LANGUAGE DETECTION USING MULTINOMIAL NAÏVE BAYES ALGORITHM

By

YASHVI VAGHASIYA * DIYA VORA ** NEHA YADAV ***

MANISH RANA ****

*-**** *Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India.*

Date Received: 04/10/2022 Date Revised: 15/10/2022 Date Accepted: 26/10/2022

ABSTRACT

In this multilingual world, automatic detection of written or spoken language using Language Identification (LID) technology is a boon in the global communication with people using different languages in different countries. For simplicity and for the purpose of this research, the process of automatically identifying the language(s) from a document is thought of as LID. Lot of ongoing research projects are in the field of Natural Language Processing (NLP) that uses LID as a part of NLP. This field exploits several algorithms evolved in the field of computer science, individually or in combination to achieve accuracy in identifying a language. Among the different approaches adopted in LID,NaïveBayes Classification n-gram text processing seems to be promising.This paper proposes the concept for categorising multiple language texts using Naïve Bayesian algorithms using Machine Learning approaches. Using techniques from existing researches, this paper proposes a way to recognize multilingual documents and calculate the relative proportions of these languages.

Keywords: Language Identification, N-Gram Model, Multilingual Naïve Bayes Algorithm, Classification, Natural Language Processing (NLP).

INTRODUCTION

Thousands of languages are in use in this world for regional and community communication. In this globalization era, people from one linguistic and cultural background would have a necessity to understand another language and respond appropriately. Currently, many Natural Language Processing (NLP) techniques take advantage of text processing algorithms to improve the performance of word processing applications and reduce the amount of search space required. For example, by identifying various demographic properties of documents in written texts, one can determine the structure of a language.

In another aspect, language detection can be a language processing task when it wants to determine the language of a text or document. LID using artificial intelligence and machine learning was a challenge a few years ago as there was insufficient datasets on the languages. Today, with abundance of computer generated data available in languages, and with powerful machine learning models, researches on LID is gaining momentum.

LID has generated a lot of interest due to its applications in key areas of research such as Machine Translation (MT), Speech Recognition (SR), and Data Mining (DM). The development of various approaches in the field of LID, with the aim of creating more efficient systems, as well as significant problems that still face the field, have been caused by a significant increase in the volume of and access to data provided not only by experts but also by users on the Internet. Even though LID systems are used in a wide variety of applications today, the fundamental ideas that were created and launched in the 1990s are still in use (Muthusamy, 1993). A significant number of methods have been created for the development of the LID activity sector.

LID methods are often based on the idea that a language document is digitally available in one of all closed known languages in which training data is offered, and is described in the process of selecting the most likely training language from the collection in the database. This work proposes to solve the problem of LID in text documents from a database of candidate language datasets.

Language detection in a written text is probably one of the most basic tasks in Natural Language Processing (NLP). The first step in any language-dependent processing of an unknown text is to determine the language in which it was written. Any language has a distinct set of joint occurrences of characters. This approach is based on three linguistic characteristics, such as character set, N-grams (Rahmoun & Elberrichi, 2007), and word list (Kanariset al., 2006).

1. Concepts and Definitions

Several key concepts and basic definitions are,

1.1 Character Set

Every language has a character set. A family of languages may have common character set, yet the words formed would belong unique to different languages in the same family of languages. For example, the Indian languages Marathi and Sanskrit, have a similar set of characters, while Hindi and Gujarati would have different script in its written form.

1.2 N-gram Algorithm

A continuous set of n elements from a given sample of text or speech is known as an "N-gram". Depending on the application, the objects can be letters, words, or base pairs. N-grams are usually assembled from a corpus of text or voice (Cavnar&Trenkle, 1994).

1.3 Wordlist

The words, in any language serve as the ultimate source of abstraction. Some languages, like Marathi and Hindi, have very similar character sets and occurrences of n-grams.

2. Literature Review

Gold (1967) characterized language identification as a closed-class problem involving data in each of a predetermined set of possible languages, and people were asked to determine the language of a particular test document.

Linguistically motivated models have also been used for language identification. Johnson (1993)selected the language with the highest stop word match for a given document from a list of stop terms from multiple languages to determine the language of the text. Grefenstette (1995) determined whether two text samples were the same language or different languages using word-part-of-speech (POS) correlation. The cross-language tokenization model was created by Giguet (1995), to determine the language of a given text based on how tokenization patterns are compared to training data. Lins and Gonçalves (2004) use closed grammar class models that were derived syntactically instead of matching words or character sequences.

