc`** for *(no location)* or **`wloc`** *(with location)*

.

attribution methods, which involve a multi-step iteration (Sarti et al., 2023).

In this work, we follow the method of Ferrando et al. (2022), which integrates the contributions of both the source and target to seq2seq predictions. Our work differs from Ferrando et al. (2022) with the introduction of the COIN token which we use for sequence classification, this generates the “with location” and “no location” label for the entire sequence. By adding our COIN token, we can look into the COIN token label, in the case of the no location, “nloc”, itself and probe which words contributed the most or least to the generation of “nloc”. Other forms of seq2seq explanations consider rationales, subsets of context, that can explain individual model predictions, where the best rationale is the smallest subset of input tokens that would predict the same output as the full sequence Vafa et al. (2021).

3 Methods

In this section, we introduce the overall framework of our proposed NoNE explanation approach. First, we briefly describe the problem of how there is not much attention given to explaining sequence-level negatives, sequences where no named entities are found, in NER models and how this is critical in areas in which explanations for false negatives are especially important. Second, we propose the use of seq2seq NER models, TANL, a modified translation model, to fill the gap by running experiments on NER for location entities in crisis-related tweets. Third, we present our explanation generation method. Last, we introduce our automatic and manual validation of our NoNE approach.

Problem Definition. The NER task is to identify and classify all entity occurrences in a sequence. In the sequence-level view of the NER task, there are two cases of sequences: (1) sequence contains named entities (sequence-level positive) (2) sequence does not contain named entities (sequence-level negative) There have been various studies on explaining when named-entities are detected in sequences, however, not much attention is given to explaining when no named entities are detected. To the best of our knowledge, there has been no explainable NER papers that produce explanations for when models predict no named entities. It is important to explain when models predict no named entities to allow users to distinguish the two cases of sequence-level negatives, (1) sequence-level true negatives: when there are no named entities detected and the model is correct and (2) sequence-level false negatives: when there are no named entities but the model is not correct.

Seq2Seq NER with COIN token. We utilized Paolini et al. (2021)’s TANL framework to formulate NER as a seq2seq task. Paolini et al. (2021)’s TANL frames structured prediction tasks such as NER into a text-to-text translation problem. The augmented languages are designed in a way that makes it easy to encode structured information (such as relevant entities) in the input and to decode the output text into structured information (Paolini et al., 2021). We modify the use of TANL by adding the COIN token, which serves as the prompt for a sequence classifier that classifies the sequence based on presence of a location entity: ”nloc” for no location and ”wloc” for with location. The presence of the COIN token label (”nloc” for no-location label) allows us extract attribution scores that we use for explanation word selection.

Explanation. To analyze the NoNE cases, i.e., the negative cases in which no named-entity is found, we look at the word probabilities that explain the label in the output corresponding to the COIN token in the input. The feature-attribution explanations we use in this paper are word-level explanations and have the following form: Given a decision by the model, i.e., a word in the output sequence, the explanation contains a probability for each word in the input sequence. This probability reflects the importance of that word's contribution to the decision.

In order to generate our probabilities, we use Inseq (Sarti et al., 2023), a toolkit for transformer sequence models (Wolf et al., 2020) that uses attribution methods mainly sourced from Captum (Kokhlikyan et al., 2020). Generated probabilities are in the token-level (subword). Inseq aggregates the subword output of the model to probabilities at the word level. Specifically, it uses the logits of the subword tokens, which are commonly used in feature attribution (Bastings et al., 2022).

With the generated classification label of our COIN token, we can infer the word-level generation probability score of the sequence-level negative label. This allows us to investigate explanation approach for sequence-level negatives of NER.

For the explanation generation, we distinguished our low confidence negative tweets from the high confidence negative heuristically. The criteria we set for a tweet to be considered low confidence is that the tweet must contain at least one word with individual word-level probabilities that is in the bottom 30th percentile of the entire distribution of word-level probabilities in the entire set.

We set feature attribution so that there would be no more than 2 highlighted words as explanations in any sentence, since this would be comfortable for the disaster managers who need to review the explanations. The highlighted word is the word that contributes the least to the generation of the 'nloc' label of our COIN token. We consider the least contributing words as our possible explanation words with the rationale that the generation of these words were the most likely considered possible location words or having location-like form by the model. Hence, we hypothesize that with showing these explanation words to our enduser, they can determine whether or not the NoNE explanation words are possible location words.

