## HW1 Writing Questions

Zheng Wang

**Q1.1**

Some sentences have Named Entity, but some do not.

**Q1.2**

199 documents in the training data have invalid labelings.

**Q1.3**

I noticed that most of NER tags are "O", and then comes to "B-LOC" and so on, as I what I have expected. The difficulty is the imbalanced training dataset. A dataset with a high proportion of "O" tags can be imbalanced, meaning that there are far more "O" tags than named entity tags (e.g., B-PER, I-ORG, etc.). This imbalance can affect the performance of NER models, as they may become biased toward predicting "O" tags and may struggle to recognize named entities effectively.

**Q1.4**

Unexpectedly, I noticed that "of", "The", "the" are the most common tokens. The data imbalance can make it challenging for NER models to learn to recognize and classify named entities accurately when most of "of", "The", "the" are not appearing in named entity.

**Q1.5**

The plot looks like a triangle with dots inside. Most of the documents have length from 0 to 60 and most numbers of named entities are from 0 to 10. So there is not much relationship between named entities and document length. The number of named entities will not increase as the document length increases.

**Q1.6**

The argument is that the tokens with type of "ORG" should be increased in order to be equal to the distribution of other other types of tokens. As is shown in the graph, the percentage of "ORG" is less than others. In order to help the model learn and make equal predictions on each type of tokens, an equal distribution of all the types should be provided.

**Q2.1**

1. In which situations did the system perform effectively?

The HMM can perform well when NER tags are well-defined and distinct from each other. For example, if the NER task involves recognizing simple entity types like "PERSON," "ORG," or "LOCATION," the HMM can work effectively as it can capture transitions between these tags. With a large and diverse training dataset, the HMM can learn the transition and emission probabilities effectively. If there are enough examples of transitions between tags and observations, the model can make accurate predictions.

1. when did it encounter challenges? For instance, does the model excel in predicting certain NER tags more than others?

In cases where the dependencies between Named Entity Recognition (NER) tags are complex and not easily represented as first-order Markov chains, Hidden Markov Models (HMM) can struggle to capture long-range dependencies and understand the context and relationships between multiple entities in a sentence. Additionally, when there is significant ambiguity or overlap between NER tags, such as distinguishing between "ORG" and "LOCATION" when an organization name resembles a place name, the HMM may have difficulty making accurate distinctions.

Handling rare or unknown words not seen in the training data is a challenging aspect of NER. NER tags that occur frequently in the training data tend to be predicted more accurately by the HMM because the model has had more exposure to these tags and can learn better transition and emission probabilities for them. For instance, tags like "O" are often high-frequency tags and are usually predicted well. Tags representing simple and distinct entity types, such as "PERSON," "LOCATION," or "ORGANIZATION," are easier for the HMM to predict accurately because they have clear boundaries, and the model can capture transitions between them more effectively.

On the other hand, tags representing rare or complex entity types that occur infrequently in the data may be predicted less accurately by the HMM. This is because the model might not have had enough training examples to learn robust probabilities for these tags, making them more challenging to predict accurately.

1. Could you offer any hypotheses about the reasons behind these patterns and suggest potential improvements?

The use of smoothing techniques, such as add-k smoothing, can help mitigate data sparsity issues and improve the performance of the HMM. Smoothing ensures that even if certain transitions or emissions are rare or unseen in the training data, the model can assign non-zero probabilities to them during prediction.

Complex NER tasks with intricate dependencies between tags may benefit from more advanced sequence labeling models, such as CRFs or LSTMs. Increasing the size and diversity of the training data can help the HMM better capture tag transitions and emissions.

**Q2.2**

Unknown words are words that the model has not seen during training, and they can occur frequently in real-world text data. If not handled properly, unknown words can lead to errors in NER. Suppose the model encounters the word "Zyxxel" during testing, but "Zyxxel" was not seen in the training data. Without proper handling, the model may assign a default tag (e.g., "O" for non-entity) to "Zyxxel," even though it might be a valid organization name. This can result in a false-negative error.

An effective approach is to treat unknown words as a special token, such as "<unk>." When the model encounters an unknown word, it can tag it as "<unk>" or make a probabilistic prediction based on context. For example, if "<unk>" appears after "at" or "in," it might be tagged as a location.

By incorporating context, the model can better adapt to unseen words and reduce the chances of false negatives. Additionally, techniques like subword tokenization or character-level modeling can help the model recognize parts of unknown words that resemble known entities. This approach allows the model to handle unknown words more gracefully.

Smoothing techniques are used to handle data sparsity issues and improve the model's robustness and generalization. They impact the model's probability estimates for transitions and emissions. Without smoothing, the model may assign zero probabilities to unseen transitions or emissions. For example, if the training data has no occurrences of the transition "B-ORG" to "I-ORG," the model would assign a probability of zero to this transition. This can lead to errors during prediction. Add-k smoothing adds a small constant (k) to each count, ensuring that even unseen transitions and emissions have non-zero probabilities. For instance, if "B-ORG" to "I-ORG" transitions were unseen in training, add-k smoothing assigns them a small positive probability. This reduces the risk of zero probabilities and makes the model more robust.

