# Abstract

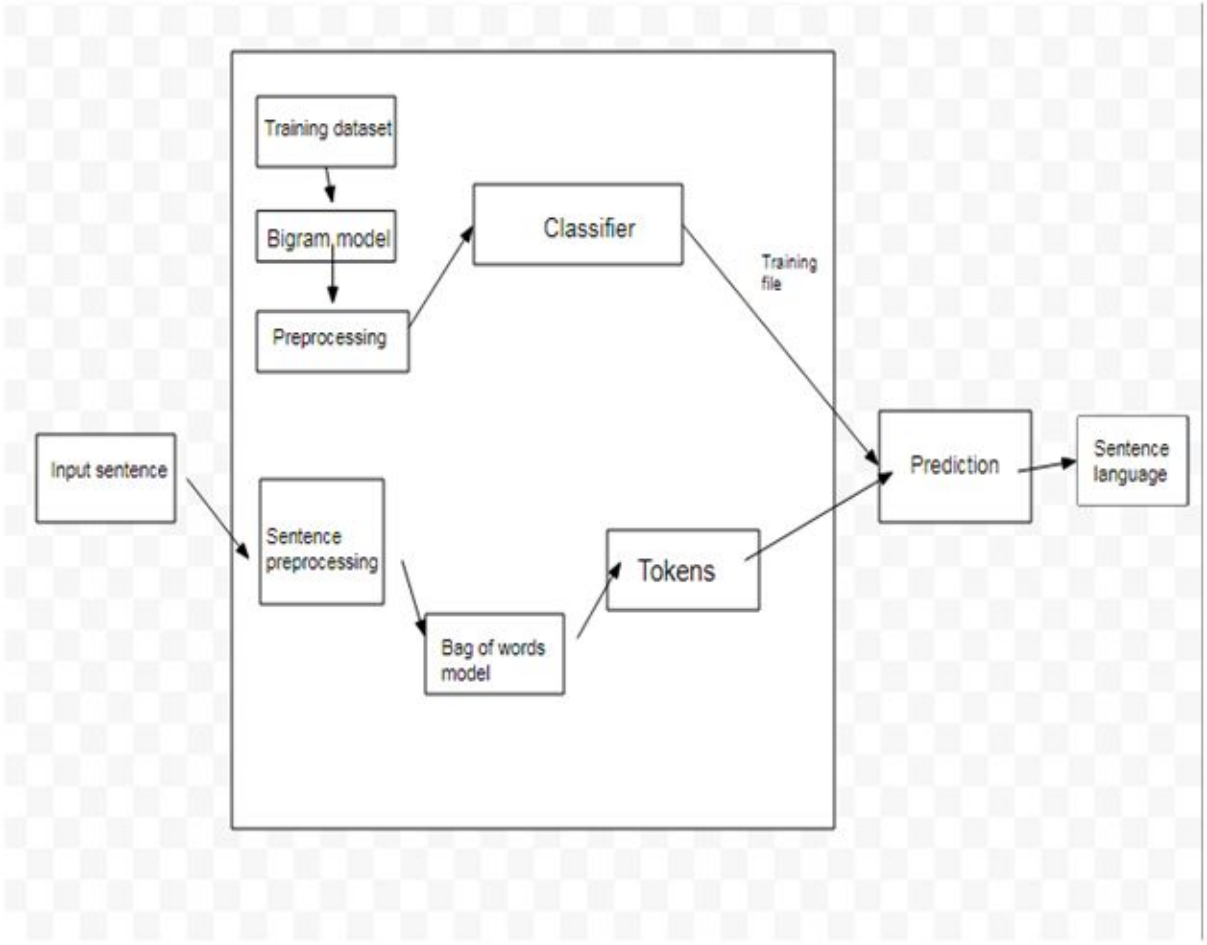
Language identification means to know in which subject a specific text related to , is it English , Arabic , French , .. etc language identification is used in many techniques One technique is to compare the compressibility of the text to the compressibility of texts in a set of known languages. This approach is known as mutual information based distance measure. The same technique can also be used to empirically construct family trees of languages which closely correspond to the trees constructed using historical methods. another technique is voice recognition which is converted to text , the text language should be verified to answer in the right language . and many other techniques . in this documentation we will represent a simple software that works to identify the language of the entered sentence using machine learning technique , we learn it to detect the most popular 6 languages [English , Arabic , French , German , Spanish and Franco Arab ] , we used popular algorithms in Supervised machine learning which are Naive Bayes and Maximum Entropy algorithms that we will describe below .

# Data set

we had 6 different datasets " dataset for each language" , but we had have to collect the Franco dataset ourselves using twitter APIs " Streaming API" for specific users on twitter. each language datasets contains more than 13,000 sentences "Franco other languages are bigger ". all datasets contains sentences of the language which have the most popular words in each language .

