Bag of words with Naive Bayes to detect languages

Santiago E. Bocel
Universidad Rafael Landívar
Guatemala City, Guatemala
santiagobocel10@gmail.com

Roberto Solares
Universidad Rafael Landívar
Guatemala City, Guatemala
betosolaresgar@gmail.com

Brenner Hernandez
Universidad Rafael Landívar
Guatemala City, Guatemala
velasquezbrenner@gmail.com

Pablo Muralles
Universidad Rafael Landívar
Guatemala City, Guatemala
pablomuralles28@gmail.com

ABSTRACT

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content.

There are different types of text classifiers, for example, language detection, process automation, virtual legislation, sentiment detection, etc. That is why the classification of texts is becoming an increasingly important tool, since it allows us to obtain information from data and make use of them quickly, something that is very important in the information age.

On the other hand, machine learning in its most basic form is the practice of using algorithms to analyze data, learn from it, and then make a determination or prediction about something in the world. Where there are endless techniques for learning, representation and optimization.

It is for these reasons that the combination of machine learning with text classification is a very powerful but at the same time very complex tool and a field in which there is still much to explore.

1 INTRODUCTION

Text classification is one of the applications of Machine Learning and consists of cataloging the texts based on their content, that is, performing an analysis of the words to decide what type of text is being identified.

This work is ideal for a machine as they are ideal for processing large amounts of information. However, since the machine does not initially know how to catalog a text based on any criteria, it requires a learning process in advance.

In this research, the Bag of Words (BOW) model will be used, which is a method used in language processing to classify words according to the tag.

Many language processing tasks involve a classification, in which different machine learning methods can be used such as Maximum Entropy (ME), Support Vector Machines (SVE), Naive Bayes (NB) and many more, that is why that in this work the Naive Bayes algorithm was used, more specifically

the Multinomial Naive Bayes algorithm. The Multinomial Naive Bayes algorithm is of great help as it proposes a solution to the categorization of text by assigning a label or a category to an entire text or document.

All these methods and tools can be applied in different types of classifiers such as the classification of feelings, which is the process of automating or identifying opinions in the text and labeling them as positive, negative or neutral, based on the emotions or labels that it possesses. each one, the spam classification of some text, the automatic generation of subtitles, among others. In the case of this research, we focus on knowing the language in which a text is written, language recognition.

This paper is structured as follows. The Fundamentals section describes the prior knowledge that the reader is recommended to possess in order to have a better understanding of the research. In the section The problem, the problem to be solved with this investigation is detailed as well as its requirements. The Similar Implementations section tries to describe the state of the art by showing what techniques exist for text classification as well as what other experiments and projects have solved similar problems. In the section Our Solution, it is explained in detail how the solution was carried out, as well as why certain decisions were made and what problems were encountered when carrying out the same. The Results section describes what information could be obtained both in the training phase as well as in the experimentation and testing phase. Finally, the Conclusions section expresses the thoughts and observations of the authors after conducting the research.

2 FUNDAMENTALS

2.1 Artificial intelligence

Its origins were in robotics but while this hardware was being made there was a need to create software that allows this hardware to appear to have intelligence. Artificial intelligence seeks to be able to understand, perceive, predict and to some extent be able to manipulate its environment. It has four approaches which are the following systems that think like humans, systems that act like humans, systems that think rationally and systems that act rationally.

Artificial Intelligence has different foundations such as the philosophy, mathematics, probability, economics, neuroscience, communication and much more. Computational engineering is the most important providing the artifact, that is, the computer and the software, operating systems, the infinity of programming languages and the same tools to be able to generate applications. Later the theory of control and cybernetics that shared the theory of control, cybernetics and the objective function. And to finish the linguistics that computational linguistics contributed.

There are different branches of artificial intelligence such as: Robotics, strong AI, reasoning and decision making, knowledge representation, planning, computer vision, machine learning, data mining, and language processing.

