Named Entity Identification and Recognition using Character Language Models and Zero-Shot Learning

Cindy Hao
University of Pennsylvania
cindyhao@seas.upenn.edu

Jeffrey Xiao
University of Pennsylvania
jxiao23@seas.upenn.edu

Abstract

The tasks of named entity identification (NEI) and recognition (NER) represent interesting problems in the field of machine learning. What constitutes a named entity, and how can we recognize whether a given token is named or unnamed? Furthermore, how can we classify entities into specific labels, especially labels that have not been seen before? This paper seeks to analyze patterns in characters to train a Character-level Language Model (CLM) that will categorize tokens as named or unnamed as well as label it into one of 18 different groups. We show that CLM's are relatively reliable in categorizing tokens into named and unnamed categories, and that through zero-shot learning, we can label entities with similar, if not greater, success than other models.

[1] [2] [3]

1 Introduction

The objective is of this paper is two-fold. First, we build and train a Character-level Language Model (CLM) (Yu et al., 2018) to determine whether a given token is a named entity or non-named entity – this is Named Entity Identification. We will show that a CLM is a strong and relatively reliable model for identifying and recognizing named entities. Second, we classify these named entities into one of 18 different labels using zero-shot learning (Zhou et al., 2018) – this is Named Entity Recognition. This allows named entities to be labeled without annotated data and to be be identified as new types.

2 Problem Definition and Algorithms

2.1 Task Definition

For the input, we used the OntoNotes 5.0 Release, which contained 4 different training datasets, each with a different subset of annotated labels. The first dataset contained no annotated labels, the second had 6, the third had 12, and the final had 18.

Each dataset contains two relevant pieces of information: the label and the token itself. The tokens vary in terms of content - including both named and unnamed entities. Labels vary in topic and content, such as PERSON, ORGANIZATION, GPE (geo-political entity), WORK_OF_ART, LAW, PERCENT, and others (Table 5).

2.2 Algorithm Definition

We will be implementing this project in 2 main parts. The first part is Named Entity Identification, which will require determining which words are named entities. The second part will be zero shot entity type classification.

Regarding the first part, we will use a model where we train entity CLM on a list of entity tokens and a nonentity CLM on a list of non-entity tokens. The examples over this model will allow us to learn a score of how likely it is a sequence of characters forms an entity. We can split each work into characters in order to determine the perplexity using entity and nonentity CLMs. We propose

[1] Link to Video: https://bit.ly/37JiIGt
[2] Link to GitHub Repo: https://bit.ly/37CTMjC
[3] Link to Project Folder: https://bit.ly/37C7Wla

to experiment with an N-gram model using SRILM (Stolcke, 2004). The choice of the N-gram lamguage model using SRILM was because of evidence that this model outperforms CBOW, skip-gram, and log-billinear (Yu et al., 2018) for the English language.

We would then use the two learning models to predict future classifications. We create 2 N-gram models, one for entities and another for non-entities. For each token we predict, we calculate the complexity with the entity and non-entity model. We find the model that gives the lower complexity in order to label each token. We write to a text file 1 if we predict this as an entity and 0 otherwise. To calculate the accuracy, we compare our predictions to the true y-labels.

The second part of the project will require entity typing. We define a type as a conceptual container that binds entities together to form a coherent group. For this component, we will utilize a zero shot entity typing approach that can flexibly identify newly defined types. Specifically, we will use a RoBERTa model. With a mention in a sentence and a taxonomy of entity types and their definitions, the zero shot entity typing system will identify the set of types that are appropriate for the mention of the word in that context. Here, we used data about Wikilinks. The data was stored as a mapping from a title to a list of sentences. We found titles that fell into each of the 18 categories and used the sentences to create vectors.

2.3 Expectations

We expect the N-gram to yield a relatively high accuracy since due to its stellar performance which has been found to be "remarkably close to the result of state-of-the art NER systems" (Yu et al., 2018).

The task of named entity recognition was tougher since English is a confusing language and the type of even the same word can greatly vary depending on usage. Furthermore, unlike NEI which was a binary classification, this required us to classify each entity into 18 labels which meant we needed much more complex representations. Therefore, our hopes for this part of the task weren't as high as for NEI. While using a RoBERTa model with a token classification head seemed like a promising start, there was uncertainty in whether our examples from WikiLinks were representative of the OneNotes dataset. Additionally, there was also ambiguity as to whether cosine similarity, a measure document similarity in text analysis, would be robust to also work on individual tokens.

