Task 1. Named Entity Extraction: Survey, Implement, and Apply

Q1:

**The paper I read**: Luo, L. *et al* (2018) *An attention-based BiLSTM-CRF approach to*

*document-level chemical named entity recognition. Bioinformatics*, 34(8), 2018, 1381–1388

(<https://academic.oup.com/bioinformatics/article/34/8/1381/4657076>)

**The state-of-art**: Attention-based bidirectional Long Short-Term Memory with a conditional random field layer (Att-BiLSTM-CRF) approach for document-level chemical NER

**Issues** to solve:

1. In the previous CRF-based chemical NER methods, the performance depends on the feature engineering for neural network architecture methods, which ius labor-intensive and skill-dependent.
2. Taggin inconsistency in sentence-level NER.

**Main contributions** of the work:

1. The attention mechanism that captures similar entity attention at the document level.
2. Domain features (i.e. POS) used in traditional NER methods with neural network architectures.

**Features:**

1. Word embedding:
   1. Low dimensional and dense, compared with the bag-of-words (BOW) representation.
2. Character embedding:
   1. The concatenation of the forward and backward representations from the BiLSTM.
   2. Contain rich structure information of the entity.
   3. Learn interior representations of the entity names and alleviate the out-of-vocabulary problem.
3. Additional features:
   1. POS and chunking
   2. Chemical dictionary feature

**Att-BiLSTM-CRF model**:

1. **Embedding Layer**: passing through this layer, a sentence will be represented as a sequence of vectors
2. **BiLSTM Layer**: Generate representation for each word vector: = [].
3. **Attention Layer**:
   1. Attention matrix A to calculate the similarity between the current target word and all words in the document
   2. Each element in A is derived by comparing the current target word representation with the word representation in the document: . The score is referred as an alignment function. Four alternatives: manhattan distance, euclildean distance, cosine distance and perceptron.
   3. Then a document-level global vector is computed as a weighted sum of each BiLSTM output :
   4. The document-level global vector and the BiLSTM output of the target word are concatenated as a vector [gt; ht] to be fed to a tanh function:
4. **Tanh Layer**: Predict confidence scores for the word having each of the possible labels:
5. **CRF Layer**: Decode the best tag path in all possible tag paths
   1. Matrix of scores P: column is the vector obtained by the Tanh Layer. represents the score of the tag of the word in the sentence.
   2. Tagging transition matrix T: represents the score of transition from tag i to tag j in successive words and represents the initial score for starting from tag j.
   3. The score of the sentence X along with a sequence of predictions y = (; ... ; ; ... ; ): s(X,y)=
   4. The softmax function is used to yield the conditional probability: p
   5. The objective of the model is to maximize the log-probability of the correct tag sequence

**Results:**

1. Euclidean distance as the alignment function performs the best on the CHEMDNER corpus.
2. Att-BiLSTM-CRF with document-level method performs better than that with sentence-level method. It also performs better than BiLSTM-CRF with sentence-level or document-level.

Q2:

In the scope of this paper, several intriguing techniques piqued my interest for potential implementation, including character embedding, Part-of-Speech (POS) tagging, chunking, attention mechanisms, and Conditional Random Fields (CRF). However, due to constraints, I ultimately managed to execute the implementation of a BiLSTM-CRF model, albeit without the integration of attention mechanisms. My feature extraction approach centered solely around word embeddings, forgoing the amalgamation of POS, chunking, dictionary features, word embeddings, and character embeddings.

Given my familiarity with CRF being limited, I devised a strategy to begin with the development and training of a BiLSTM-CRF model. My intention was to gauge its effectiveness in inference before venturing into more intricate techniques. To streamline the training process, I opted for a smaller dataset sourced from the Annotated Corpus for Named Entity Recognition by Abhinav Walia (<https://www.kaggle.com/datasets/naseralqaydeh/named-entity-recognition-ner-corpus>), as opposed to the CHEMDNER corpus or CDR task corpus referenced in the paper. However, it came as a surprise that the training of the chosen dataset consumed approximately 2 days to complete 10 epochs, even when leveraging a rented GPU for acceleration.

The specific configuration is as follows:

GPU: V100-32G Quantity: 1 VRAM: 32 GB

CPU: Intel Core Processor Instance Memory: 59 GB Cores: 12 cores

Regrettably, this extended training duration left me with insufficient time to incorporate additional techniques as intended.

Nevertheless, I did manage to successfully train a standalone BiLSTM model for comparative purposes. Notably, this training regimen concluded within a mere 30 minutes, even when conducted on a CPU.

