Analysis on Formality Classification in Conversational Bots

Jawil Ricauter Dodero[1], Federico Flaviani[1], Sebastián Alarcón[1]
jwricauter@gmail.com, fflaviani@usb.ve, sebastian.alarcon@gmail.com

[1] Departamento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Caracas, Venezuela

Abstract: One of the main problems that conversational agents or chatbots face is the detection of context in conversations. This investigation addresses the detection of language style (formal or informal) in conversations, its implementation for chatbots and a flow of action to change the context of conversations in situ and, therefore, the conversation itself. The investigation consists of four phases: statistical analysis of the data, data preprocessing, classification and evaluation. During the data statistical analysis, the basic characteristics of the data obtained from chats are analyzed; in the preprocessing phase, the common techniques for text mining are used, such as stemming, elimination of stopwords, among others. During the classification phase, four algorithms were evaluated to determine the best one for detecting this type of context in chats. In the evaluation phase a flow was proposed using the best classifier to adapt the messages that the chatbot sends to the style of the interlocutor, recognizing the context of their language with an accuracy between 71% and 80%, which is improved with the feedback gives the user himself to the system and the proposed action strategy.

Keywords: Contextual Chatbot; Natural Language Processing; Text Classification.

1. Introduction

Conversational bots or Chatbots are conversational software systems that are designed to emulate the communication capabilities of a human who automatically interacts with a user. The use of this type of systems has revolutionized the fields of communication, customer service and public relations in general, speeding up requests and questions from users to public and private entities [1].

The use of a specific type of language is a turning point in public relations. The use of informal language in a certain target clients can create an environment of closeness and trust between the provider of goods and services and the client, just as the use of formal language is a standard of appropriate use that instills respect and seriousness.

Text mining and natural language processing are key tools for developing agents to deal with the aforementioned problem. The text mining processes in instant messaging services and social networks in general require a special ability to mine dynamic data, which often contains a poor and non-standardized vocabulary [2]. The use of these techniques to detect context in conversations and user choices is widely used to give personalized treatment to users, in the case of global and very popular chatbots, such as Siri and Alexa that seek to start to talk as naturally as possible using recognition agents to detect context changes in conversations with users [3] [4].

PANA Technologies is a Venezuelan company that is committed to a technological approach to the problem of security in the region, providing a simple and effective platform to help drivers and users in general. The growth of the company brought with it the increase in projects related to it. One of them, the Penny chatbot, was created with the intention of facilitating communication between PANA Technologies and potential customers. In its second version, a feature was developed that included the sale of subscriptions through Stripe for the WhatsApp instant messaging platform, as well as the implementation of a natural language processing model to turn the conversation that the chatbot had into an experience more similar to what they would have with a human.

This work presents a comparative analysis between different supervised learning strategies such as the Naive Bayes classifier, logistic regression, support vector machine and gradient boosting using data extracted from archived chats, with the aim of determining the model that best fits the dynamic scenario of instant messaging. Related works in contextual data analysis using machine learning techniques are described in Section 2. Investigation results are explained in Section 3. The methodology to be followed in this research and its application is described in Section 4. Section 5 presents and discusses the results. Finally, Section 6 concludes the paper and discusses future directions.

2. Related Works and Analysis

The field of natural language processing in chatbots is still in early stages. Therefore, relatively similar works could be found; however, they were not directly applied to chatbots or conversation style detection.

In a study carried out by the University of Rochester and the Amazon Alexa Machine Learning [10] team, an improved contextual language model for conversational agents was recreated which can be adapted to each user based o generalized information of previous conversations' context. In their application in chatbots, they use a classifier designed to estimate conversation topics; as in this paper, the data is taken from a database of multiple conversations with users. Unlike this research, which has a smaller focus, they use a deep neural network to detect a wide range of topics in conversations.

Another study conducted by the VNU International School and the Vietnam National University focuses on the recognition of intents and general contexts, such as time and place in the Vietnamese language [13]. For this, they proposed a framework that models the intents problem as a classification problem and the contextual problem as a simple neural network.

3. Contribution

This work's main contribution is a series of steps that can be followed to successfully reproduce a context-aware chatbot that can dynamically adapt its responses to the other party's formalism style with ease. By using a previously trained Bayes classifier, with a relevant set of phrases associated with their formalism level, the chatbot can decide how to match the formalism level it is presented to by comparing as few as three phrases to the pretrained data.

Using the classifier described in this work, a grouping algorithm was designed and implemented, through which each conversation is labeled and classified according to a level of formality. With the data that results from this classification, a conversational flow can be generated at one of the possible levels. This flow contains dozens of predesigned phrases that address the most common topics customers tend to inquire about, either from the company or conversations without specific orientation. The response to those phrases also feeds the classifier giving more training material.

4. Methodology Application

To formulate the methodology to be followed, different sources were searched. Specifically, there is a study that compares different ways of approaching a research work in the machine learning field [12]. The process to be developed is made up of an analysis in effective detection of formal and informal language styles in a dynamic set of inputs, such as text messages.