Machine learning and data mining consists of machines learning without having to be explicitly programmed for that task by means of previous data. Language processing is how a computer can analyze audio and then convert it to text for analysis.

2.2 Machine Learning

It consists of machines learning without having to be explicitly programmed for a given task, relyng

only on previous data.

Some of the fundamentals for this branch are inferential statistics, computer science and pattern recognition. Going deeper with statistics, Bayes' theorem is very important since it tells us the probability that an event will occur given the knowledge of certain previous conditions related to this event. Scientists investigated how to apply the biology of human neural networks to machines. This ended with the creation of artificial neural networks, a computer model that is based on the way our neurons share information with each other through a network of interconnected nodes. Following this, MIT Dean Edmonds and Marvin Minksy created a program capable of learning through experience to get out of a maze.

Through inferential statistics, pattern recognition, and a sample of data, machine learning is able to draw inferences from new data sets for which it has never been trained. It does this by performing large data analyzes in order to deduce or infer which is the most optimal result for a certain situation. Taking into account that it has not been designed or programmed to carry out this task, rather it generates a capacity to learn through data and it is here where the step from programming through rules to autonomous learning is generated.

There are three main types of machine learning: supervised learning, unsupervised learning and reinforcement learning. Supervised learning is when the agent is trained through data that is an example of the inputs and outputs that it should have, that is, with data that is labeled.

Unsupervised learning consists of learning through input patterns where the values for the outputs are not said, it means that a specific type is not identified, but similarities are sorted and grouped. Finally, learning by effort, this is not taught with data but thanks to the environment that surrounds it and the actions it takes, depending on its result, it learns.

2.3 Joint and Conditional Probabilities

The joint probability function describes the probability of two events occurring simultaneously. In practice this is just the multiplication of the probabilities of all events. While the conditional probability describes the probability of one event occurring in the presence of a second event. In practice this is the multiplication of the probabilities of one event if the other event occurs and the probability of the other event to occur.

As we can see, the joint probability and the conditional probability are basically equal, so they are equivalent and we can denote it as follows.

$$P(A|B)\ P(B) = P(A,B)$$

Figure 1: Joint and conditional probability

We can also express a joint probability in terms of chain of conditional probabilities and these are known as the chain rule.

$$P(A_1,A_2,A_3...,A_n) = P(A_1|A_2,A_3,..,A_n)P(A_2|A_3,\ ...,A_n)\ ...\ P(A_n)$$

Figure 2: Chain rule

2.4 Statistical inference

For an artificial intelligence to try to predict using some classification algorithm it must first learn its probabilistic theories about the world from experience. Statistical inference is the induction that allows establishing a truth with a higher probability index than the others. With this, artificial intelligence acquires a tool to solve specific problems, such as the classification of information.

In Bayesian learning methods, learning is formulated as a form of probabilistic inference, using observations to update a prior distribution on the hypothesis.

One simply calculate the probability of each hypothesis given the data, and make predictions on these samples. This means that the predictions are made using all the hypothesis, weighted by their probabilities and not only using the best of the hypothesis. In this way, learning is reduced to probabilistic inference.

2.5 Bayes Theorem

The Bayes theorem is used to calculate the probability of an event, having information in advance about this event. In this way, it is possible to calculate the probability of an event A, also knowing that the event A fulfills a certain characteristic that determines its probability.

Based on the above, it is understood that this branch of statistics is essential for the elaboration of inference rules that can represent a viable way of learning for the way in which machines analyze their results.

The Bayes theorem is attractive for artificial intelligence since it comes from the fact that its development is axiomatic, allowing constructive growth based on a single derivation rule, so that by significantly increasing the number of repetitions, there will be a perfectible improvement mechanism which will represent the foundation for learning by repetition, erring less and less until reaching an optimal knowledge of what is involved.

