وزارة التعليم العالي و البحث العلمي

جامعة بابل - كلية العلوم للبنات

قسم علوم حاسبات

Language detection using machine learning

بحث تخرج تقدمت به

سبأ حامد علي

الى مجلس كلية العلوم للبنات في جامعة بابل وهو جزء من متطلبات نيل درجة البكالوريوس في علوم الحاسوب

بأشراف

الاستاذة

جنان علي عبد

2023 م

1444 هـ

بسم الله الرحمن الرحيم

صدق الله العلي العظيم

الاهداء

الحمد لله الذي هدانا لهذا وما كنا لنهتدي لولا ان هدانا الله

وصلت رحلتي الجامعية إلى نهايتها بعد تعب ومشقَّة..

وها أنا ذا أختم بحث تخرُّجي بكل همَّة ونشاط،

وأمتنُّ لكل من كان له فضل في مسيرتي،

وساعدني ولو باليسير،

الأبوين، والأهل، والأصدقاء، والأساتذة المُبجَّلين..

أُهديكم بحث تخرُّجي.....

الشكر والتقدير

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء والمرسلين سيدنا محمَّد وعلى اله وصحبه ومن تبعهم بإحسان إلى يوم الدين، وبعد ..

فإني أشكر الله تعالى على فضله حيث أتاح لي إنجاز هذا العمل بفضله، فله الحمد أولاً وآخرًا.

ثم أشكر أولئك الأخيار الذين مدوا لي يدَ المساعدة، خلال هذه الفترة، وفي مقدمتهم أستاذتي المشرفة الأستاذة / جنان علي عبد التي لم تدَّخر جهدًا في مساعدتي، ،وكانت تحثني على البحث، وترغِّبني فيه، وتقوّي عزيمتي عليه فلها من الله الأجر ومني كل تقدير حفظها الله ومتّعها بالصحة والعافية ونفع بعلومها .

الخلاصة

Abstract

Language detection is one of the important tasks in natural Language processing (nlp) . It is used in many applications like translation , chats and others . Many nlp and machine learning techniques were used for Language detection problem.

In the proposed model , we used Bag of words method to convert text to number features . We used naive _ bayes method for text classification . The system was on trained online dataset including 17 Language .

Results showed the effectiveness of the proposed system with an accuracy of 97% .