# Methodology

this figure explains the block diagram of language identification technique (software) , each block will be described in more details .




**Input Sentence**

it's the sentence that will be entered to the software (classifiers)

in our case it's a text file that contains many sentences which written in different languages.

---

**Training Data set**

these are the data sets that the classifiers will learn from in our case we used 6 different datasets , one dataset for each language each dataset contains more than 15,000 sentence (more for popular languages ).

---

**Bi-gram Model**

A bigram or diagram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words. A bi-gram is an n-gram for n=2. The frequency distribution of

every bigram in a string is commonly used for simple statistical analysis of text in many applications, including in computational linguistics, cryptography, speech recognition, and so on.

in our case bi-gram will increase the accuracy of identifying the language , as some languages are similar in a set of words will may reduce the accuracy of predicting the language of a sentence . Although it reduces the performance as it works on each adjacent words , but in our case the sentence length is good enough to work a bi-gram model on it (maximum 30 words in a sentence)

---

**Preprocessing**

Preprocessing step is an important step , it's divided to many steps like :

***tokenization :*** which breaking the sentence down to set of tokens

***Stemming :*** it reduces the overhead on the classifiers as it normalize the tokens (means a standard format)

**note that we didn't remove stop words as it effects on the language identification**

---

**Bag of Words :**

Also known as the vector space model. In this model, a text (such as a sentence or a document) is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity.

in our case we use it to create a vector for each token that will be entered to the classifier to be predicted .

---

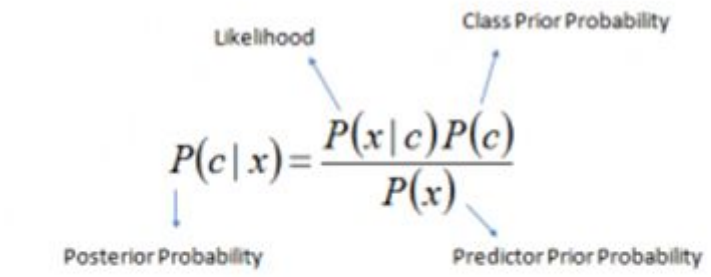
**Classifiers :** as we said before we built our techniques based on two different classifiers : ***Naive Bayes and Maximum Entropy*** we are going to describe them in the next section .

## Experiment 1

**Naive Bayes :**

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:




$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

in our case we used the library of **nltk.classifier** to import the naive bayes classifier , using the 6 different datasets we built and train our classifier .

## Experiment 2

**Maximum Entropy :**

The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naive Bayes classifier that we discussed in the previous article, the Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit our training data, selects the one which has the largest entropy. The Max Entropy classifier can be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more.

we regularly use it when we don't know anything about the prior distributions, and when it is unsafe to make any such assumptions.

Moreover Maximum Entropy classifier is used when we can't assume the conditional independence of the features. This is particularly true in Text Classification and language identification problems where our features are usually words which obviously are not independent.

in our case we used the library of **nltk.classifier** to import the MaxEnt classifier , using the 6 different datasets we built and train our classifier .

---

**Prediction :** prediction is the final step which uses both the training file that the classifier got out after learn itself how to identify each language with its tokens used in this language and the set of tokens that we want to predict (identify ) .

# Results

we entered a set of sentence with our prediction for each sentence and we get 95-100 % as an accuracy for naive Bayes algorithm for the 6 languages

using maximum entropy classifier , we trained it with only one iteration and we get an accuracy between 45-60% , but after 5 iteration the training accuracy became constant and we got 100%

# Discussion

**About Classifiers :** as we present in the result section we entered a set of sentence with our prediction for each sentence and we get 95-100 % as an accuracy for naive Bayes algorithm for the 6 languages

using maximum entropy classifier

`we trained it with only one iteration and we get an accuracy between 45-60%`

, but after 5 iteration the training accuracy became constant and we got 100%

for maximum entropy classifier we got better results as we increased the number of training (learning ) iteration , which means increases the tokens fitting for each language ,which increase the ability of detect the similar sentences with small difference written in different languages .

**About Preprocessing :** Before using Bi-gram model the accuracy reduced to 35-60 % as we said before there are similar words between the different languages which effects in the probability of predicting the sentence

but after using bigram model the accuracy increased to 95-100% , but we Sacrifice the performance (time) for the accuracy

# Conclusion

we have develop a software that it's used to identify the language of a sentence using machine learning and NLP approaches , we worked on 6 languages [Arabic , English ,French , Spanish, ,German , and Franco Arab] . we used two algorithms and compared the results ,Naive Bayes and Maximum Entropy classifiers ,we got results between 95-100%