The most direct application of language identification is in multilingual text search, where search results are generally better if the language of the query is known and the search is limited to those documents predicted to be in that language (Qafmolla, 2017). Gazeau and Varol (2018) to improve the efficiency of parsing, it can also be used to "word-define" foreign language terms in the multilingual document. It can also be used to generate linguistic corpora (Gadgil et al., 2018).

3. Methodologies and Mathematical Modelling

3.1 N-gram

An N-gram can refer to either an n-set of words or an n-set of characters. This model represents documents using the n-gram frequency vector rather than the term frequency vector. A string of n consecutive characters makes up an n-gram character. All the n-grams that can be created for each document are obtained by moving a window of n-blocks across the text. A symbol for n-grams of symbols and a word for n-grams of words correspond to one stage in this movement. The frequency of detected n-grams is then measured. The term "n-gram" is sometimes used in the scientific literature to describe sequences that are neither ordered nor straight; for example, a bigram may consist of the first and second letters of a word. A resulting n-gram is a group of n consecutive characters. With this method, the corpus does not require any language processing. Moving a window of n blocks in the body text results in extracting all n-grams for a particular document (often n = 2, 3, 4, or 5). Each character move is like taking a "picture," and all these "frames" together make up the set of all n-grams in the text. Using the chosen value of n, it trimmed the corpus texts. The different types of n-gram models are,

- unigram: p(wi) (i.i.d. process)
- bigram: p(wi |wi−1) (Markov process)
- trigram: p(wi |wi−2, wi−1)

Equation 1 can estimate the probabilities of n-grams by computing the relative frequency in the training corpus.

$$\hat{p}(w_a) = \frac{c(w_a)}{N}$$

$$\hat{p}(w_b \mid w_a) = \frac{c(w_a, w_b)}{\sum_{w_b} c(w_a, w_b)} \approx \frac{c(w_a, w_b)}{c(w_a)} \qquad (1)$$

where N represents the total number of words in the training set and c(•) represents the number of words or word sequences in the training data.

3.2 Naïve Bayes Classifier

The Naïve Bayes classifier uses the notion of Bayes' theorem. Based on the highest posterior for the input string, this classifier provides the most likely class for the input string. N-grams can be used as features when constructing the Nave Bayes classifier for LID. Let T be a set of training samples, and let each sample be represented by n feature vectors, X = x_1, x_2, ...,x_n, with the class labels. Let there be m classes, such as K_1, K_2, ..., K_m. In Equation 2 for prediction, sample X is chosen to belong to class K_i if and only if,

$$P(K_i \mid X) > P(Kj \mid X); \text{for } 1 \leq j \leq m; j \neq i \qquad (2)$$

where P (K_i | X) is the probability of a class K_i given a sample. In Equation 3 Bayes' theorem states that,

$$p(K_i \mid X) = \frac{p(X \mid K_i)\, p(K_i)}{p(X)} \qquad (3)$$

where P(X|K_i) represents the probability that sample X belongs to class K_i, and P(X) does not affect the comparison of models. The relative count frequency in the sample is the class prior probability P(K_i). According to the assumption of Naive Bayes, the statistical independence functions are assumed, and the class Ki is chosen such that j P (x_j | K_i) P (K_i) is optimized, where P (x_j | Ki) is the probability that a particular n-gram is observed in the given language and the classified word consists of j n-grams.

3.3 Multinomial Naïve Bayes Algorithm

The Multinomial Naive Bayes algorithm is a probabilistic learning method that is used primarily in Natural Language Processing (NLP). The algorithm uses Bayes' theorem to predict the tag of a piece of text, such as an email or a newspaper article. It determines how likely each tag is for a given sample and then produces the most likely tag as an output. A naive Bayes classifier is a group of algorithms that follow the same rule, every feature that is classified is not related to any other feature.

It calculates the probability of an event occurring based on prior knowledge of the conditions associated with the event. It is based on the following Equation 4.

$$P(A|B) = P(A) * P(B|A)/P(B) \qquad (4)$$

where, P(B) = prior probability of B,

P(A) = prior probability of class A,

P(B|A) = occurrence of predictor B given class A probability.

4. Proposed Concept

This paper proposes a concept explained by Shanmugam (2017), which was used for managing customer complaints of a transport operator, for classifying customer emails and redirecting them to the respective customer service agents/queueas shown in Figure 1.

