

### I. Generate Unigram, Bigram and Trigram:

For each Article (or a given Volume, or all the Corpus of articles), generate the Unigrams, Bigrams and Trigrams with or without normalization (Porter, Lancaster and Snowball).

#### Part 4 - N-Gram Language Model

N-Gram:

Model

Unigram

Generate

Test sentence

Compute raw probability

Example of Unigrams for a given Volume:

#### Part 4 - N-Gram Language Model

N-Gram:

Model

Unigram

Generate

N-Gram

Frequency

Probability  $P(W_n|W_{\dots})$

the

4479

0.0534

</s>

3594

0.0429

</s>

3594

0.0429

and

2763

0.033

of

2301

0.0275

83823 Unigrams with 4229 unique Unigrams

</s> is either a ".", "?" or "!" (see the example provided with TP n°1)

Example of Trigrams for a given Volume:

## Part 4 - N-Gram Language Model

N-Gram:

Model

Trigram

Generate

N-Gram	:	Frequency	Probability P(Wn W...)
</s> <s> these		72	0.02
algorithm </s> <s>		66	1
problem </s> <s>		63	1
particle swarm optimization		63	0.9545
exploration and exploitation		63	0.7241

83821 Trigrams with 23592 unique Trigrams

The probabilities are estimated based on Markov Hypothesis and using Chain Rule, as mentioned below :

$$P(\mathbf{w}_{1:n}) = \prod_{k=1}^n P(w_k | w_{k-1})$$

For Unigram :

$$P(w_n) = \frac{C(w_n)}{N}$$

For Bigram :

$$P(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n)}{\sum_w C(w_{n-1} w)} = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

For N-Gram :

$$P(w_n | w_{n-N+1 : n-1}) = \frac{C(w_{n-N+1 : n-1} w_n)}{C(w_{n-N+1 : n-1})}$$

Where:

$w_n$  : is a token.

$C(w_n)$ : is the number of occurrences of the unigram  $w_n$ .

$N$ : indicates the number of tokens.

$C(w_{n-1} w_n)$  : is number of occurrences of the bigram  $w_{n-1} w_n$ .

$\sum_w C(w_{n-1} w)$  : is number of occurrences of bigrams starting with  $w_{n-1}$ .