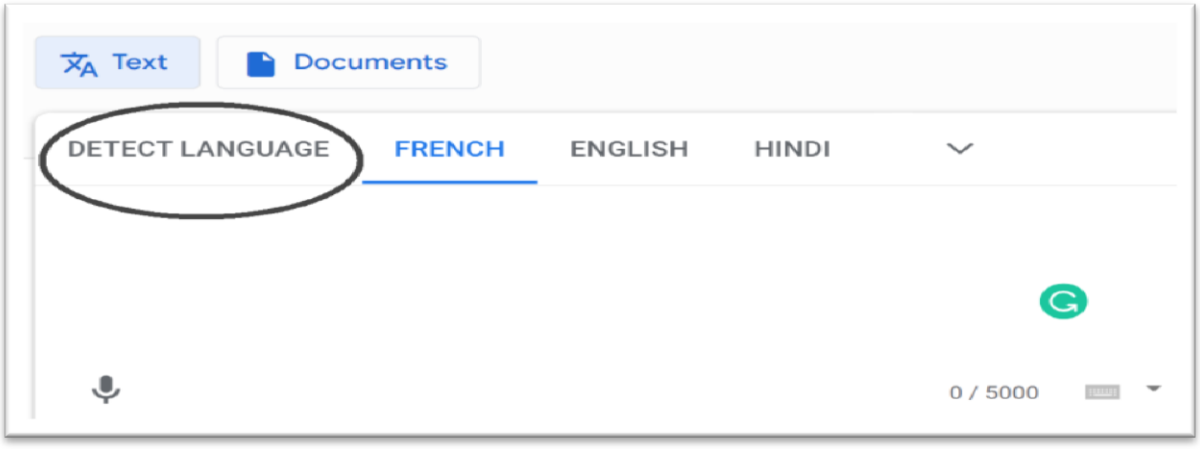
{ فهرست العناوين }

Chapter one

1. Chapter one

1.1 The introduction:-

Language detection is a natural language processing task where we need to identify the language of a text or document. Using machine learning for language identification was a difficult task a few years ago because there was not a lot of data on languages, but with the availability of data with ease, several powerful machine learning models are already available for language identification .As a human, you can easily detect the languages you know. For example, I can easily identify Arabic and English, but being an Arab , it is also not possible for me to identify all Arabic languages. This is where the language identification task can be used. Google Translate is one of the most popular language translators in the world which is used by so many people around the world. It also includes a machine learning model to detect languages that we can use if we don't know which language you want to translate [1] .



(fig (1) : An example of Language Detection model)

The most important part of training a language detection model is data. The more data you have about every language, the more accurate your model will perform in real-time.

1.2 Why we need language detection?

The initial stage in any pipeline for text analysis or natural language processing is language identification. All ensuing language-specific models will yield wrong results if the language of a document is incorrectly determined. Similar to what happens when an English language analyzer is used on a French document, errors at this step of the analysis might accumulate and provide inaccurate conclusions. Each document's language and any elements written in another language need to be identified. The language used in papers varies widely depending on the nation and culture.

1. Monolingual catboats: When a user starts speaking in a particular language, a bot must be able to recognize it even if it hasn't been properly educated to carry on a discussion in that language.

2. Spam filtering: Spam filtering systems that support many languages must identify the language that emails, online comments, and other input are written in before utilizing real spam filtering algorithms. Internet platforms cannot efficiently remove content from certain countries, regions, or locations suspected to be creating spam without this identification.

3. Recognize the language used in emails and chats: Language detection identifies the language of a text as well as the words and sentences where the language diverges. Since business messages .

4. (chats, emails, and so on) may be written in a variety of languages, it is frequently utilized.

5. Linguistic blending: Some people are used to having conversations that are bilingual. Hinglish, an amalgam of Hindi and English terminology used in India, would be a good illustration of this. In these situations, a language detection model will examine the number of words in a sentence written in one or more languages, with the language with the most words serving as the primary language for the interaction but the secondary language also being recognized and receiving a high confidence score in our ranking [2] .

1.3 Project outline

This project include also :

Chapter 2: Explain natural language processing fundamentals , method , tasks and steps

Chapter 3: Includes the practical side of the project , how it was been designed and implemented .

Chapter 4:Conclusion and future work .

Chapter two

2. Chapter two

2.1. Introduction to natural language processing nlp

Natural language processing (NLP) is the field of computer science and linguistics concerned with the interactions between computers and natural languages [3][4][5] . Which started as a branch of artificial intelligence which in turn branched out from informatics ,There is debate about the convergence and divergence of natural language processing from the field of computational linguistics. The Association for Computational Linguistics has defined computational linguistics as focusing on the theoretical aspects of natural language processing. Modern algorithms for natural language processing are based on machine learning , especially statistical machine learning[6]. Recent research in statistical machine learning algorithms requires an understanding of a number of disparate fields, including linguistics, computer science, and statistics [7].

2.1.1. Text preprocessing

Text processing is a branch of Natural Language Processing (NLP) that deals with the processing of written language itself. Also referred to as text mining [8] , this branch of natural language processing takes raw text and converts it into a form that a computer can understand and extract information from as logical structures or structured data.

Or it can be said: the process of extracting previously unknown information by a computer, by automatically extracting information from several written sources . Text processing techniques (text mining) are similar in the final goal to extracting information, but extracting

information is not limited to extracting it from the written text. Information can be extracted from images, sound, structured databases, etc.

Under the name of text processing, there are many algorithms that vary in their mechanism of action between statistical algorithms, deep learning, and linguistic algorithms [9] .

2.1.2 Text representation

In natural language processing, this conversion of raw text into a suitable digital form is called Text Representation. We'll look at different ways to represent text, or to represent text as a digital vector. Regarding the bigger picture of any natural language processing problem.