Here’s the results I get:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1 Score |
| BiLSTM | 81.00% | 91.50% | 85.93% |
| BiLSTM-CRF | 80.39% | 92.44% | 85.99% |

Based on the results, I observed that the performance improvement of BiLSTM-CRF compared to BiLSTM is not substantial for the dataset I used. This might be attributed to the fact that the sequence labeling task within the dataset is relatively straightforward, and the interdependencies between labels are not notably intricate. Consequently, the added complexity introduced by the CRF layer might not yield a significant advantage in this context. In tasks like Named Entity Recognition (NER), where label dependencies are prevalent and meaningful, the BiLSTM model alone may struggle to effectively capture these dependencies, as its primary focus is on making predictions for individual tokens. In contrast, the CRF layer integrated into the BiLSTM-CRF model explicitly accounts for label transitions, ensuring that the predicted labels create coherent and valid sequences.

Q3

Here's an overview of how my extension operates: To begin, I initiate the server, which involves loading the model and initializing the word and tag dictionaries. Following this, I can navigate to a web page and utilize the mouse to select specific portions of text. Once the text is chosen, the web page triggers the reading process, which subsequently prompts a POST request directed at the server.

Upon receipt of the text, the server undertakes a series of steps. It processes the incoming text, transforming it into a list of indices. This index list is then forwarded to the loaded model for inference. The outcome of this inference is presented as a list of tag indices. The server proceeds to translate these tag indices into their corresponding actual tags. If a word is recognized as a named entity, the server inserts the associated tag directly after that particular word.

The concluding stages involve the server sending the revised text back to the frontend. Subsequently, a discreet window materializes on the web page, presenting the resulting output for perusal.

To enhance the utility of my extension, the initial step involves incorporating a character embedding layer at the outset of the architecture. This addition empowers the model to accurately identify and label unfamiliar words. Relying solely on word embedding would invariably lead the model to classify these words as non-named entities. Another enhancement I can implement involves deploying the model and server within a Docker container and subsequently constructing the Docker image. This encapsulation of the server within a container guarantees its seamless operation across various environments without any disruptions. Furthermore, in order to enhance the model's efficacy across a range of diverse web pages, I recognize the necessity of obtaining datasets sourced from online platforms. This involves procuring datasets that have been compiled from the internet to specifically cater to the varying characteristics of web content. As a potential solution, I plan to acquire datasets such as MultiNERD, WikiNEuRal, and wikigold, all of which originate from Wikipedia. Additionally, I intend to consider datasets like Ritter and BTC, both of which have been curated from Twitter. By incorporating such internet-derived datasets into my training regimen, I aim to equip the model with a broader contextual understanding of web-based content, thereby bolstering its performance on a multitude of online sources.

Q4

Q 4.1

I familiarized myself with the principles of Conditional Random Fields (CRF) and their application in named entity recognition (NER). I gained an understanding of how both attention mechanisms and CRF can contribute to enhancing the overall performance of NER models. My exploration extended to various alignment score functions within the attention mechanism framework.

Furthermore, I delved into the practical implementation of CRF and attention mechanisms within recurrent neural networks. This involved acquiring insights into their configuration and integration into the architecture. Recognizing the significance of diversified feature embeddings in NER, I expanded my knowledge on the importance of incorporating multiple types of embeddings to enrich the model's understanding and performance.

Q 4.2

The outcomes I have achieved in terms of entity extraction thus far fall short of my expectations. As previously discussed, my strategy moving forward entails the incorporation of additional feature embedding layers, encompassing character embedding, Part-of-Speech (POS) tagging, and chunking. Furthermore, I intend to implement an attention layer in line with the approach outlined in the paper.

To address the limitations encountered, my plan involves training the model on a more extensive dataset, sourced from a diverse array of web pages. This broader dataset should contribute to a more comprehensive understanding of web-centric content, ultimately enhancing the model's performance.

Expanding my horizons, I aim to delve into advanced techniques such as transformers, BERT, and large-scale language models. I am eager to explore their potential contributions to refining my model's capabilities. By experimenting with these techniques, I aspire to achieve a substantial improvement in the efficacy of my entity extraction model.

Q4.3

We can build an app that scans cookizng blogs, articles, and websites, extracting named entities like ingredients, cuisine types, and dietary preferences using NER. Users provide dietary requirements and flavor preferences. The app uses NER to extract ingredients from recipes and matches them to users' preferences. It generates personalized, healthy recipes that align with their dietary needs and taste preferences, offering an innovative solution for individuals seeking nutritious and tailored meal options.