**Q3.1**

I evaluated the MEMM using a validation dataset to assess its performance. The primary metric used was the mean F1 score. I trained the MEMM model and then assessed its performance on a validation dataset. The mean F1 score obtained was 0.5121, indicating moderate performance. This score represents the model's ability to correctly identify named entities in text. In the process, I experimented with various factors to understand their impact on model's performance:

1. I conducted extensive feature engineering to extract relevant information from the text data. These features included part of speech, capitalization, token length, the presence of special characters, token frequency, and more. These features aimed to capture different aspects of the text that might be indicative of named entities.
2. I fine-tuned the hyperparameters of the MaxEnt Classifier used within the MEMM model. This optimization involved adjusting the number of iterations to find the best setting for the model.
3. To gain insights into the model's performance, I conducted an error analysis. This involved examining where and why the model made errors, such as false positives and false negatives.

**Q3.2**

The MaxEnt Classifier in the MEMM model considered several features when making predictions, and some features stood out as particularly important:

Part of Speech: The part of speech of a token emerged as a crucial feature. It helps the model understand the grammatical role of each word in a sentence is vital recognizing named entities.

Capitalization: The capitalization feature proved significant. It assists in identifying proper nouns, which are often indicative of named entities.

Token Length: The length of a token was found to be relevant, as entities can vary in length.

Special Characters: This feature helped in identifying entities, such as numbers or punctuation.

Token frequency, both within the document and across the entire dataset. This feature assigns higher importance to words that occur frequently, as they are more likely to be named entities. The prominence of POS and capitalization features aligns with their ability to provide critical context for identifying named entities in text. Token frequency also contributed significantly, ensuring that commonly occurring words were given more attention in the model's predictions.

Experiments with contextual features, such as the previous and next words, showed varying impacts on performance, suggesting that their usefulness depended on the specific context.

Error Analysis:

The system faced challenges when dealing with less common or unconventional named entities, leading to false negatives. These errors occurred when the model failed to recognize entities that deviated from standard patterns. False positives were observed when the model incorrectly identified non-entities as named entities. These errors often resulted from patterns in the text that resembled real entities.

**Q3.3**

The MEMM system exhibited strengths when handling frequently occurring named entities with standard patterns, leveraging features like capitalization and POS.

Challenges arose when the model encountered less common or unconventional named entities. This resulted in false negatives, where entities outside standard patterns were missed.

False positives occurred when the system incorrectly identified non-entities as named entities due to overgeneralization.

We could improve the system:

1. Further exploration of feature engineering, potentially incorporating word embeddings or contextual embeddings, enhance model's ability to capture nuanced information from the text.

2. Implementing stronger regularization in the MaxEnt Classifier could mitigate overfitting and reduce false positives.

3. Expanding the training data like data augmentation, which includes synonym replacement or paraphrasing, could help the model generalize better to diverse entity formats.

4. Careful fine-tuning of hyperparameters, guided by a detailed error analysis to address specific error patterns, is essential for enhancing the system's performance.

**Q4.1**

When comparing the results of my HMM and MEMM on the validation dataset, it is evident that the MEMM outperforms the HMM. The MEMM achieved a significantly higher mean F1 score of 0.5121, indicating better performance in recognizing named entities, compared to the HMM's lower F1 score of 0.4583.

The reason behind the MEMM's superior performance lies in its ability to consider more complex features and capture contextual information. Unlike the HMM, which relies primarily on transition probabilities and observable emissions, the MEMM incorporates features like part of speech, capitalization, token length, and token frequency. These additional features enable the MEMM to make more informed predictions by considering linguistic and structural cues in the text. Additionally, the MEMM's flexibility in modeling dependencies between observations and labels within a sequence allows it to adapt better to the complexities of named entity recognition.

**Q4.2**

In error analysis, I observed certain error patterns made by the HMM that the MEMM was able to mitigate. One common error pattern in the HMM was its struggle to recognize named entities that deviated from typical patterns or structures. For instance, in the sentence "Apple Inc. is located in California," the HMM might incorrectly identify "Inc." as an organization due to its appearance as a standalone word. This is because the HMM relies heavily on transition probabilities and lacks the context-awareness to discern that "Inc." is a part of the organization name "Apple Inc."

Another error pattern observed in the HMM was its difficulty in distinguishing between entities and non-entities when faced with certain ambiguous words. For example, the word "Bank" could refer to a financial institution or simply a riverbank, and the HMM struggled to disambiguate based on context alone.

The MEMM, on the other hand, was able to address these error patterns by considering features like capitalization and part of speech, which provided valuable contextual information. It could recognize that "Inc." in "Apple Inc." is an organization due to its capitalization pattern and part of speech as a proper noun. Additionally, the MEMM's flexibility in learning from the training data allowed it to adapt to different contexts and make more informed predictions.

**Q4.3**

In error analysis for the MEMM, I observed certain error patterns that the HMM did not exhibit. One notable error pattern in the MEMM was related to its sensitivity to the training data. When the training data did not adequately cover specific named entity variations or uncommon patterns, the MEMM was more prone to false negatives. For instance, if the training data lacked examples of certain rare location names, the MEMM might fail to recognize them in the validation dataset.

Another error pattern in the MEMM was its susceptibility to misclassifying entities in noisy or unstructured text. In cases where the text was poorly formatted or contained errors, the MEMM could make incorrect predictions.

These error patterns in the MEMM can be attributed to its dependence on the quality and diversity of the training data. Unlike the HMM, which relies on simpler transition probabilities, the MEMM's performance is influenced by the richness of the features and the representation of the data in the training set.

To mitigate these error patterns, improving the training data quality and considering data augmentation techniques could be beneficial. Additionally, fine-tuning the MEMM's hyperparameters and incorporating more advanced features might help enhance its robustness.