Abstract : In this project, we develop a statistical language model using n-grams. We trained our model on a corpus of English sentences and …

**METHOD** : A n-gram model is a statistical model of language to compute probabilities of a word given some history in the text. More precisely, if w1, …, wN denote a succession of N words, the n-gram model computes the probability P(wN | w1, …, WN-1), thus making the assumption that WN depends only on these N-1 previous words and not on those before. This is known as the N-order Markov assumption.

To derive these probabilities, one must go through the whole text corpus used for training and count the number of occurrences of each of the existing chains of N words.

Different problems arise: if we want sophisticated models, it is tempting to go for high orders of n-grams. But these models then suffer from the lack of data (on an average data set, most 4-grams or 5-grams occur once at most). Such models tend to overfit the training data. Another problem is the presence of unknown words in the test set. These words, which did not appear in the training set, are unknown to the model and would be assigned a probability of zero if we did not explicitly fix that issue. This is not a problem when we work on a closed vocabulary, where we decide first of a list of words and limit the test to sentences containing those words. But this condition is too restrictive and we chose to work on an open vocabulary. The idea is to add a word unknown (UNK) in the vocabulary and to treat each unknown word and each first occurrence of known words as an additional occurrence of UNK. After this transformation of the vocabulary, we treat the pseudo-word UNK as any other word for the training of the n-gram.

To solve the issue of known words appearing in an unknown context, several smoothing technics can be used. We implemented a few of these and will compare their performance in the section “results”.

**Laplace smoothing**

The aim of this smoothing method is to make every n-gram possible, by adding one occurrence for every one of them. In consequence, n-grams never occurring are now occurring once, and n-grams occurring k times now occur k+1 times. We then normalize by a different factor to take into account the new number of occurences of all the n-grams.

**Back-off**

Using the trained n-grams models, we can now compute the probability of a complete sentence with the chain rule.

INSERT FORMULA HERE

This formula shows the necessity of introducing a new word denoted by the token SOS (Start of Sentence) to be able to compute the probability of the first words of the sentence. This specific token is added not only at the beginning of the test sentence for which we want the probability, but also in every sentence in the training set, as many times as required by the order of the n-gram (n-1 SOS for a n-gram).

A problem occurs when one of the n-grams included in the sentence does not appear in the training corpus. We would have a probability zero assigned to this n-gram and the probability assigned to the sequence would be zero. The idea of back-off is to reduce the context: instead of looking for the n-1 previous words, we can look for only the n-2 previous words and see if this is an existing context of words in the n-1 -gram model. We can even go further and reduce the order until we find a non-zero probability of reducing only once is not enough.

**Interpolation**

We can solve the issue mentioned in the previous section in a different way. Instead of backing-off to a smaller order, we can compute the probability of wN given W1, …, WN-1 as a weighted sum of the probabilities in every k-gram model for k<=N.

INSERT FORMULA HERE

The coefficients lambdas have to be determined such that this interpolation algorithm is optimized. One way to do that is to compute the lambdas using the expectation maximization algorithm to optimize the likelihood of some validation data given these parameters.

INSERT REFERENCE HERE

**Methodology for evaluation of the models**

**EXPERIMENTS**

**Data**

We used a dataset containing 93 424 sentences in English. It is available online at the following address:

INSERT LINK HERE

We divided it into 2 sets for training and testing. For methods where we need to tune parameters like the interpolation method, a third set (validation) is required. It is taken as a subset of the training set.

**References**