Attribute representation is a common step in any ML project, whether the data is text, images, videos, or speech. However, the attribute representation of text is often more complex compared to other data formats.

Bag of words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. The bag-of-words model has also been used for computer vision. The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier. [10]

An early reference to "bag of words" in a linguistic context can be found in Zellig Harris's 1954 article on Distributional Structure. [11]

The Bag-of-words model is one example of a Vector space model . [12]

The attribute classification methods are classified in four ways:

- Basic vectorization approaches
- Distributed representations
- Universal language representation
- Distributed representations

2.2 Natural language processing tasks

Natural language processing (NLP) techniques, or natural language processing (NLP) tasks, break down human text or speech into smaller parts that computer programs can easily understand. The following are common word processing and analysis capabilities in Natural Language Processing (NLP) [13].

- Mark the parts of speech **:** This is a process in which natural language processing (NLP) software tags individual words in a sentence according to contextual uses, such as nouns, verbs, adjectives, or adverbs. It helps the computer understand how words form meaningful relationships with one another.

- Clarify the meaning of the word : Some words may have different meanings when used in different scenarios. For example, the word bat means different things in these sentences:

    - A bat is a nocturnal creature: A bat is a nocturnal creature.
    - Baseball players use a bat to hit the ball: While here you mean baseball players use a bat to hit the ball.

    By stating the meaning of the word, NLP software determines the intended meaning of the word, either by training its language model or referring to dictionary definitions .

- Speech recognition : Speech recognition converts audio data into text. The process involves breaking down words into smaller parts and overcoming challenges such as accents, diphthongs, intonation, and incorrect use of grammar in everyday conversation.

- Machine translation : Machine translation software uses natural language processing to convert text or speech from one language to another while maintaining contextual accuracy.

- Recognize named entities **:** This process assigns unique names to people, places, events, companies, and so on. Natural language processing (NLP) software uses named entity recognition to determine the relationship between different entities in a sentence.

- Sentiment analysis : Sentiment analysis is an AI-based approach to interpreting sentiments conveyed by textual data. NLP analyzes text for words or phrases that show dissatisfaction, happiness, doubt, regret, and other hidden emotions.
- Language detection + translation

2.3 Machine learning and deep learning in nlp

- Machine learning (ML) : For natural language processing[14] (NLP) and text analytics [15] involves using machine learning algorithms and "narrow" artificial intelligence (AI) to understand the meaning of text documents. These documents can be just about anything that contains text: social media comments, online reviews, survey responses, even financial, medical, legal and regulatory documents. In essence, the role of machine learning and AI in natural language processing and text analytics is to

improve, accelerate and automate the underlying text analytics functions and NLP features [16] that turn this unstructured text [17] into useable data and insights.

- Deep Learning : Is a branch of Machine Learning that leverages artificial neural networks (ANNs) [18] to simulate the human brain's functioning. An artificial neural network is made of an interconnected web of thousands or millions of neurons stacked in multiple layers, hence the name Deep Learning [19] .

- **Naïve Bays Algorithm for Data Classification**

Bayesian classifiers are statistical classifiers. They can predict class

membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bays' theorem . Studies considered that naïve Bayesian classifier is comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Naïve Bays method is important for several reasons. It is very easy to construct, not needing any complicated iterative parameter estimation schemes. This means it may be readily applied to huge data sets. It is easy to interpret, so users unskilled in classifier technology can understand why it is making the classification it makes. And finally, it often does surprisingly well.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naïve". Bays’ Theorem: Bays’ theorem is named after Thomas Bays, a nonconformist English clergyman who did early work in probability and decision theory during the 18th century.

Let X be a data tuple. In Bayesian terms, X is considered “evidence.” As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, we want to determine $P(H|X)$, the probability that the hypothesis H holds given the “evidence” or observed data tuple X. In other words, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X. $P(H|X)$ is the posterior probability of H given X.

$P(H)$is the prior probability of H.

$P(X|H)$is the posterior probability of X given H.

$P(X)$is the prior probability of X.

