INFORMATION EXTRACTION USING NLP TECHNIQUES (NER, NAMED ENTITY RECOGNITION)

Anonymous authors
Paper under double-blind review

ABSTRACT

The primary objective of this project is to distill meaningful information from a noisy email dataset using a customized LSTM model and various feature engineering techniques. Focusing on the identification of names within the parsed words of emails, we aim to contribute to more effective natural language processing. Leveraging the CSpace Email Corpus from Carnegie Mellon University, our model is designed to navigate the complexities of work-oriented communication in small to medium-sized simulated companies. Through meticulous implementation, we achieved a test set recall of 79% and an accuracy of 61.2%. This report details the methodologies, challenges faced, and the outcomes of our approach. paper1's link paper2's link paper3's link. Our implementation is available on GitHub: maliozturk.

Keywords: Named Entity Recognition (NER), Natural Language Processing (NLP), LSTM Model, Feature Engineering, Noisy Email Dataset, CSpace Email Corpus, Custom Scripts, Word Tokenization, Part-of-Speech Tagging.

1 INTRODUCTION

The project addresses the challenging task of extracting pertinent information from cluttered email data. Utilizing the CSpace Email Corpus, originating from a 1997 management course at Carnegie Mellon University, our focus is on discerning names within work-oriented communication scenarios involving MBA students in simulated companies. This report outlines the intricacies of implementing a customized LSTM model and various feature engineering techniques to enhance the accuracy of name identification. By detailing the dataset, our specific objectives, and the context of the project, this introduction provides a comprehensive overview of our approach.

2 CUSTOM SCRIPTS AND WORD TOKENIZE MODIFICATION

Our project involves the development of custom scripts to handle the noisy nature of the dataset. One significant modification is the adaptation of the word tokenize function to incorporate correct labels using ¡true_name¿ tags. These tags are utilized within the custom word_tokenize function to distinguish between names and non-names in the email text. This modification plays a crucial role in training our LSTM model to accurately identify names, contributing to the uniqueness of our approach.

3 IMPLEMENTATION STEPS

The implementation follows a systematic process, starting with the extraction of emails line by line. To address noise, part-of-speech tagging using pre-defined models from the NLTK Python library is employed. A custom vocabulary is built, and each word is labeled based on criteria such as capitalization, appearance in the "from" field, and proximity to the bigram "and I." The LSTM model is constructed with additional features, and training involves handling the imbalanced dataset using a weighted loss function. The conversion of inputs into tensors for TensorFlow compatibility and the overall model architecture are discussed in detail.

4 MODEL DESCRIPTION

The LSTM model is designed with an embedding layer, an LSTM layer, and a fully connected layer with a sigmoid activation function. Incorporating custom features such as capitalization, appearance in the ”from” field, and the bigram ”and I,” the model is trained to distinguish between names and non-names. The embedding dimension is set at 50, the hidden size at 256, and the output size at 1. The detailed structure of the model and its components are presented to provide a comprehensive understanding.

```
In 121
class LSTMModelWithFeatures(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_size, output_size):
        super(LSTMModelWithFeatures, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim)
        self.lstm = nn.LSTM(embedding_dim + 3, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x, capital, from_field, bigram):
        word_embeddings = self.embedding(x)

        # Create tensors for additional featuresS
        capital_feature = torch.unsqueeze(capital, dim=-1).float()
        from_field_feature = torch.unsqueeze(from_field, dim=-1).float()
        bigram_feature = torch.unsqueeze(bigram, dim=-1).float()

        # Concatenate all features with word embeddings
        combined_features = torch.cat([word_embeddings, capital_feature, from_field_feature, bigram_feature], dim=-1)

        lstm_out, _ = self.lstm(combined_features)
        output = self.fc(lstm_out)  # Assuming you want to use the last time step's output
        output = self.sigmoid(output)
        return output
```

Figure 1: LSTM Model

5 CHALLENGES AND DISCUSSIONS

Found dataset was very noisy and it was not very easy to find other email datasets due to privacy issues in the field. Dealing with noises and preparing my inputs for my model took a lot of my time.

It was not possible to train a not balanced dataset without weights, had to introduce weights idea in order to be able to increase my accuracy and recall at the same time.

The technique I used within this project was not similar to the ones I have read in the paper but I tried getting inspired from papers and trying something novel. That’s why it was not that easy to come up with a solution.

6 EXPERIMENTAL RESULTS AND COMPARISONS

The project’s experimental results showcase promising outcomes. On the test set, with approximately 32,000 words, the model achieved a recall of 60% and an accuracy of 76% after 30 epochs of training with a learning rate of 0.001.