This theorem appear from the solving of the conditional probability equation, since we could see that it is equivalent to the joint probability.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = events
$P(A|B)$ = probability of A given B is true
$P(B|A)$ = probability of B given A is true
$P(A), P(B)$ = the independent probabilities of A and B

Figure 3: Bayes theorem

2.6 Naive Bayes

Naive Bayes is a simple and powerful algorithm. Despite the significant advances in Machine Learning in recent years, it has proven its worth. It has been successfully implemented in many applications, from text analysis to recommendation engines.

It is one of the simplest and most powerful algorithms for classification based on Bayes' Theorem with an assumption of independence between the predictors. It assumes that the effect of a particular feature on a class is independent of other features.

Using the algorithm it is quick and easy to predict the kind of test data set. They also work well in multiclass prediction. When the independence assumption is held, a Naive Bayes classifier performs better compared to other models and less training data is required. It works well for categorical input variables compared to numeric variables.

The disadvantages of using this algorithm are that if the categorical variable has a category in the test data set, which was not observed in the training data set, the model assigns a probability of 0 and will not be able to make a prediction. This is often known as the zero frequency. The smoothing technique can be used to solve this problem. Another disadvantage is the assumption of independent predictors. In real life it is almost impossible for us to obtain a set of predictors that are completely independent.

This algorithm serves a core part in our solution, wo we will talk more on how we implemented it.

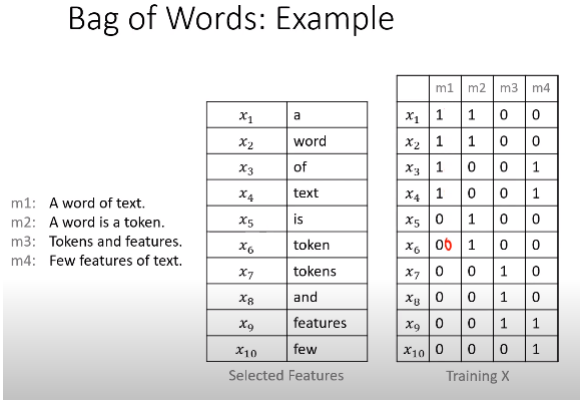
$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i \mid Cause)$$

Figure 4: Naive Bayes

2.7 Bag of Words

The Bag of Words is a method often used for document classification. This method turns text into fixed-length vectors by simply counting the number of times a word appears in a document, a process referred to as vectorization.

Although these vectorization methods are easy to compute, it lacks any contextual information. It literally is a bag of words – there is no order, it's only the word counts that matter. It is a data recovery system. It does the work of identifying all documents that are important for the user who seeks the information.



x_1	a
x_2	word
x_3	of
x_4	text
x_5	is
x_6	token
x_7	tokens
x_8	and
x_9	features
x_{10}	few

	m1	m2	m3	m4
x_1	1	1	0	0
x_2	1	1	0	0
x_3	1	0	0	1
x_4	1	0	0	1
x_5	0	1	0	0
x_6	0	1	0	0
x_7	0	0	1	0
x_8	0	0	1	0
x_9	0	0	1	1
x_{10}	0	0	0	1

Figure 5: Bag of Words

This is also a core part of our project, thus, we will detail how our solution uses this algorithm.

3 PROBLEM

We are four university students and are currently coursing our fourth year of computer engineering. In this semester we enrolled to the Artificial Intelligence course, and so, we were assigned with this final project. We had to implemente the fist stage of a text classification system to identify languages through input text.

For this, we were appointed to use a Bag of Words system, in conjunction with a Bayesian model, more specifically, Naive Bayes.

This implementation must support input text from files and user given phrases. These file inputs must follow a specific structure; the first part of the input must be phrase in any language, the second part must be a name tag describing the language in which that phrase was written; our solution will have to split this parts with a given separator, in this case being a '|' (pipe character). Some examples could be:

* la vida empieza cada cinco minutos | español
* it is a good day to be happy | ingles
* eu te quero com tudo meu coração | portugues

If the input is a user given phrase, we must be able to ask them for the corresonding name tag and associating each other.