$P(H)$, $P(X|H)$, and $P(X)$may be estimated from the given data. Bayes’ theorem is useful in that it provides a way of calculating the posterior probability, $P(H|X)$, from $P(H)$, $P(X|H)$, and $P(X)$.

Bayes' theorem is

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \ldots . (2.1)$$

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X=(x_1, x_2, … , x_n), depicting n measurements made on the tuple from n attributes, respectively, A_1, A_2, … , A_n.

2. Suppose that there are *m* classes, C_1, C_2, … , C_m. Given a tuple, ***X***, the classifier will predict that ***X*** belongs to the class having the highest posterior probability, conditioned on ***X***. That is, the naïve Bayesian classifier predicts that tuple ***X*** belongs to the class C_i if and only if

$$P(Ci|X) > P(Cj|X) for\ 1 \leq j \leq m, j \neq i \ldots \ldots (2.2)$$

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(Ci|X) = \frac{P(X|Ci)P(Ci)}{P(X)} \ldots . . \ldots (2.3)$$

3. As P(X) is constant for all classes, only $P(X|Ci)P(Ci)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1) = P(C2) = …= P(Cm), and

we would therefore maximize $P(X|Ci)$. Otherwise, we maximize $P(X|Ci)P(Ci)$. Note that the class prior

probabilities may be estimated by $P(Ci)=|Ci,D||D|$⁄,where $|Ci,D|$ is the number of training tuples of class Ci in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|Ci)$. In order to reduce computation in evaluating $P(X|Ci)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(X|Ci) = \Pi P(xk|Ci) \ldots . (2.4)$$

$$= P(x1|Ci) \times P(x2|Ci) \times \ldots . \times P(xn|Ci)$$

We can easily estimate the probabilities $P(x1|Ci)$, $P(x2|Ci)$,…, $P(xn|Ci)$ from the training tuples. Recall that here xk refers to the value of attribute Ak for tuple X.

5. In order to predict the class label of X, $P(X|Ci)P(Ci)$. is evaluated for each class Ci.

The classifier predicts that the class label of tuple X is the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j). for\ 1 \le j \le m, j \ne i \ldots . (2.5)$$

In other words, the predicted class label is the class C_i for which $P(X|\ C_i)P(C_i)$ is the maximum.

Figure (2.2) depicts the naïve bayes algorithm steps.

Algorithm: Naïve Bays

Input:

D: a data set containing n objects.

Output:

C: a selected class

Method:

For all values of the class attribute C_i

Calculate the prior probability $P(Ci)$

For all attribute values of tuple X (xk):

Calculate the posterior probability of xk conditioned on each class C_i=$P(xk|Ci)$

For all values of the class attribute C_i):

Calculate the posterior probability of X given Ci $P(X|\ C_i)$ =$\Pi P(xk|\ C_i)$ nk=1

Calculate $P(X|C_i)$ $P(C_i)$

For all values of the class attribute Ci:

return the largest value of $P(X|C_i)P(C_i)$

Figure (2.1) Naïve Bayes Algorithm

Example :

Predicting a class label using naïve Bayesian classification.

We wish to predict the class label of a tuple using naïve Bayesian classification, given the training data in table(2.4). The data tuples are described by the attributes age, income, student, and credit rating. The class label attribute, buys computer, has two distinct values (namely, {yes, no}).

Table (2.4) Electronic store Database

Age	Income	student	credit-rating	buy-computer
Youth	High	No	Fair	No
Youth	High	No	excellent	No
Middle	High	No	Fair	Yes
Senior	Medium	No	Fair	Yes
Senior	Low	Yes	Fair	Yes
Senior	Low	Yes	excellent	No
Middle	Low	Yes	excellent	Yes
Youth	Medium	No	Fair	No
Youth	Low	Yes	Fair	Yes
Senior	Medium	Yes	Fair	Yes
Youth	Medium	Yes	excellent	Yes
Middle	Medium	No	excellent	Yes
Middle	High	Yes	Fair	Yes
Senior	Medium	No	excellent	No

Let C1 correspond to the class buys computer = yes and C2 correspond to buys computer = no.

