**Report**

Probabilistic Graphical Model

03/01/2015

N-most probable configurations

Master MVA

Authors:

Amine ER-RAQABI & Mohamed Nait N’Mbark

**Introduction**

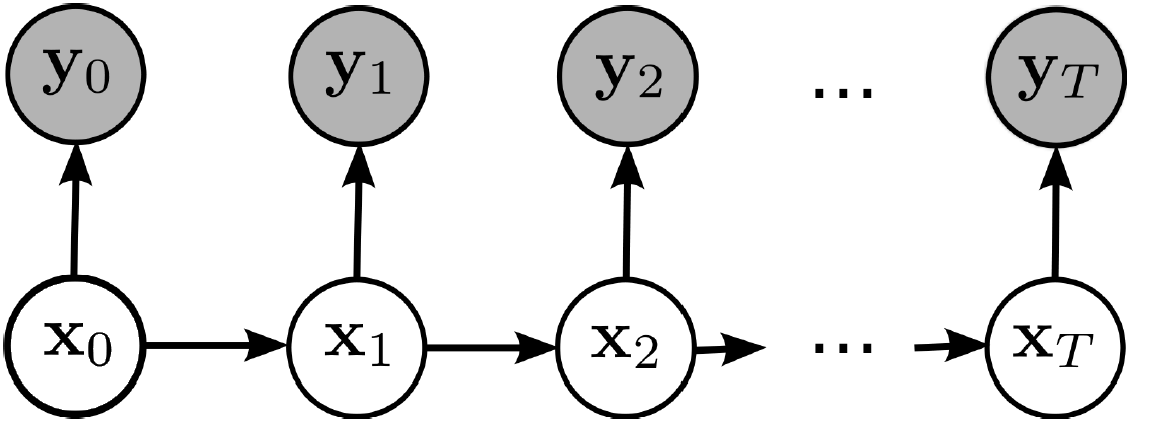


Figure 1: HMM graph

Considering an HMM , the N- Best List algorithm consists of the following :

N Best Combinations

Knowing:

Retreive

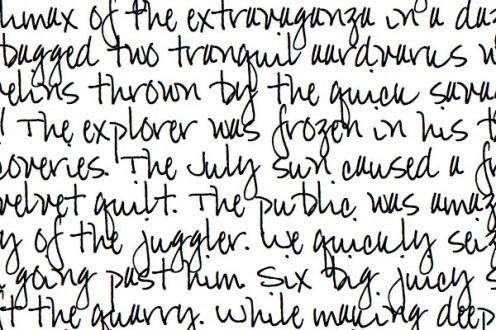
Inference Theory

Based on some observations, and using the inference theory through the message passing down then up a graph, and more specifically by handling the forward backward coefficients ( alpha and beta ) considering the Max-Product algorithm - that we will expose in details later in this report- we can find the best combination for the hidden variables ( Viterbi algorithm ) but also the N-best combinations using a sequential algorithm as the paper we studied, explains clearly

Application fields:

Such algorithm is widely used in issue like voice recognition and handwriting recognition as well.

Voice recognition Handwriting recognition



*Difficulty:*

Working on such an interesting project was not with any difficulty. Its common applications being quite complicated to implement, we tried to find an easy application to illustrate the power of this algorithm. It consists of retrieving the accented words from words with deliberately removed accents

**Application/Experiment**

Our application idea to illustrate in a real case this algorithm was inspired from the world of characters recognition, completion, correction…etc. The problem raised here is the following:

*Given a non accented word in French, how to retrieve its N- most likely (correct) spellings (with accents) in the French language?*

Example: “médecin” **The output**

1. Médecin
2. Medécin
3. Mêdecin
4. Medecin

**The input**

N-Best List Algo

Medecin

Figure 2: Example of the described application

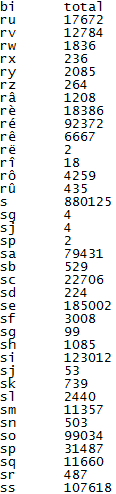
We can see that the “correct” word does not appear at the first place. Indeed all depends on the quality of the database used, but especially on our model that took only bigrams into consideration - as it will be mentioned latter in this report.

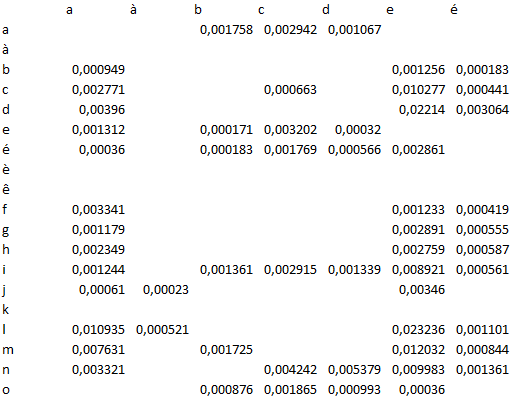
I- Prepare required data

We had to prepare specific data compulsory for such computations on any Hidden Markov Model: Emission Matrix, Transition Matrix, vector of initial probabilities

***Transition Matrix:***

Basically ,we started by looking for linguistic statistics that will help us gather bigrams of letters and their corresponding frequency in a set of different sources ( news-papers, books, novels … etc)





**A sample of the transition matrix**

**A sample of bigrams used**

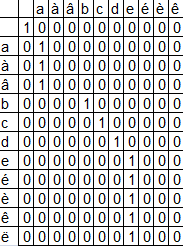
* The frequencies of the bigrams were transformed to a probability that represents – for our Hidden Markov Model – the transition probabilities
* The possible states of each hidden variables are all the possible letters in the French Language (accented or not ) :

Space + a à â b c d e é è ê ë f g h i î ï j k l m n o ô p q r s t u û v w x y z

***Emission Matrix:***

The Emission matrix sums up the likelihoods of the observations ( the probability of an observation knowing the hidden variable)

In our case, the observed variables containing only non-accented letters, the emission matrix would be a sparse matrix as follows:



* A non-accented letter would give a probability 1 of observing the same non accented letter ( i.e. the case of the letter « a »)
* A accented letter should give a probability 1 of observing its non-accented version ( i.e. the case of the letter « é » giving probability 1 of observing « e » )

**A sample of the emission matrix**

***Initial distribution vector***

This vector is formed of the probabilities of observing all letters ( accented or not) at the beginning of a word, which means the probabilities of the bigrams:

[ SPACE – X ] where X is any letter of the considered alphabet.

II-Test & Results

We finally can run the algorithm of N-best list, using the matrices and data gathered, and check the performance of our application’s results:

Here are some tests run on several words with the 3 best combinations found at each time:

* universite

|  |  |  |
| --- | --- | --- |
|  | Combinations | Probabilities |
| The best | universite | 4.1887e-14 |
| The second best | université | 8.1839e-15 |
| The third best | univérsite | 1.6587e-18 |

* antecedent

|  |  |  |
| --- | --- | --- |
|  | Combinations | Probabilities |
| The best | antécédent | 8.4538e-14 |
| The second best | antécedent | 7.8419e-14 |
| The third best | antecédent | 6.4538e-14 |

* fute

|  |  |  |
| --- | --- | --- |
|  | Combinations | Probabilities |
| The best | fute | 5.9800e-07 |
| The second best | futé | 1.1684e-07 |
| The third best | fûte | 2.2178e-08 |

**Conclusion**

This algorithms works well for several cases, but fails in some particular cases.

Thus a model based on trigrams would be a realistic improvement that worth being investigated but will necessitate a more complex graphical model.