The solution must process any given number of lines or phrases with any given number of labels. When inserting a new line or phrase the solution must be readjusted. It must be taken into account that when reading a CSV file or similar, the data should be normalized and cleaned. In addition, the solution must have a cold start option and at the time of interaction with the user, the recommendations must improve.

The sofware had to be develped using any JVM compatible language, with the option of using any

type of data base.

4 SIMILAR IMPLEMENTATIONS

Another type of algorithms commonly used in automatic classification of texts are:

4.1 Support vector machines(SVM)

This algorithm is a text classification method. It corresponds to learning machines that take different characteristics of the elements that want to classify and take them to a multidimensional vector space. It is in this space, where the algorithm identifies a hyperphan that separates the vectors into a class different of the rest.

The solution for a Bag of Words classification is simple. Suppose that in the following image the blue dots represent a language and the red dots other language, if a line is drawn between them, when there is a new point, we can say what color is going to have, depending on the side of the line in which you are.

Figure 6: Support vector machine

The line that best separates the zone of the blue dots from the red dots area is the line that maximizes the margin between them. Support vector machines are a Machine Learning technique that finds the best possible separation between classes. With two dimensions it is easy to understand what you are doing. Normally, automatic learning problems have a lot of dimensions. So instead of finding the optimal line, the SVM finds the hyperphan that maximizes the margin of separation between classes.

One of the most advanced agents is Karspersky's LightAgent product, which tries to research and offer cybersecurity in large organizations. This is done in python and with libraries like TensorFlow and NumPy.

4.2 K-nearest neighbors algorithm (KNN)

This is a learning algorithm based on supervised type instances of Machine Learning. It is used to classify new samples or to predict. It is a simple method, looks for the closest observations to which you are trying to predict and classify the point of interest based on most data that surrounds it.

The solution for a Bag of Words classification is also simple.

The distance between the word to classify and the rest of the words of the training dataset should be calculated. Then select the closest elements (with less distance, according to the function that is used). Finally, perform a "majority vote" between the K points: those of a class / dominant label will decide its final classification.

Since the KNN algorithm is highly used in recommendation systems, it is at the core of Netflix recommendation systems, so much so that Netflix hosts an annual competition in order to improve

its system more and more.

4.3 Different Types of Naive Bayes

As we can see, the ease of the naive bayes algorithm has made it one of the most used when dealing with text classification problems, especially those that use a bag of words model. There are several types of implementations of this, some of these are:

A great example of this is the Tunisian commercial bank that its using a Naive Bayesian classifier methodology for loan risk assessment.

4.3.1 Bernoulli Naive Bayes. The Bernoulli Naive Bayes algorithm is excellent to use when working with boolean variables, that is, when there is presence or absence of these, this is because the algorithm assumes that the features are totally independent.

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Figure 7: Bernoulli Naive Bayes Formula

4.3.2 Multinomial Naive Bayes. The multinomial naive bayes algorithm is excellent to use when working with several samples of different variables, since these represent the frequency of occurrence and make the problem can be represented as a histogram.

4.3.3 Gaussian Naive Bayes. The gaussian naive bayes algorithm is excellent to use when working with continuous data because making use of the mean and standard deviation gives us adjusted

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Figure 8: Bernoulli Naive Bayes Formula

data for each case.

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Figure 9: Bernoulli Naive Bayes Formula

5 OUR SOLUTION

In a way or another, we used all of the information previously exposed to acomplish this solution for the given problem. We developed a Bag of Words with Naive Bayes using the Java programming language.