Validation. We carried out a systematic manual validation of our explanations to recognize whether or not our system is effective in helping an enduser, a disaster manager, in quickly reviewing the NER output. As the disaster manager wants to capture all the tweets that contain location entities and not miss any tweet with a location, we want to minimize the number of false negative tweets. We refer the entire tweet a false negative when the model predicts that the entire tweet doesn't have any location entity ('nloc' is generated) but there is/are actual location entity/entities in the tweet. We carried out two validation approaches on our set of low-confidence negative predictions: automatic validation and manual validation. The disaster managers have no time to check all the individual tweets. Hence, we want to generate explanations only to the sequences where our model has lower confidence on their prediction and the choice of threshold was mentioned above.

Our automatic validation is calculated by how many false negative tweets would be missed if the disaster manager only looked at our set low confidence cases. We consider our NoNE approach good if our enduser misses a relatively low number of false negative tweets when looking only at our low confidence tweets. Most of the false negative tweets should be in the low confidence tweets.

The manual validation is calculated as the success rate of the disaster manager on differentiating true negative tweets and false negative tweets given only the explanation words and their respective immediate neigborhood words (three-word phrase). The authors act as a disaster managers (enduser) to do the manual classification of sequence-level false or true negatives. Table 2 shows example three-word phrases for validation by our enduser, the word at the middle being the word that has the least contribution to the prediction of 'nloc' (no location label for the tweet). The enduser will see these three-word phrases and decide accordingly whether the tweet is a false negative or a true negative. In the next section, we present the experimental setup.

visited DHQ Mirpur
Mirpur Police
in Stuart AB

Table 2: Examples of NoNE explanation three-word phrases for validation by our enduser

4 Experimental Setup

In this section, we discuss the experimental setup of our study. We start with a description of the datasets we used. Then, we discuss the experiments we conducted to do the NER task.

4.1 Datasets

Our experiments use the crisis-related Twitter NER Datasets from English-language IDRISI-RE (Suwaileh et al., 2023), however, we chose our own training and test split and made sure that they are mutually exclusive. Each disaster event in the original IDRISI-RE dataset is split in train and dev. A detailed train-test split of the tweets we used in the experiments can be found on Table 3.

For each tweet data, we appended the COIN token #0 at the end and labeled them with non-word labels “ABX” and “ABY” which correspond to the classification of the entire tweet as without location and with location respectively.

Our training and test sets are mutually exclusive and the training set contains events that precede the event in the test data (which is necessary for the disaster management scenario). We hold-out 10% of our training set for dev, but this set is just used to determine the effectiveness of training on flood data and not for hyperparameter tuning which was not needed.

4.2 Experiments

BERT-based NER (Suwaileh et al., 2022) is tested as a baseline model, due to its high performance reported by Suwaileh et al. (2023) The model was fine-tuned on the training data. We do not generate explanations for the BERT model as our explanation approach is for seq2seq models only.

TANL-based NER (Paolini et al., 2021), the T5-base NER model, is our main model. We fine-tune the pre-trained model on our training data. We used the same hyperparameters of Paolini et al. (2021)’s TANL paper in training with the CoNLL03 NER dataset except for the max sequence length, where we set it to 512 and the number of beams for beam search to 4.

TANL-based NER with COIN token. We modified the original TANL-based NER framework with our addition of the class-of-input (COIN) token Our NoNE explainability approach works as follows: We append a COIN token `#0` to the input sequence, i.e., to each training tweet. Then we assign a label to each, reflecting whether or not the input sequence contains a named entity. The label appears at the end of the targeted output sequence: `wloc` for *with location entities* or an `nloc` for *no location entities*.

5 Results

In this section, we discuss the results of our experiments. First, we present the model results of our TANL-based NER models both with and without COIN token and we compare this with the results of a baseline BERT-NER model trained on crisis-related tweet data. Then, we demonstrate the results of the validation of our NoNE explanation approach both automatic and manual validation. Lastly, we discuss some insights gained from our NoNE explanations that showcase our model’s shortcomings and insights on edge cases.

5.1 Model Results

We ran a comparison between the TANL NER with no COIN, and with the COIN token to ensure it didn’t affect the NER performance. Among the TANL NER with COIN models, the best performing was the TANL NER with COIN model trained with FLD train set (TANL.FLD.COIN). In exploratory experiments, we determined that the model was relatively robust to the choice of training data so we chose to train on the FLD train set. We also observed that adding the COIN token does not disadvantage. For the models fine-tuned with the FLD train set and the MIX train set (both flood and earthquake), we actually observed an F1-score increase of up to 0.09. However, this is not consistent across train sets.

The performance of the models on the test set are shown in Table 4. This is promising as not only can we generate explanations with the tokens but we can also improve or maintain performance of the model.