The tuple we wish to classify is

X = (age = youth, income = medium, student = yes, credit rating = fair)

We need to maximize $P(X|C_i)P(C_i)$, for i = 1, 2. $P(C_i)$, the prior probability of each class, can be computed based on the training tuples:

P(buys computer = yes) = 9/14 = 0.643

P(buys computer = no) = 5/14 = 0.357

To compute $P(X|C_i)$, for i = 1, 2, we compute the following conditional probabilities:

P(age = youth / buys computer = yes) = 2/9 = 0.222

P(age = youth / buys computer = no) = 3/5 = 0.600

P(income = medium / buys computer = yes) = 4/9 = 0.444

P(income = medium / buys computer = no) = 2/5 = 0.400

P(student = yes / buys computer = yes) = 6/9 = 0.667

P(student = yes / buys computer = no) = 1/5 = 0.200

P(credit rating = fair /buys computer = yes) = 6/9 = 0.667

P(credit rating = fair / buys computer = no) = 2/5 = 0.400

Using the above probabilities, we obtain

P(X| buys computer = yes) = P(age = youth | buys computer = yes) * P(income = medium |buys computer = yes) * P(student = yes | buys computer = yes) * P(credit rating = fair |buys computer = yes)

= 0.222*0.444*0.667*0.667 = 0.044.

Similarly ,P(X| buys computer = no) = 0.600*0.400*0.200*0.400 = 0.019.

To find the class, C_i, that maximizes $P(X|C_i)P(C_i)$, we compute

P(X| buys computer = yes) P (buys computer = yes) = 0.044*0.643 = 0.028

P(X| buys computer = no) P (buys computer = no) = 0.019*0.357 = 0.007

F Therefore, the naïve Bayesian classifier predicts buys computer = yes for tuple X .for hypothesis and it is robust to noise in input data.

Chapter three

3. Chapter three

3.1 Introduction

In these days, and with the spread of social media applications, the text over where on the net. Natural language processing nlp provide a group of tasks that can deal with these texts .One example is the language detection take which is necessary in many online application . For example it is used before translation.

3.2 The proposed model

The first step in the proposed is to read the dataset ,and apply preprocessing to delete any extra symbols, space to convert text to numbers using Bag_of_words method.

The next step is to split date to train and test.

The fourth step is to train the model using naive bayes classification

The last step is to evaluate the model using accuracy metric.