The interaction that the user will have with our solution will be through the command line. They will be presented with a series of menus, firstly they will see the main menu:

```
What do you want to do?
  1) Train
  2) Infer
  3) Show Knowledge
  4) Exit
Put the number of the option:
```

Figure 10: Main menu

From this point, the user will be able to select any of the options inserting a number available

among the given options. If the user selects 'Training', they will see the training menu:

```
What kind of training do you want to do?
  1) One phrase
  2) Bulk file
  3) None
Put the number of the option:
```

Figure 11: Training menu

In the training menu, the user can insert a standalone phrase an its tag, or a file with multiple lines following the structure previously stated:

```
Insert the phrase: this is a demostration
Insert the tag: english
1 new words are analyzed
```

Figure 12: One phrase

```
You are in: /home/mochis/git/bag-of-words
Insert the path to the file: sample_data.txt
16200 new words are analyzed
```

Figure 13: Bulk file

All this would have been in vain if the backend of our agent did not work, it has two major tasks: training and inferring.

5.1 Training

In the training part we use the following structure:

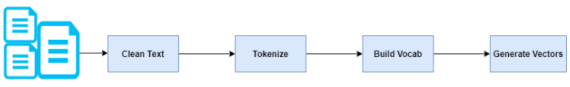


Figure 14: Training Structure

In the cleaning phase, the first thing we do is remove all misspelled words (letters combined with numbers) and the numbers themselves, since they do not add any value to any type of text in any human language. After this we remove stopwords if the text is in a certain tag.

Table 1: Stopwords per tag

WORD	NUMBER	WORD	NUMBER
Afrikaans	51	Arabic	162
Armenian	45	Basque	98
Bengali	116	Breton	126
Bulgarian	259	Catalan	218
Chinese	542	Croatian	179
Czech	346	Danish	101
Dutch	275	English	570
Esperanto	173	Estonian	35
Finnish	772	French	606
Galician	160	German	596
Greek	75	Hausa	39
Hebrew	194	Hindi	225
Hungarian	781	Indonesian	355
Irish	109	Italian	619
Japanese	109	Korean	679
Latin	49	Latvian	161
Marathi	99	Norwegian	172
Persian	332	Polish	260
Portuguese	408	Romanian	282
Russian	539	Slovak	110
Slovenian	446	Somalia	30
Southern Sotho	31	Spanish	577
Swahili	74	Swedish	401
Thai	115	Turkish	279
Yoruba	60	Zulu	29

5.2 Infer

At the time of inferring we do not clean the text since we do not know the exact label. What we do first is to calculate the conditional probabilities of each word given a certain tag. Finally, since we have our joint probabilities, we apply the Bayes theorem as can be seen in Figure 3.

Now suppose that a word that we are trying to infer is not found in our dataset, this would give

us a value of 0 so all our probability would be 0. We solve this using Laplace Smoothing, which is basically adding one to the frequency of each word.

$$\hat{\theta}_i = \frac{x_i + \mu_i \alpha d}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

Figure 15: Laplace Smoothing

So the algorithm and the workflow at the time of inferring would be as follows:

Input:

Training dataset T,

F= (f_1, f_2, f_3,.., f_n) // value of the predictor variable in testing dataset.

Output:

A class of testing dataset.

Step:

1. Read the training dataset T;
2. Calculate the mean and standard deviation of the predictor variables in each class;
3. Repeat

   Calculate the probability of f_i using the gauss density equation in each class;

   Until the probability of all predictor variables (f_1, f_2, f_3,.., f_n) has been calculated.
4. Calculate the likelihood for each class;
5. Get the greatest likelihood;

Figure 16: Naive Bayes Algorithm

6 RESULTS

As part of the user experience, one could see the knowledge owned by the solution:

As part of the results achived when trying to infer a language from an input, we got the following results with the training data detailed in Figure 18.