5.2 Validation Results

As previously mentioned in our Methods section, we had two validation criteria for our NoNE approach. We carried out the validation on the low confidence negative tweets set that consists of 379 tweets in total. The breakdown of these tweets per disaster event can be found in Table 5.

First, we ran an automatic validation where we measure the percentage of false negatives in the low confidence tweet set. We found that the low confidence negative tweets contain 68% of the the total

Event	IDRISE Orig. Split	# of Tweets	Event	IDRISE Orig. Split	# of Tweets
Our training set			**Our test set**		
Sri Lanka Flood 2017	train, dev	457	Midwest US Floods 2019	train,dev	864
Kerala Flood 2018	train, dev	1,300	Pakistan Earthquake 2019	train, dev	616
Maryland Flood 2018	train, dev	422	California Wildfires 2018	train, dev	1,075
Ecuador Earthquake 2016	train, dev	1,153	Cyclone Idai 2019	train, dev	1,388
Italy Earthquake 2016	train, dev	590	Hurricane Dorian 2019	train, dev	938
Kaikoura Earthquake 2016	train, dev	1,231			
Mexico Earthquake 2017	train, dev	1,300			
Total		6,453	Total		4,881

Table 3: Training-test split of the data used in the experiments, created from the original IDRISI splits. The training and test set are mutually exclusive.

Model	Midwest US Floods	Pakistan Earthquake	California Wildfires	Cyclone Idai	Hurricane Dorian
BERT-NER	**0.95**	0.84	0.77	0.90	0.86
TANL.FLD	0.83	0.81	0.84	0.81	0.63
TANL.EQK	0.83	0.80	0.83	0.82	0.71
TANL.MIX	0.84	0.81	**0.85**	**0.84**	0.71
TANL.FLD.COIN	**0.88**	**0.79**	0.83	**0.84**	0.72
TANL.EQK.COIN	0.85	0.75	0.83	0.78	0.74
TANL.MIX.COIN	0.82	0.77	0.84	0.81	0.73

Table 4: Performance of Models using F1-score as metric. The presence of the COIN token does not impair the performance of the the TANL-based NER models.

number of sequence-level false negative, meaning the disaster manager misses a relatively low number of false negatives when looking only at the low confidence negative tweets. The breakdown of the % of sequence-level false negative found in low confidence negative tweets per disaster event can be found in Table 5. We observe that are approach perform quite well for California Wildfires and Pakistan Earthquake at 84% and 79%, respectively.

Lastly, we conduct our manual validation approach: given only the explanation words and their respective immediate adjacent words (neighborhood words), how often can the enduser differentiate between sequence-level true and false negative? The enduser was able to correctly differentiate sequence-level true and false negative 79% of the time. We show the breakdown of the success rate of identifying both sequence-level false negative and true negative by disaster event in Table 5. From the percentages, we observe that there is room for improvement in identifying the sequence-level false negatives and the optimal window size can be further investigated in this regard. On the other hand, as shown on Table 5, the enduser is able to identify sequence-level true negatives at a high rate.

5.3 Insights from NoNE explanations

In this subsection, we present examples of four categories of NoNE explanations in which no named entity was found and discuss the insights that our NoNE approach delivers. We present four cases of NoNE explanations: (1) False Negative sequence where the model incorrectly predicts with low confidence and enduser agrees with the ground truth (2) True Negative sequence where the model predicts with low confidence and the enduser agrees with the ground truth (3) True Negative sequence, enduser predicts false negative, disagrees with the ground truth and (4) False Negative sequence, enduser predicts true negative and disagrees with ground truth. Cases 1 and 2, found in Table 6, reveal how our model works and its shortcomings. For cases 3 and 4, we present how our NoNE approach unveils possible annotation errors in the ground truth of the dataset, which can be found in Table 7.

The first case (top of Table 6) are false negative

Event	# Low conf. neg seqs	% FN seqs in low conf. (Automatic)	Success rate %FN found (Manual)	Success rate %TN found (Manual)
Midwest US Floods 2019	24	56	60	100
Pakistan Earthquake 2019	40	79	59	100
California Wildfires 2018	67	84	50	83
Cyclone Idai 2019	103	59	31	94
Hurricane Dorian 2019	145	65	75	95

Table 5: Results of Validation of NoNE Approach. The automatic validation results show the percentage of the total sequence-level false negatives found in the low confidence negative tweet set. The manual validation results two values: (1) success rate of enduser to correctly identify sequence-level false negatives and (2) success rate of enduser to correctly identify sequence-level true negatives.