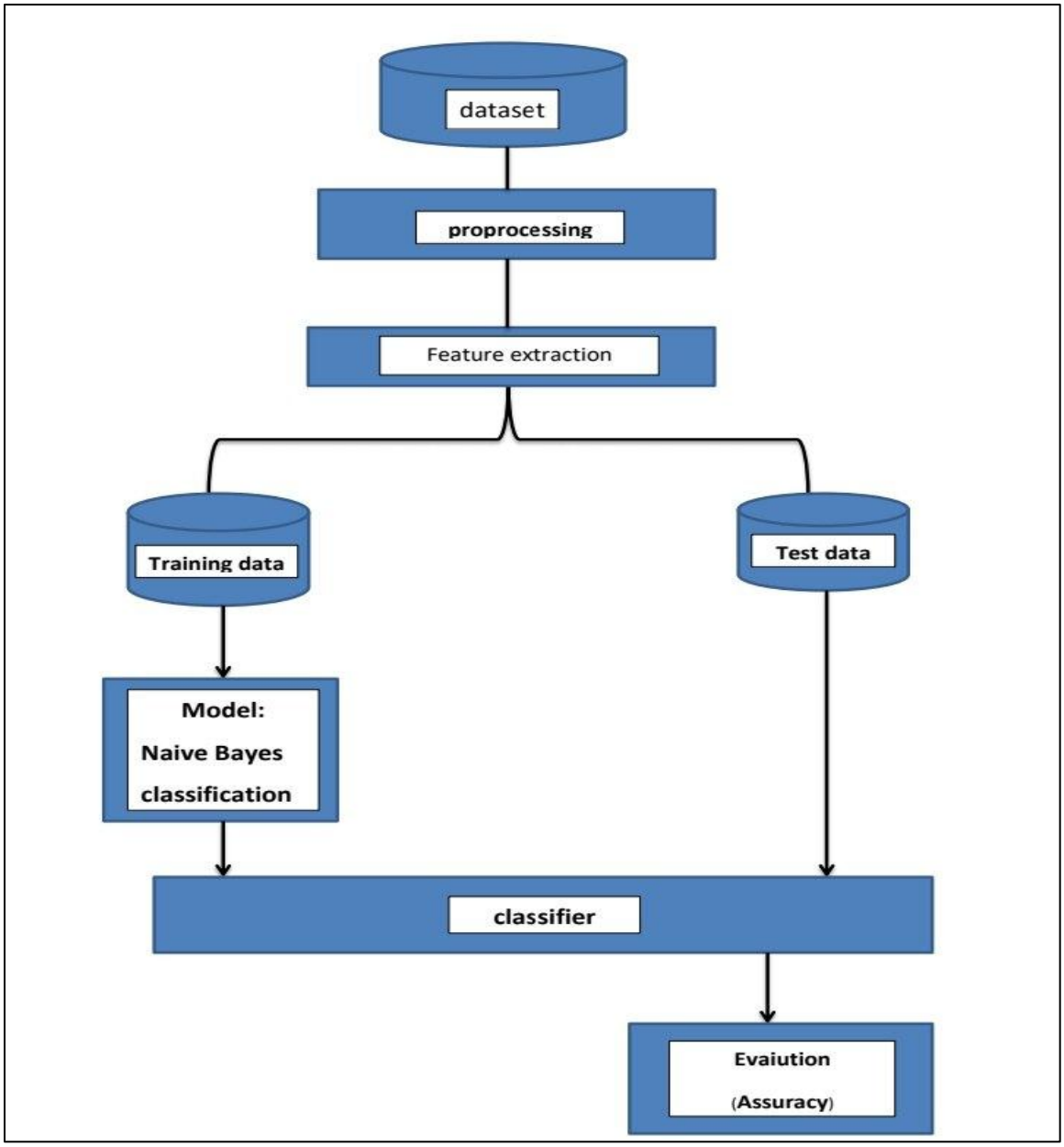


fig 3.1 The proposed model

3.3 About the Dataset

It's a small language detection dataset. This dataset consists of text details for 17 different languages, ie, you will be able to create an NLP model for predicting 17 different language and number of rows 10267 .

	Text	Language
0	Nature, in the broadest sense, is the natural...	English
1	"Nature" can refer to the phenomena of the phy...	English
2	The study of nature is a large, if not the onl...	English
3	Although humans are part of nature, human acti...	English
4	[1] The word nature is borrowed from the Old F...	English
...	...	...
10332	ನಿಮ್ಮ ತಪ್ಪು ಏನು ಬಂದಿದೆಯೆಂದರೆ ಆ ದಿನದಿಂದ ನಿಮಗೆ ಒ...	Kannada
10333	ನಾರ್ಸಿಸಾ ತಾನು ಮೊದಲಿಗೆ ಹೆಣಗಾಡುತ್ತಿದ್ದ ಮಾರ್ಗಗಳನ್...	Kannada
10334	ಹೇಗೆ ' ನಾರ್ಸಿಸಿಸಮ್ ಈಗ ಮರಿಯನ್ ಅವರಿಗೆ ಸಂಭವಿಸಿದ ಎ...	Kannada
10335	ಅವಳು ಈಗ ಹೆಚ್ಚು ಚಿನ್ನದ ಬೈಡ್ ಬಯಸುವುದಿಲ್ಲ ಎಂದು ...	Kannada
10336	ಟೆರ್ರಿ ನೀವು ನಿಜವಾಗಿಯೂ ಆ ದೇವದೂತನಂತೆ ಸ್ವಲ್ಪ ಕಾಣು...	Kannada

10337 rows × 2 columns

fig 3.2 the language detection dataset

3.4 Model Implementation

To implement this model, we used packages Like:

1. Pandas: pandas is a python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive.
2. Sklearn : include the train _ test following

- Sklearn.preprocessing : The Sklearn. preprocessing package provides several common utility functions and transformer classes to change

raw feature vectors into a representation that is more suitable for the downstream estimators.

