## HW1 Writing Questions

Zheng Wang

## 5740 extra

**Q6.1**

NER is a NLP task where the goal is to identify and categorize named entities in text. What sets it apart from traditional NER is that in nested NER, these entities can have a hierarchical or nested structure. This means that one entity can be contained within another entity, like nesting dolls.

The key differences between nested NER and traditional NER are as follows:

Hierarchical Structure: In nested NER, named entities can nest inside each other, creating a layered structure. For example, in the sentence "The spokesperson for the Ithaca Zoo said...", "the Ithaca Zoo" is an outer entity containing an inner entity "Ithaca." Traditional NER doesn't consider such nested relationships.

Entity Dependencies: Nested NER needs to account for dependencies between entities. This means that an inner entity can depend on an outer entity inside-out dependency and outside-in dependency. In traditional NER, these dependencies are typically not considered.

Multi-Stage Framework: Nested NER often uses a multi-stage approach, breaking down the recognition process into multiple sequential stages. Traditional NER typically uses a single-stage framework where entity boundaries and categories are determined simultaneously.

Error Propagation: Multi-stage nested NER models may suffer from error propagation, where mistakes made in one stage can affect subsequent stages.

Tagging Scheme: Nested NER can use different tagging schemes compared to traditional NER. While traditional NER often uses the BIO (Begin, Inside, Outside) tagging scheme, nested NER may use tuple-based schemes to represent the hierarchy of entities.

What makes nested NER more challenging is the need to handle this nested structure, account for entity dependencies, manage a multi-stage framework, deal with error propagation, and adapt to different tagging schemes. These complexities require more sophisticated models and algorithms to accurately identify and categorize entities within a nested context, making it a more intricate task compared to traditional NER.

**Q6.2**

Based on the models trained in Parts 2 and 3, they are likely to face challenges when dealing with sentences containing nested named entities. These models are primarily designed for flat NER tasks and may not inherently handle the hierarchical or nested structure of entities well.

Example 1: Sentence: "The CEO of Apple Inc., Tim Cook, visited San Francisco."

Outer Entity: "Apple Inc."

Inner Entity: "Tim Cook"

Observations: The models may recognize "Apple Inc." as an organization and "San Francisco" as a location, but it might struggle to identify "Tim Cook" as a person nested within "Apple Inc."

Example 2:

Sentence: "The book 'Harry Potter and the Sorcerer's Stone' by J.K. Rowling is a classic."

Outer Entity: "Harry Potter and the Sorcerer's Stone" (Book Title)

Inner Entity: "J.K. Rowling" (Author)

Observations: The models may correctly identify "J.K. Rowling" as an author but could struggle to recognize the entire book title as a nested entity within the author's name.

Example 3:

Sentence: "In 'The New York Times' article by John Doe, the author discusses climate change."

Nested Entities:

Outer Entity: "The New York Times" (News Outlet)

Inner Entity: "John Doe" (Author)

Observations: The models might identify "The New York Times" as a news outlet but may not recognize "John Doe" as the author within the context of this sentence.

In these examples, the models are likely to pick up some of the entities, such as organizations, locations, and authors, but may struggle to identify the nested entities accurately. The information missed by the models would be the nested relationships, where one entity is contained within another. These models may not have been trained explicitly to handle nested named entities, and their performance might not be optimal for such cases.

In comparison to my initial hypothesis, I expected the models to perform reasonably well in recognizing flat entities like organizations and locations but struggle with nested entities. The observations align with this hypothesis, as the models tend to identify flat entities while missing the hierarchical relationships between entities.

**Q6.3**

1. Layered-based Approach:

In the layered-based approach, the task of nested entity recognition is divided into different layers or stages. Each layer focuses on capturing a specific level of nested entities. For example, one layer may aim to identify inner entities, while another layer targets outer entities. The layers are connected sequentially, with the output of one layer serving as input to the next layer. This allows the model to build a hierarchical understanding of nested entities. These approaches can capture entity dependency, which is the relationship between different entities in a sentence, especially entities with nested structures.

1. Region-based Approach:

In the region-based approach, the input sentence is divided into regions or spans, and the goal is to classify each region as a specific named entity or part of one. These approaches often adopt a single-stage framework, where entity boundaries and categories are predicted simultaneously for each region. Region-based models are suitable for sentences with complex nested structures, as they consider all possible spans in the sentence.

These two approaches, layered-based and region-based, offer different strategies for capturing nested entities. Layered-based models focus on building a hierarchical understanding of entities by processing them in different stages, while region-based models directly classify regions within a single stage to handle complex nested structures in sentences.