Case 1: False Negative. Model has *low* confidence. Enduser agrees with ground truth
`1(a) RT @USER : # PMAJK visited { DHQ } { Mirpur} visited injured # earthquake [ #0 | nloc ]`
`1(b) Prisoners at { Mutimurefu }, also victims of # CycloneIdai , as the winds blew off the roofs of four cells at the prison [ #0 | nloc ]`
Case 2: True Negative. Model has *low* confidence. Enduser agrees with ground truth
`2(a) The death toll just keeps risingD # { californiawildfire } [ #0 | nloc ]`
`2(b) In times of crisis like this we must all come together to provide relief , serve lives and give hope and encouragement . Do nt ask what others are doing but look at { yourself } and ask : What can l do to help my compatriots . # CycloneIdai # PamberiNeZimbabwe [ #0 | nloc ]`

Table 6: Examples illustrating the insight delivered by our NoNE explanation approach that reveal how our model works and its shortcomings. {} indicates words that are important in the explanation because they are assigned low probabiliy of contributing to the prediction of nloc (the no-location label). Underlining indicates ground truth location entities.

predictions for which the model has low confidence based on our threshold and where our enduser agrees with the ground truth upon manual validation. These were missed by our TANL-based NER model, leading to a false negative tweets. These examples show our model's shortcomings. The locations that were annotated in the ground truth are underlined. The explanation words selected by our NoNE approach are indicated by ({ }), these are words that are assigned a low probability of contribution to the prediction of 'nloc' (no location label for entire tweet). For case 1(a), the words "`DHQ`" and "`Mirpur`" contributed the least to the prediction of 'nloc'. A possible reason for "`DHQ Mirpur`" to be missed by the model is how the tweet is grammatically incorrect. For case 1(b), the word "`Multimurefu`" contributed the least to the prediction of 'nloc'.

The second case (second row of Table 6) are true negative predictions for which the model has low confidence based on our threshold where our enduser agrees with the ground truth upon manual validation. For case 2(a), although "`californiawildfire`" contains a location word, according to Suwaileh et al. (2023)'s annotation protocol if the location name is a hashtag it is not considered as a location entity. For case 2(b), the explanation word chosen is "`yourself`" that may have been because of the preceding preposition "`at`".

We present how the NoNE approach can deliver important information about the ground truth data and helps to uncover errors in the ground truth of the data set as seen in cases 3 and 4 found in Table 7. The third case (third row of Table 7) are the false negative predictions for which the model has *low* confidence and the enduser disagrees with the ground truth. The locations that were annotated in the ground truth are underlined. In the explanation, some words (indicated by { }) have low attribution scores reflecting a weak contribution to the prediction of the 'nloc' tag. For case

Case 3: False Negative. Model has *low* confidence. Enduser *disagrees* with ground truth
3(a) # 3moNews | # CycloneIdai did not damage fuel pipeline , { Zimbabwes } energy minister has said # 3mob [#0 | nloc]
3(b) LATEST : Details of # Earthquake victims . Names of those who lost their lives so far . Number of injured . Source : # { Mirpur } Police [#0 | nloc]
Case 4: True Negative. Model has *low* confidence. Enduser *disagrees* with ground truth
4(a) More footage from { Abaco } in the eye of Dorian . I just ca nt believe there s a newborn in this damaged structure now . Praying for this family . [#0 | nloc]
4(b) Friends in { Stuart } AB zones are activated for evacuation . [#0 | nloc]

Table 7: Examples illustrating the insight delivered by our NoNE explanation approach that reveal possible annotation errors of the ground truth. {} indicates words that are important in the explanation because they are assigned low probabiliy of contributing to the prediction of nloc (the no-location label). Underlining indicates ground truth location entities.

3(a), the word "Zimbabwes" was not used as a location in the context of the tweet, rather modifies "minister". Here, the NoNE approach is delivering the insight that the NER model is performing well on some edge cases. In example 3(b), "Mirpur" modifies "Police", and, as such, is not considered a location according to the IDRISI-RE annotation protocol (Suwaileh et al., 2023).

The fourth case (bottom of Table 7) are true negative predictions for which the model has *low* confidence and the enduser disagrees with the ground truth. The explanation words with low attribution scores are indicated by ({ }). These words contributed the least to the prediction of the 'nloc' tag. Example 4(a) is considered to have no location entities in the ground truth and our model agrees with this in *low* confidence. The word "Abaco" was considered by our NoNE approach to contribute the least to the prediction of 'nloc' and from context, the enduser considers "Abaco" as a location word. A similar case is observed in 4(b) where the NoNE approach highlights Stuart as the least contributor to the prediction of 'nloc' and is considered as a location word in the context. We consider these examples as possible annotation errors in the data. In an actual disaster scenario, these are the cases that we need to minimize.