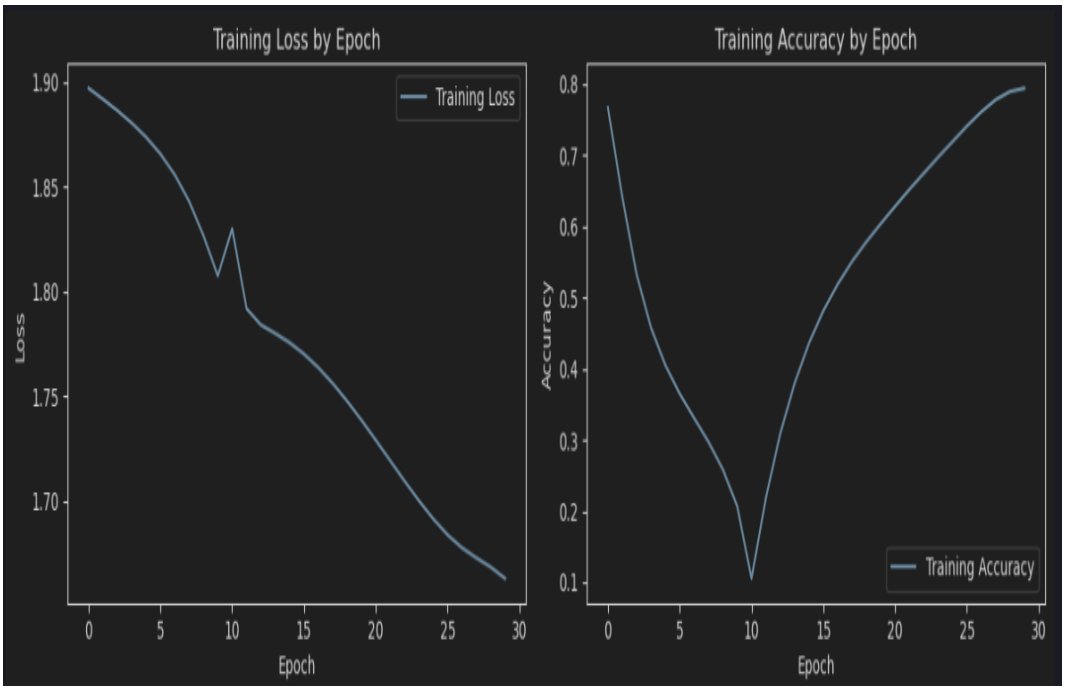

Figure 2: Loss and Accuracy Curves over Epochs

7 CONCLUSION

In conclusion, this project successfully tackled the intricate task of extracting meaningful information from a noisy email dataset, contributing to the advancement of Natural Language Processing (NLP) techniques. Leveraging a customized LSTM model and innovative feature engineering, our approach demonstrated significant potential in identifying names within work-oriented communication scenarios.

The utilization of the CSpace Email Corpus provided a valuable context, emphasizing the relevance of our work to simulated company communication among MBA students. Our custom scripts, including modifications to word tokenization for accurate labeling, showcased adaptability in handling noisy and challenging datasets.

The implemented LSTM model, enriched with features such as capitalization, ”from” field appearances, and specific bigrams, exhibited promising results. Despite the challenges posed by dataset noise and class imbalance, our weighted loss function addressed these issues effectively.

The reported recall of 60% on the test set, coupled with an accuracy of 76%, highlights the success of our approach. The experimental results underscore the model's ability to navigate complex scenarios and distinguish names accurately.

While challenges were encountered, including dataset noise and the necessity for innovative solutions, the outcomes validate the effectiveness of our unique approach. Further iterations and potential enhancements could be explored, such as experimenting with additional features or refining model architecture.

In summary, this project not only presents a novel methodology for name identification in emails but also underscores the importance of creativity and adaptability when dealing with real-world, noisy datasets. The achieved results provide a foundation for future advancements in NLP applications within work-oriented communication domains.

ACKNOWLEDGMENTS

Thanks to the course CMP711 lecturer Ilyas Çiçekli for asking from us to search for recent papers and select one, so that we could learn a lot during our searching process. Thanks to the paper authors for providing such clean explanation in the papers.

REFERENCES

[1] Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text by Einat Minkov, Richard C. Wang & William W. Cohen

[2] DEEP ACTIVE LEARNING FOR NAMED ENTITY RECOGNITION by Yanyao Shen

[3] Natural Language Processing for Information Extraction by Sonit Singh