Assignment 3 – Part 1 – Understanding the Challenge

Submitters:

Ori Braverman 318917010

Elie nedjar 336140116

Introduction

In this part of the assignment, we need to distinguish between the two artificial languages:

1. Positive Examples: Sequence of the form: [1-9] +a+ [1-9] +b+ [1-9] +c+ [1-9] +d+ [1-9]
2. Negative Examples: Sequence of the form: [1-9] +a+ [1-9] +c+ [1-9] +b+ [1-9] +d+ [1-9]

Now we will determine whether the sequences can be correctly classified using different techniques.

Bag-Of-Words approach

The two languages **can't** be distinguished with the BoW approach.

The BoW represent the data as a set of words and ignore their order and therefore cannot differentiate between languages. This technique creates a vocabulary of known words and uses a vector to represent the frequency/presence of words in each text.

For example, BoW will represent the sequences 1a1b1c1d and 1a1c1b1d with the same vector.

Bigram/trigram-based approach

The two languages **can't** be distinguished with the Bigram/trigram-based approach.

The Bigram/Trigram model analyse the frequency of adjacent pairs (bigrams) or triples (trigrams) of characters in the dataset.

Bigram and Trigram store 2 or 3 consecutive characters and can't see any other dependencies beyond that fixed window.

The sequence in the languages can have variable length of digits ([1-9]) between each character.

So, that means that the critical pattern that distinguish the languages (b and then c in positive, c and then b in negative) can be separated by an arbitrary number of digits.

Therefore, this critical pattern can't be captured by a fixed window of 2/3 characters.

If the sequences have a fixed number of digits in the critical pattern:

- With 0 digits, both Bigram and Trigram models can distinguish the languages.
- With 1 digit, only the Trigram model can distinguish the languages.

Convolutional Neural Networks

The two languages **can't** be distinguished with CNN approach.

As mentioned in the Bigram/Trigram approach, the critical pattern can be separated by an arbitrary number of digits.

CNNs use filters of a fixed size to detect local patterns from the input. Fixed-size filters cannot capture dependencies that span beyond their size.

Even with a sliding window approach, where the filter moves across the sequence, the window size remains fixed.

If the critical pattern has a fixed length that is smaller or equal to the filter size then the CNN will be able to get the necessary pattern.