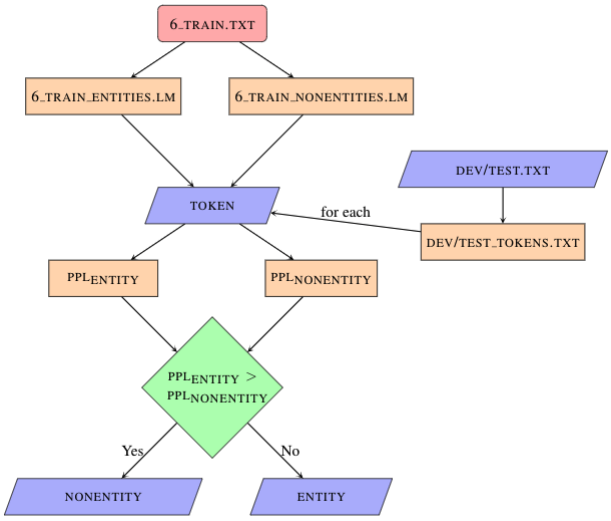

Figure 1: NEI training process on 6-label learning model

3 Experimental Evaluation

3.1 Methodology

To begin the identification process, we first separated tokens into named entities and non-named entities to create individual learning models. These learning models were based off of characters. For instance, the token "REPORTER" was separated into "R E P O R T E R". Each of these N-gram models were implemented using the SRILM library with an order of 6.

To predict a given token, we ran the both the entity and nonentity N-gram CLM's to produce 2 perplexity scores. Classification between entity and non-entity was determined

based on the lower perplexity score (Figure 1).

We evaluated our language model using an accuracy score from the SciKitLearn library. We used this to compare the predicted label to the actual label of each token.

The entity classification process was trained on a subset of the Wikilinks Dataset. The dataset was organized such that every title was mapped to a list of sentences that had linked to the title's wikipedia page. For example, the title "Suicide" would be mapped to a list of sentences that contained the word "Suicide" at some point within the sentence.

We selected around 10 of the most popular (i.e, most linked to) titles for each of the 18 entity labels. We then used a subset of the sentences (up to 20) that were mapped to each of the selected titles to create an embedding for the label itself. In order to create the embedding, we created a RoBERTa model with a token classification head on top for Named-Entity-Recognition (NER) tasks. This model is a Pytorch `torch.nn.Module` subclass and consisted of a linear layer on top of the hidden states output. We used the hidden states from the output (which was list of vectors) to find our vector, and the final vector we used for each sentence was the n^{th} vector in the list of vectors from the output hidden states, where n was the position of the title in the sentence. If the entity is "Suicide" and the sentence is "I do not think suicide is good", we would use the 5th vector (0 indexed) since the word is at index 5. For each of the labels, we created the vectors for every sentence in each of the chosen titles. We then averaged all of those vectors to find a single vector for each label. For example, if for the "PER" label had 10 titles associated with it and each of the 10 titles had 20 sentences, we would be the roBERTa model to create 200 vectors and find the average of those for the final vector representation of "PER". Each of the vectors were of length 768.

Finally, we wanted to predict the test file with the 18 labels using our model. The OntoNotes 5.0 Data included sentences with each of the words on separate lines. For each word, we found the sentence the word was in and the position of the word in the sentence. We then were able to find vectors of the same dimension for each of the words in the file. This, of course, depended on the context (the sentence the word was in). Finally, we compared each of the the words' vectors using cosine similarity with each of the 18 labels' vectors. We labeled each word with the label that corresponded to the vector it was most similar to. This became our final prediction for the entity recognition task.

3.2 Results

For the natural language identification portion, we repeated the prediction process using the 6, 12, and 18 label training sets. As shown in Tables 1, 2, and 3, our dev and testing accuracies were consistently in the 90 percent range. This is similar to the 92.8 accuracy score of the N-gram model using SRILM used on the English language from (Yu et al., 2018).

For the natural language recognition portion, as shown in Table 4, our accuracy scores were in the mid-high 80 percent range, suggesting that our model was able to classify named entities into 18 different labels quite successfully. This included both non-entity and entity labels.

