Challenge

Refael Shaked Greenfeld 305030868,

Danit Yshaayahu 312434269

1. No, the two languages can't be distinguished using BOW because the BOW approach doesn't express the order of the words, for example:

Look on the positive sentence $1a1b1c1d1$ it's vector will be something like that: $\{1: 5, 2: 0, \ldots, a: 1, b: 1, c: 1, d: 1\}$ but same vector will represent also the negative sentence *1a1c1b1d1*. Because there is same representation for two sentences in different languages we can't distinguish between the languages.

2. No, we can't distinguish between the languages. Look on the positive sentence *1a1b1c1d1* and the negative sentence *1a1c1b1d1* both have same bigrams: (1a: 1, a1: 1, 1b: 1, b1:1, 1c: 1, c1:1, 1d: 1, d1:1) so because there is the same representation for two sentences in different languages we can't distinguish between the languages. We now look at similar example for trigram, the positive sentence *11a11b11c11d11* and the negative sentence *11a11c11b11d11* have same representation: (11a: 1, 1a1:1, a11: 1, 11b: 1, 1b1:1, b11:1, 11c: 1, 1c1:1, c11:1, 11d: 1, 1d1:1, d11:1) so we can't distinguish them by trigram either.

3. similarly to the previous answers, Convolution network work with a fix sized windows, are limited by length of max window size, so if we suppose in the negative that there is such set of n windows that can capture such a language so they have to a maximum window length in size $K \in N$ and for an input in the form of $[1-9]^{k+1}a^{k+1}[1-9]^{k+1}b^{k+1}[1-9]^{k+1}c[1-9]^{k+1}d^{k+1}[1-9]^{k+1}$ and in the other hand an input in the form $[1-9]^{k+1}a^{k+1}[1-9]^{k+1}c^{k+1}\ [1-9]^{k+1}b^{k+1}[1-9]^{k+1}d^{k+1}[1-9]^{k+1}$ it wouldn't be able to identify between the two because his biggest filter is limited by K characters.