POS tagging and NetworkX

Introduction

Part-of-speech (POS) tagging is a fundamental task in natural language processing that involves labeling each word in a sentence with its corresponding part of speech (e.g., noun, verb, adjective, etc.). This task is essential for many downstream applications, such as sentiment analysis, machine translation, and speech recognition. In this report, we present a Vanilla RNN model implemented using the Keras deep learning framework for POS tagging.

libraries

- Keras: a high-level neural network API written in Python that runs on top of TensorFlow. It provides a user-friendly interface for building and training deep learning models.
- NumPy: a Python library for scientific computing that provides support for large, multi-dimensional arrays and matrices.
- NLTK: a popular Python library for natural language processing that provides various tools and resources for tasks such as tokenization, stemming, and POS tagging.
- Spacy: is a popular open-source library for natural language processing in Python. It provides pre-trained models for POS tagging and dependency parsing for various languages,
- NetworkX: is a Python library for creating, manipulating, and analyzing graphs and networks. It can be used to represent the syntactic structure of a sentence, where nodes represent words, and edges represent their relationships.

Data description

downloads the "conll2000" dataset from NLTK. This is a dataset that contains a large number of sentences with their corresponding POS tags.

Each sentence in the list is itself a list of tuples, where each tuple contains a word from the sentence and its corresponding POS tag, using the universal tagset.. The training set contains 7909 sentences, and the test set contains 1643 sentences.

Steps

1. Data preparation: "conll2000" dataset is preprocessed using the NLTK library to extract the words and their corresponding tags. The words and tags are converted to integer indices.
2. Model architecture: The Vanilla RNN model is implemented using the Keras Sequential API. It consists of an embedding layer that maps each word to a dense vector representation, followed by a single-layer RNN that processes the sequence of word embeddings and produces a sequence of hidden states. Finally, a dense layer with a softmax activation function is used to predict the most likely tag for each word.
3. Model training: The model is trained using the Adam optimizer and the categorical cross-entropy loss function. The training is performed for a fixed number of epochs, and the best model is selected based on its performance on the validation set.
4. Model evaluation: The trained model is evaluated on the test set using the accuracy metric, which measures the proportion of correctly predicted tags. Additionally, the model is tested on a few sample sentences to verify its performance.

Task enhancement

- Incorporating character-level information, such as prefixes and suffixes, to better capture morphological features of words.

For the second section

Steps

1. Load the sample data and initialize a Spacy NLP model.
2. Iterate through each document and process it using the Spacy NLP model.
3. Extract the named entities from each document and store them as nodes in a NetworkX graph.
4. Connect the nodes in the graph based on their co-occurrence within a sentence.
5. Visualize the graph using NetworkX.

Conclusion

By presented a Vanilla RNN model implemented using the Keras deep learning framework for POS tagging. The model achieved a test set accuracy of 98.19% This means that out of all the words in the test set, the model correctly predicted their POS tags 98.19% of the time., which demonstrates its effectiveness in this task. Overall, the model provides a solid baseline for POS tagging, and it can be extended and adapted to more complex NLP tasks.

And by using Spacy I can gain the best results without the train model from scratch.