In Natural Language Processing, language identification or language guessing is the matter of determining which

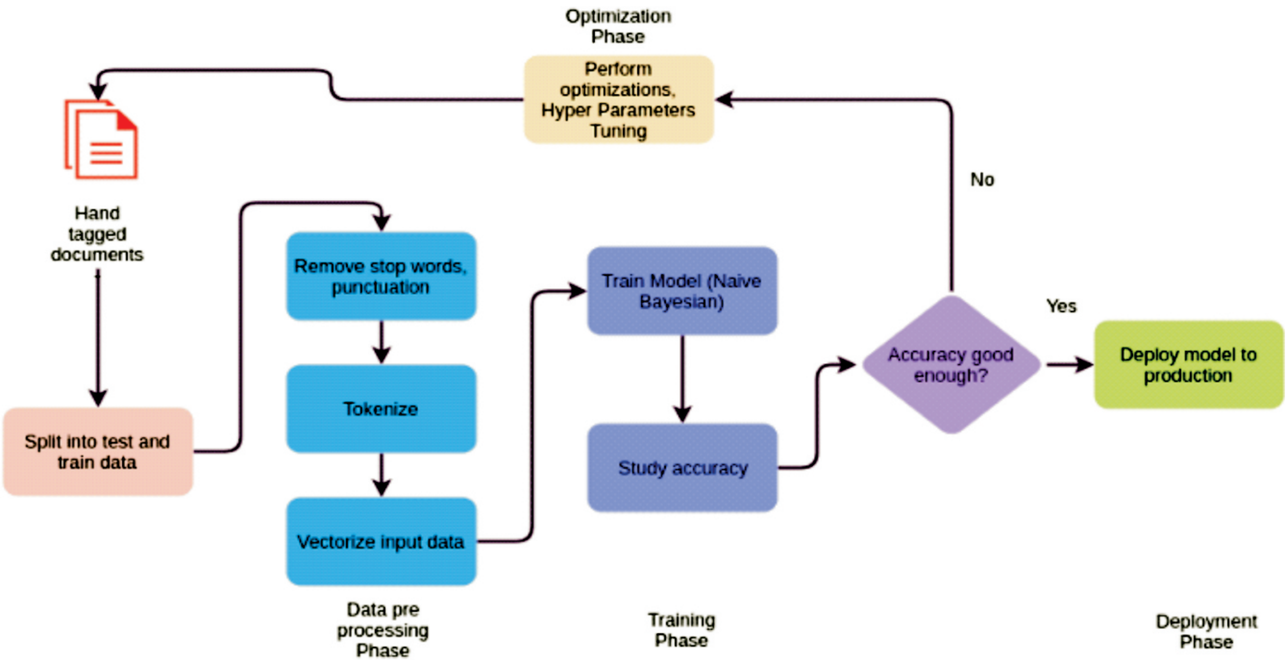
Optimization Phase
Perform optimizations, Hyper Parameters Tuning
Hand tagged documents
No
Remove stop words, punctuation
Train Model (Naive Bayesian)
Yes
Accuracy good enough?
Deploy model to production
Tokenize
Split into test and train data
Study accuracy
Vectorize input data
Data pre processing Phase
Training Phase
Deployment Phase

Figure 1. Proposed Language Detection Methodology Adopted from Shanmugam (2017)

natural language a given content is in. Computational approaches to the current problem view it as a special case of text categorization, solved with various statistical methods. There are several statistical approaches to language identification, using different techniques to classify the information. One technique is to match the compressibility of the text to the compressibility of texts during a set of known languages. This approach is understood as mutual information based distance measure. The identical technique may be accustomed to empirically construct family trees of languages which closely correspond to the trees constructed using historical methods. Mutual information based distance measure is actually accepted more than conventional model-based methods and is not generally considered to be either novel or better than simpler techniques. Another technique, as described by Caviar and Treble (1994); Dunning (1994) is to make a language n-gram model from a "training text" for every language. These models may be supported characters (Caviar & Treble 1994) or encoded bytes (Dunning 1994); within the latter, language identification and character encoding detection are integrated. Then, for any piece of text desirous to be identified, the same model is created, which model is compared to every stored language model. The foremost likely language is the one with the model ,that's most like the model from the text, which needs to be identified. This approach may be problematic when the input text is in a language, where there's no model. Therein case, the strategy may return another, "most similar" language as its result. Also, problematic for any approach are pieces of input text that are composed of several languages, as it is common online.

