École national Supérieure d'Informatique, Algiers
2CSSID (2023-2024)
Natural Language Processing (NLP)

Lab02: Tweets' grammaticality judgment

Abdelkrime Aries

Grammaticality refers to how well a sentence conforms to the rules of a particular language. It seeks to judge structure and not meaning. In this lab, we want to test Algerian tweets statistically. In other word, we want to identify the grammatical status of a tweet directly with the probability of its occurrence. We chose to us N-Grams.

1 Program description

Here, different functions are described; either those implemented or not. You have to understand how they are implemented to respond on the different questions.

1.1 Ngram class

Given an already implemented Bi-grams class (N=2), implement a new one with N predefined. The class must:

- take in consideration the case where N is less than 1 or greater than 6;
- support both Lidstone and Interpolation smoothing. Let say for both, the parameters are predefined (in case of interpolation, no need to estimate them using another validation dataset);
- The parameters must be defined in the constructor;
- Interpolation parameters are from the lowest order to the highest;
- The smoothing type must be defined while predicting.

These are some recommendations concerning **score** function:

- Any division by zero will have a log probability of **-math.inf**
- log probability of 0 is **-math.inf**
- For unigrams, the probability of interpolation is that of Lidstone;
- For n-grams $n > 1$, using interpolation, we consider higher order as 0 when probability is 0 or there is division by zero. Example, 0.7 * -inf + 0.3 * 0.02 is 0.06 and not -inf (-inf will be handled as a 0)

For **predict** function, anything plus minus infinity is minus infinity

This must be implemented.

1.2 Grammaticality class

Not implemented.

2 Questions

Answer these questions at the start of your code, as comments.

1. We want to consider unknown words in general texts, propose a solution. We want to take in consideration different variants of Arabizi, propose a solution.
2. If we train a model on the inverse of texts. Will we get the same probability as the one in the right direction? Why? Will this affect grammaticality judgment? Why?
3. Can we use Viterbi to calculate the probability of a text? Why/How?

4. Describe how can we decide that a text is grammatical or not based on its probability.

3 Evaluation

- Duration: 1h
- Grade
  - **Exceptions** (1pt) = $N < 1$ (0.5pts) + $N > 6$ (0.5pts)
  - **fit grade** (6pts) = processing all data (2pts) + all N-grams (2pts) + all less than N– grams (2pts).
  - **score grade** (5pts) = correct and less complex estimation (2pts) + applying smoothing (2pt) + log probability (1pt).
  - **predict grade** (3pts) = correctly calculate the log probability (2pts) + correctly taking smoothing in consideration (1pt).
  - **questions grade** (4pts) = 1pt for each question.
  - **In time grade** (1pt): after the deadline, each late 2 minutes are -0.25. So, 8 minutes then you will get 0.

Ramadan kareem