6 Conclusion and Future Work

We have proposed an approach that generates word-level feature relevance explanations for seq2seq NER that is designed to be particularly helpful in cases in which no named entity (NoNE) is found. The approach leverages a class-of-input (COIN) token that reflects the lack of a named entity in the input sequence. We evaluated our NoNE approach using both automatic and manual validation, of which we demonstrated that our enduser is able to successfully differentiate true and false negatives given the explanation word and their immediate neighborhoods at a high rate. Lastly, we demonstrated the potential of our NoNE explanations in revealing false negatives, which are crucial for the disaster management location NER scenario that provides the larger context for our work, and also showed that the explanations can uncover issues with the ground truth annotations.

Future work should carry out a deeper investigation of this potential. We believe that with further work, the COIN token could also improve the NER performance of the model. As we did not optimize our models and we only used one type of seq2seq model, we believe there is room for improvement in this. We hope that our work inspires other researchers to consider developing explainability focused on negative cases. Furthermore, we want to incorporate user perspectives from disaster practitioners on the effectiveness of explainability focused on negative cases to improve our NoNE approach.

Limitations

We validated our explanations using manual evaluation. Other evaluation methods would solidify the usefulness of the generated explanations. Furthermore, we use saliency as explanation as this is what most explainable models use but other possibilities were not explored.

In the manual validation, we noticed that the enduser's judgement of the three-word phrase explanation is impacted by: the enduser's understanding of the kind of tweets are being investigated, the enduser's background knowledge of the locations and lastly the enduser's understanding of what the annotation protocol was in the first place. As the validation was done by the authors and not by an actual disaster manager, the effectiveness of our approach was not validated in a real scenario.

We focused only on sequence-level negatives in this study and did not discuss nor investigate ambiguous token-level negative cases where the model has a low confidence labeling a token as not-location. These token-level decisions do affect the sequence-level output but we did not investigate this relation.

Ethics Statement

For social media monitoring to detect disasters and to estimate the magnitude of the disaster event, we are interested in finding and labeling those tweets of people that are witnessing or experiencing the disaster and therefore gathering personal information about the locations where these witnesses are located. Most research on disaster-event social media analysis for including this current study is done on English tweet datasets while other languages spoken in often disaster-sensitive areas are not receiving attention (Moitra et al., 2022).

References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2021. Interpretability Analysis for Named Entity Recognition to Understand System Predictions and How They Can Improve. *Computational Linguistics*, 47(1):117–140.

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385, Online. Association for Computational Linguistics.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Onur Güngör, Tunga Güngör, and Suzan Uskudarli. 2020. Exseqreg: Explaining sequence-based nlp tasks with regions with a case study using morphological features for named entity recognition. *Plos one*, 15(12):e0244179.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 953–962, Hong Kong, China. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. TriggerNER: Learning with entity triggers as explanations for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8503–8511, Online. Association for Computational Linguistics.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1731–1751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aparna Moitra, Dennis Wagenaar, Manveer Kalirai, Syed Ishtiaque Ahmed, and Robert Soden. 2022. Ai and disaster risk: A practitioner perspective. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–20.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, Oskar van der Wal, Malvina Nissim, and Arianna Bisazza. 2023. Inseq: An interpretability toolkit for sequence generation models. *ArXiv*, abs/2302.13942.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Wenjuan Sun, Paolo Bocchini, and Brian D Davison. 2020. Applications of artificial intelligence for disaster management. *Natural Hazards*, 103(3):2631–2689.

Reem Suwaileh, Tamer Elsayed, and Muhammad Imran. 2023. Idrisi-re: A generalizable dataset with benchmarks for location mention recognition on disaster tweets. *Information Processing Management*, 60(3):103340.

Reem Suwaileh, Tamer Elsayed, Muhammad Imran, and Hassan Sajjad. 2022. When a disaster happens, we are ready: Location mention recognition from crisis tweets. *International Journal of Disaster Risk Reduction*, 78:103107.

Keyon Vafa, Yuntian Deng, David Blei, and Alexander Rush. 2021. Rationales for sequential predictions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10314–10332, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online. Association for Computational Linguistics.

Yu Wang, Yun Li, Ziye Zhu, Bin Xia, and Zheng Liu. 2019. Sc-ner: A sequence-to-sequence model with sentence classification for named entity recognition. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Andrea Zugarini and Leonardo Rigutini. 2023. Sage: Semantic-aware global explanations for named entity recognition. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.