Malato (2020) the model used is a Multinomial Naive Bayes, which is a very simple model and very powerful when it comes to Natural Language Processing. It almost has no hyper parameters, so can focus on the pre-processing phase, which is the most critical. According to scikit-learn docs, Multinomial Naive Bayes can take count vectors as input features, which is exactly what is required. It is proposed to create a model that, once fed with a text, is able to detect its language. The text can be a sentence, a word, a more complicated text, and so on. The output variable can be, for example, the language code (like “en” for English). A good idea would be to have a model that detects the language of a text even if this text contains words that the model has not seen in the training phase. We want a model that can generalize the underlying structure of a language in a way that makes it detect it properly (Malato, 2020). The same methodology is proposed for identifying a language from the family of North Indian languages having almost identical written form or script.

Conclusion

This paper discussed basic concepts used in language classification and identification based on earlier works. Especially, the paper focused on technologies such as N-gram and applying ML using Naïve Bayes algorithm for classification, and the model is good even with mixed-language sentences. An application developed using this concept for identification of North Indian Languages would be highly useful.

References

[1]. **Babu, V. J.,&Baskaran, S. (2005, February).** Automatic language identification using multivariate analysis. In International Conference on Intelligent Text Processing and Computational Linguistics, (pp. 789-792). *Springer, Berlin,* Heidelberg.https://doi.org/10.1007/978-3-540-30586-6_89

[2]. **Cavnar, W. B., &Trenkle, J. M. (1994, April).** N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval,* (Vol. 161175).

[3]. **Dunning, T. (1994).** *Statistical Identification of Language (pp. 940-273).* Las Cruces: Computing Research Laboratory, New Mexico State University. Retrieved form https://www.researchgate.net/profile/Ted-Dunning/publication/2263394_Statistical_Identification_of_Language/links/0deec51cb2675ae546000000/Statistical-Identification-of-Language.pdf

[4]. **Gadgil, A., Joshi, S., Katwe, P., & Kshatriya, P. (2018).** Automatic language identification using hybrid approach and classification algorithms.International *Research Journal of Engineering and Technology,* 5(3), 3178-3181.

[5]. **Gazeau, V., &Varol, C. (2018).** Automatic spoken language recognition with neural networks. I.J. *Information Technology and Computer Science,* 10(8), 11-17. https://doi.org/10.5815/ijitcs.2018.08.02

[6]. **Giguet, E. (1995, September).** Categorization according to language: A step toward combining linguistic knowledge and statistic learning. In *International Workshop of Parsing Technologies (IWPT'95)*.

[7]. **Gold, E. M. (1967).** Language identification in the limit. *Information and Control,* 10(5), 447-474. https://doi.org/10.1016/S0019-9958(67)91165-5

[8]. **Grefenstette, G. (1995, December).** Comparing two language identification schemes. In *Proceedings of JADT,* 95, 263–268.

[9]. **Johnson, S. (1993).** Solving the Problem of Language Recognition. *Technical Report, School of Computer Studies.*

[10]. **Kanaris, I., Kanaris, K., Houvardas, I., &Stamatatos, E. (2006).** Words versus characters n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools,* 16(6), 1047-1067. https://doi.org/10.1142/S0218213007003692

[11]. **Lins, R. D., & Gonçalves, P. (2004, March).** Automatic language identification of written texts. In *Proceedings of the 2004 ACM Symposium on Applied Computing,* (pp. 1128-1133). https://doi.org/10.1145/967900.968129

[12]. **Malato, G. (2020, October 8).** *An Efficient Language Detection Model using Naive Bayes; a Simple Language Detection Model in Python,* Retrieved from https://towardsdatascience.com/an-efficient-language-detection-model-using-naive-bayes-85d02b51cfbd

[13]. **Muthusamy, Y. K. (1993).** *A Segmental Approach to Automatic Language Identification.* (Doctoral dissertation). Oregon Graduate Identification of Science and Technology, Portland, Oregon, USA.

[14]. **Qafmolla, N. (2017).** Automatic language identification. *European Journal of Language and Literature,* 3(1), 140-150. https://doi.org/10.26417/ejls.v7i1.p140-150

[15]. **Rahmoun, A., & Elberrichi, Z. (2007).** Experimenting n-grams in text categorization. *International Arab Journal of Information Technology,* 4(4), 377-385.

[16]. **Shanmugam, D. C. (2017, October 30).** *How Redbus*

uses Scikit-Learn ML Models to Classify Customer Complaints? Retrieved form https://medium.com/redbus-in/how-to-deploy-scikit-learn-ml-models-d390b4b8ce7a

ABOUT THE AUTHORS

Yashvi Vaghasiya, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India.

Diya Vora, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India.

Neha Yadav, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India.

Dr. Manish Rana is currently working as an Associate Professor in the Department of Computer Science at Thakur College of Engineering and Technology, Mumbai, India.