Dataset	Accuracy (%)
6-LABEL DEV	93.89
6-LABEL TEST	94.38
12-LABEL DEV	93.67
12-LABEL TEST	93.83
18-LABEL DEV	91.10
18-LABEL TEST	91.05

Table 1: NEI accuracies trained on 6-label learning model

3.3 Data Analysis

One limitation of our model was that in named entity recognition, we did not account for the uniqueness of words. This would mean if a word, say "MEXICO", showed up multiple

Dataset	Accuracy (%)
6-LABEL DEV	92.51
6-LABEL TEST	92.78
12-LABEL DEV	92.98
12-LABEL TEST	93.48
18-LABEL DEV	91.34
18-LABEL TEST	91.65

Table 2: NEI accuracies trained on 12-label learning model

Dataset	Accuracy (%)
6-LABEL DEV	90.53
6-LABEL TEST	90.71
12-LABEL DEV	91.09
12-LABEL TEST	91.50
18-LABEL DEV	93.81
18-LABEL TEST	94.13

Table 3: NEI accuracies trained on 18-label learning model

times in the training data and had a consistent label and was predicted right, it would be predicted right every single time. This would artificially bump up the accuracy score since "MEXICO" would be classified multiple times correctly. Taking a look at our dev labels, some words, such as "CHINA", show up over 100 times. However, an argument as to why this could be acceptable would be that the text is representative of word frequencies in real world data. That is, if certain commonly words can be classified correctly, it is beneficial to have a model that predicts these common words correctly. That is, what matters is what percentage of words that show up in everyday test is predicted correctly (as opposed to what percentage of unique words is identified correctly). This counterargument would hold as long as our data is representative of real world text.

Another limitation is the process used to predict the label when we found the most similar vector with cosine similarity. This is because language is difficult to be modeled, especially the English language where different words can mean different things in different contexts. Additionally, the vectors we

NER Classification	Accuracy (%)
WITH NEI ACTUAL	89.5
WITH NEI PREDICTED	85.36

Table 4: NER accuracies trained on Wikilinks Data, predicted with 18 labels

chose from the WikiLinks data could have not been representative of the label itself. We picked the most frequently linked to ones, but there of course were issues. For example, for the "PER" label, we had ["GOD", "JESUS", "BARACK OBAMA", "GEORGE W. BUSH", "ADOLF HITLER", "MICHAEL JACKSON", "WILLIAM SHAKESPEARE", "ELVIS PRESLEY", "JOHN F. KENNEDY", "LEONARDO DA VINCI", "ARISTOTLE", "ALBERT EINSTEIN"], all of whom were male.

4 Conclusions & Future Work

Overall, we have shown that two different aspects of the natural language processing model. First, the English language contains numerous character patterns, so using the character-level sequences can help to identify named entities - even those that haven't been seen or trained on before. Our CLM performed in the 90% range on numerous dev and test samples, suggesting that this is a strong model to use and consider going forward. Second, it's possible to use a RoBERTa model, as shown, to classify entities into various labels. Our RoBERTa model trained with token classification performed successfully on several datasets with little training, suggesting that zero-shot learning was in action.

In the future, we hope to run a more comprehensive test of both our NEI and NER models by incorporating additional test sets (with > 18 labels). Additionally, our current model looks at an entity in the context of the entire sentence, with entities and nonentities. One future discussion would be to only look at entities in the context of other entities only rather than both.

Type	Description
PERSON	People
NORP	Nationalities, religious, political groups
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states
LOC	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Objects, vehicles, foods, etc
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language
DATE	Absolute or relative date or periods
TIME	Times smaller than a day
PERCENT	Percentage, including '%'.
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	"first", "second", etc.
CARDINAL	Numerals that do not fall under another type.

Table 5: Entity Types (spa)

References

Annotation specifications · spacy api documentation and schemes used for labels, tags and training data.

Andreas Stolcke. 2004. Srilm — an extensible language modeling toolkit. *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2.

Xiaodong Yu, Stephen Mayhew, Mark Sammons, and Dan Roth. 2018. On the Strength of Character Language Models for Multilingual Named Entity Recognition. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-Shot Open Entity Typing as Type-Compatible Grounding. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.