Natural language processing

About me

- Full name → Pooria Rahimi
- Student number → 99521289

Theory questions

▼ **A) Explain the challenges of NER.**

Named Entity Recognition (NER) faces challenges include entity name confusion, typos, categorization ambiguity, reference variance, and contextual problems.

1. **Ambiguity**: Many words in a language can have multiple meanings based on context. For example, "Paris" can refer to the city in France or a person's name. Disambiguating such cases accurately is challenging.
2. **Variability**: Entities can be expressed in various forms, such as different spellings, abbreviations, or synonyms. For instance, "New York City" can be referred to as "NYC" or simply "New York."
3. **Named Entity Overlap**: Entities can overlap with each other. For instance, "John Smith" could be referred to as a person and "Smith Enterprises" as an organization in the same text.
4. **Out-of-Vocabulary Entities**: NER systems often struggle with entities that are not present in their training data. This is particularly challenging in domains with specialized terminology or emerging entities.
5. **Context Dependency**: The same entity may have different categories based on the context. For example, "Apple" could refer to the company or the fruit, depending on the context.
6. **Rare Entities**: Entities that appear infrequently in the training data pose a challenge for NER systems as they may not have enough examples to learn from effectively.
7. **Noisy Text**: Text data from sources like social media, forums, or user-generated content often contain noise, such as spelling errors, grammatical mistakes, slang, and abbreviations, making it harder for NER systems to accurately identify entities.
8. **Multilingualism**: NER becomes even more complex when dealing with multilingual text, where entities may differ significantly in structure and form across languages.

▼ **B) Explain the impact of the concept of text on the accuracy of NER systems.**

The concept of text has a significant impact on the accuracy of Named Entity Recognition (NER) systems. Here's how:

1. **Quality of Text**: The quality of the input text is crucial. Texts with a lot of spelling mistakes, grammatical errors, or slang can lead to inaccuracies in entity recognition.
2. **Context**: The context in which a word is used in a text can greatly affect the accuracy. For example, the word "Apple" could refer to the fruit or the technology company, depending on the context.
3. **Domain-Specific Language**: Texts from specific domains or industries may contain jargon or terms that are not commonly used in everyday language. If the NER system is not trained on such texts, it may not accurately recognize these entities.
4. **Language and Culture**: The language of the text and cultural nuances can also impact accuracy. Different languages have different syntax, and a model trained on English text may not perform well on text in other languages.
5. **Length of Text**: The length of the text can also impact the accuracy of NER systems. Longer texts may provide more context, but they also increase the complexity of the task.
6. **Ambiguity**: Texts often contain ambiguous terms. Without sufficient context, it can be challenging for the NER system to accurately identify the correct entity.

7. **Entity Representation**: How entities are represented in the text can affect recognition. For example, dates can be written in various formats, and if the system is not trained to recognize all formats, it may miss some.

In conclusion, the concept of text plays a pivotal role in the performance of NER systems. It's important to train these systems on a diverse range of high-quality texts to ensure high accuracy.

▼ C) Explain how CRFs improve the constraints of HMMs.

Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) are both used for sequence labeling tasks, such as part-of-speech tagging, named entity recognition, and speech recognition. While HMMs model sequences as a chain of hidden states with observable emissions, CRFs are discriminative models that directly model the conditional probability of a sequence given input features.

1. **Independence Assumptions**: HMMs make strong independence assumptions. They assume that the hidden state at time `t` depends only on the state at time `t-1` (Markov property), and the observation at time `t` depends only on the state at time `t`. CRFs, on the other hand, do not make these assumptions. In a CRF, the current state can depend on any number of previous states, not just the most recent one. This allows CRFs to capture long-range dependencies in the data, which HMMs cannot do.
2. **Observation Independence**: HMMs also assume that the observations are independent given the states, which is often not true in practice. CRFs do not make this assumption, allowing them to model complex dependencies between observations and states.
3. **Parameterization**: HMMs are generative models, meaning they model the joint distribution of observations and states. This requires estimating many parameters, which can lead to overfitting. CRFs are discriminative models, meaning they model the conditional distribution of the states given the observations. This requires fewer parameters, reducing the risk of overfitting.
4. **Label Bias Problem**: HMMs suffer from the label bias problem, where states with fewer outgoing transitions are more likely to be selected. CRFs do not have this problem because they normalize over the entire sequence of states, not just the next state.

Overall, CRFs provide a more flexible and powerful framework for sequence labeling tasks compared to HMMs, particularly in scenarios where complex dependencies and rich feature representations are important for accurate predictions.

> 💡 All HMMs can be converted to CRFs. But, the reverse does not hold true.

▼ D) Find on error in the labeling of each of the following sentences tagged with the Penn Treebank collection.

- I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN

  `Error: Atlanta/NN should be Atlanta/NNP`
- Does/VBZ this/DT flight/NN serve/VB dinner/NNS

  `Error: dinner/NNS should be dinner/NN`
- I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP

  `Error: have/VB should be have/VBP`
- Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

  `Error: Can/VBP should be Can/MD`

▼ E) Explain how the BIO tagging method is used for named entities. And check the difference of this method from IO tagging and BIOES tagging?

1. **BIO Tagging**
   - **B**: This tag indicates that the token (or word) is the start of a named entity.
   - **I**: This tag indicates that the token is inside or a continuation of a named entity.
   - **O**: Outside of any entity In the BIO tagging scheme, a token is labeled as 'B' if it marks the beginning of an entity, 'I' if it comes inside an entity, and 'O' if it is outside any entity.

> The **BIO tagging scheme** is a common method used in Named Entity Recognition (NER), which is a subtask of information extraction that seeks to locate and classify named entities in text into predefined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

2. **IO Tagging**
   - **I**: Inside of entity
   - **O**: Outside of any entity In the IO tagging scheme, there's no explicit way to denote the beginning of an entity. This can lead to ambiguity when entities of the same type immediately follow each other.
3. **BIOES Tagging**
   - **B**: Beginning of entity
   - **I**: Inside of entity
   - **O**: Outside of any entity
   - **E**: End of entity
   - **S**: Single entity The BIOES tagging scheme is an extension of BIO that adds explicit tags for the end of an entity ('E') and for single-token entities ('S'). This can help to resolve some of the ambiguities that can arise in the BIO scheme.

Here's an example to illustrate these three tagging schemes:

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

With these additional tags, BIOES tagging allows for more precise labeling of named entities, especially when dealing with multi-token entities. The main difference between BIOES tagging and BIO tagging is that BIOES tagging specifies the end of a named entity, which can help avoid ambiguity, especially when consecutive named entities occur.

The main difference between BIO tagging and IO tagging is that IO tagging only uses two tags: I (Inside) and O (Outside). IO tagging doesn't differentiate between the beginning and the inside of named entities. So, in IO tagging, every token inside a named entity is labeled as I, while in BIO tagging, only the first token is labeled as B and subsequent tokens are labeled as I.

The End😔