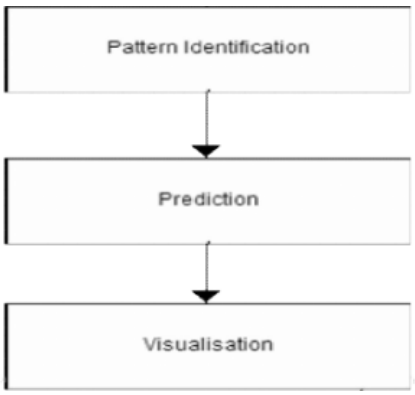


Figure 17: Infer Workflow

```
Total numbers of words: 16201
Tags: english, ingles, español, hindi
Words per tag:
ingles = 7214
español = 2739
english = 1
hindi = 6247
```

Figure 18: Knowledge base

```
Insert the phrase: this is our first acm paper

Features Set:
[this, is, our, first, acm, paper]

Probabilities:
english = 0.9999999963484988

The tag is: english
```

Figure 19: Inference results

It was also observed that there are occasions in which the agent fails in the classification due to the fact that when smoothing is applied, some tags that have few words are left with very high probabilities.

7 CONCLUSIONS

- One of the greatest advantages of Naive Bayes over other classification algorithms is its ability to process an extremely large amount of data.

- The Naive Bayes algorithm is simple to implement, works well from the beginning and adjusting its parameters is rarely necessary.

- Even though Naive Bayes has great advantages, a smoothing technique to prevent probabilities to be zero was necessary.

- Considering the amount of data they can handle, the training and process of prediction are very fast even when they start cold.

- Automatic learning was specifically applied to the supervised learning branch.

- The most difficult part of creating an intelligent agent is not applying the algorithms, but rather selecting and cleaning the data so that it is meaningful and good results are obtained.

REFERENCES

[1] AshishSingh Bhatia and Bostjan Kaluza. 2018. *Machine Learning in Java*. Packt, Birmingham, Uk.

[2] Zdravko Botev. 2006. Joint and Conditional Probabilities. Technical report. University of New South Wales.

[3] Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. 1979. Updating the formula and a pairwise algorithm for computing sample variaces. Technical report. Stanford University.

[4] Praveen Dubey. [n. d.] An introduction to bag of words and how to code it in python for nlp. https://www.freecodecamp.org/news/an-introduction-to-bag-of-words-and-how-to-code-it-in-python-for-nlp-282e87a9da04/.

[5] Srinivas Gurrala. [n. d.] Implementation of bag of words using python. https://www.excelr.com/blog/data-science/natural-language-processing/implementation-of-bag-of-words-using-python.

[6] Geoff Hulten. [n. d.] Simple features with bag of words for machine learning. https://youtu.be/zrMeJh3z23I.

[7] Kaspersky Lab. [n. d.] The svm algorithm. https://support.kaspersky.com/KSVLA/5.0/en-US/152311.htm.

[8] Michael Lanham. 2020. *Practical AI on the Google Cloud Platform*. Jonathan Hassell, editor. (1st. edition). O'Reilly Media, California.

[9] Monkey Learn. [n. d.] What is text classification? https://monkeylearn.com/what-is-text-classification/.

[10] Rajat Mehta. 2017. *Big data analytics with Java*. Packt, Birmingham, Uk.

[11] michael copeland. [n. d.] What's the difference between artificial intelligence, machine learning and deep learning? https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

[12] Stuart J. Russell and Peter Norvig. 2021. *Artificial intelligence: a modern approach*. Pearson, Hoboken.

[13] Muhammad Firman Saputra, Triyanna Widiyaningtyas, and Aji Wibawa. 2018. Illiteracy classification using k means-naïve bayes algorithm. *JOIV : International Journal on Informatics Visualization*, 2, (May 2018), 153. DOI: 10.30630/joiv.2.3.129.

[14] Madison Schott. [n. d.] K-nearest neighbors (knn) algorithm for machine learning. https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26.

[15] @timleathart. [n. d.] How to handle a zero factor in naive bayes classifier calculation? https://datascience.stackexchange.com/questions/15526/how-to-handle-a-zero-factor-in-naive-bayes-classifier-calculation.