- Sklearn.midel select: Sklearn's model selection module provides various functions to cross-validate our model, tune the estimator's hyper parameters, or produce validation and learning curves.
- Sklearn.naive bayes: Can perform online updates to model parameters via partial fit. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and Leveque
- Sklearn.metrics for evaluation purposes: In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.

3.5 Results

After training the model and testing it using the test date,we calculated the accuracy result was 79% .

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
accuracy = accuracy_score(y_test, y_prediction)
confusion_m = confusion_matrix(y_test, y_prediction)
print("The accuracy is :",accuracy)
```

```
The accuracy is : 0.973404255319149
```

Fig 3.3 The accuracy result

```
In [19]: lang_predict("This is my opinion")
         The langauge is in English

In [20]: lang_predict("كيف حالك ماذا تفعل")
         The langauge is in Arabic

In [21]: lang_predict("comment tu t'appellle")
         The langauge is in French
```

Fig 3.4 Language detection examples using the proposed model

Chapter four

4. Chapter four

4.1 Conclusions

1. Naïve biyes algorthm is good for text classification and give high accuracy .
2. Bag of words methods is suitable –for representing text as numbers.
3. Machine learning methods need successful text representation methods when dealing with text .

4.2 Future work

- Try different text representation method for better results .
- Try deep karning method for classificatin .
- Adding translation task after language detection task .

References

1. https://thecleverprogrammer.com/2021/10/30/language-detection-with-machine-learning/
2. https://heartbeat.comet.ml/using-machine-learning-for-language-detection-517fa6e68f22
3. Hutchins, J. (2005). "The history of machine translation in a nutshell" (PDF).[self-published source]
4. ^ "ALPAC: the (in)famous report", John Hutchins, MT News International, no. 14, June 1996, pp. 9-12
5. ^ Koskenniemi, Kimmo (1983), Two-level morphology: A general computational model of word-form recognition and production (PDF), Department of General Linguistics, University of Helsinki
6. https://ar.wikipedia.org/wiki/%D8%AA%D8%B9%D9%84%D9%85_%D8%A7%D9%84%D8%A2%D9%84%D8%A9
7. https://ar.wikipedia.org/wiki/%D8%A5%D8%AD%D8%B5%D8%A7%D8%A1
8. https://aiinarabic.com/glossary/text-mining/
9. https://aiinarabic.com/text-analysis-with-python-basic-processes/
10. Sivic, Josef (April 2009). "Efficient visual search of videos cast as text retrieval" (PDF). IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 4. opposition. pp. 591–605.
11. McTear et al 2016, p. 167
12. https://en.m.wikipedia.org/wiki/Vector_space_model

13. https://ai.malawad.com/%D8%AA%D9%85%D8%AB%D9%8A%D9%84-%D8%A7%D9%84%D9%86%D8%B5-%D8%A3%D8%B3%D8%A7%D9%84%D9%8A%D8%A8-%D8%A7%D9%84%D9%85%D8%AA%D8%AC%D9%87%D8%A7%D8%AA-%D8%A7%D9%84%D8%A3%D8%B3%D8%A7%D8%B3%D9%8A%D8%A9
14. https://www.lexalytics.com/resources/ml-nlp-whitepaper/
15. https://www.lexalytics.com/blog/text-analytics-functions-explained/
16. https://www.lexalytics.com/technology/features/
17. https://en.wikipedia.org/wiki/Unstructured_data
18. https://www.upgrad.com/blog/neural-network-architecture-components-algorithms/
19. https://www.upgrad.com/blog/deep-learning-vs-nlp/
20. *Sivic, Josef (April 2009). "Efficient visual search of videos cast as text retrieval" (PDF). IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 4. opposition. pp. 591–605.*