4.1. Dataset

There were 1131 phrases cataloged as formal and informal gathered from user conversations with the public relations team of the company PANA Technologies in Caracas, Venezuela. Of all these data, 70% (792 sentences) were taken from them to be used as training data and the rest 30% (339 sentences) as test data. The labels added to these phrases correspond to some inherent characteristics of informal phrases (abbreviations, hypochorisms, wildcards, proclitical forms, use of future periphrasis, colloquial adverbs, metaphorical expressions, redundancies, presence of imperative voice and vulgarities) and Formal (Full terms, original names, enclitic forms, morphological future, formal adverbs, evasion of use of imperative form, vulgarities and redundancies) in Spanish.

4.2. Exploratory Data Analysis

The first thing we observe is that there is small training dataset, which we can consider our first obstacle, so we must achieve a classification model that is effective with a short training process. The data set is divided between 643 informal sentences and 488 informal sentences, representing 56.8% and 43.2% respectively of the total data. The sentence length distribution by style is shown in Figure 1 and Figure 2.

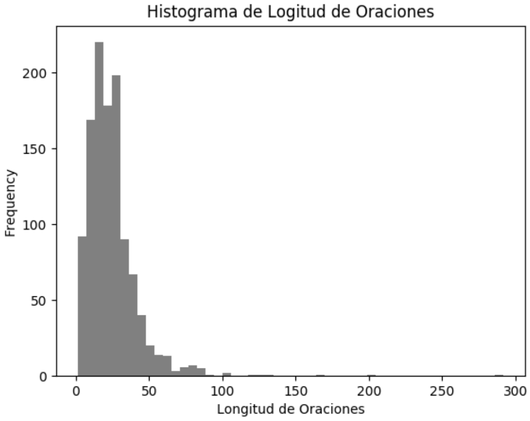

Figure 1: Phrase Length Histogram

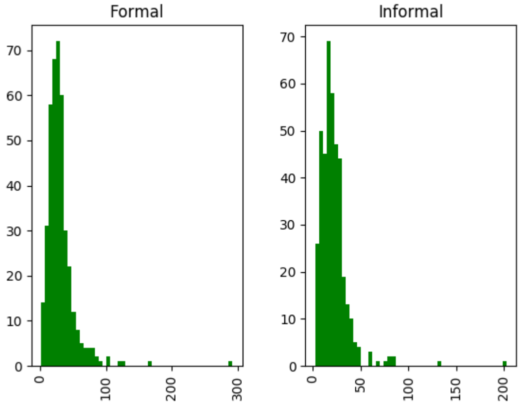

Figure 2: Phrase Length Histogram by Style

Stop words are very frequent words in texts that are not significant and do not contribute anything to the analysis [5], These words were removed after tokenizing each phrase and subsequently creating a word cloud for each category (formal and informal). The word cloud shown in Figure 3 shows a high presence of colloquial words in the informal category, as opposed to the formal word cloud shown in Figure 4.

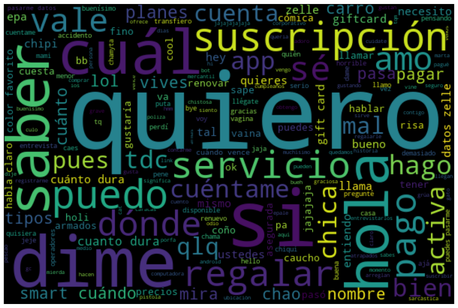

Figure 3: Informal Phrases Wordclod

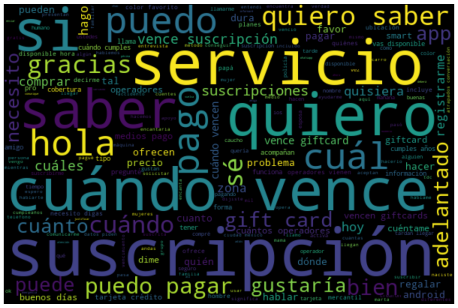

Figure 4: Formal Phrases Wordclod

Another important characteristic of the dataset is the mean of the sentences' length. For formal sentences, the mean is 31 characters, and for informal sentences it is 19 r. This implies a trend of shorter sentences in the informal category.

4.3. Preprocessing Phase

One of the challenges in text mining is converting unstructured data and semi-structured text into a structured vector-space model [6]. Due to this, a pre-processing phase must be carried out before any text mining or advanced analysis is performed. The possible steps for word processing are almost the same for all text mining tasks. The basic steps are as follows [6]:

- Choose the scope of the text to be processed
- Tokenize the text
- Remove the stopwords
- Stem
- Detect sentence limits
- Normalize case

In this case we will use all these techniques excluding the detection of sentence limits, since a large part of the messages that are used in instant messaging are phrases and not large texts.

a) Choose the scope of the text to be processed: Due to the short nature of text entries in the database, it is easy to determine the scope of the text, as they can be transformed into a single vector per entry.

b) Tokenize: In this part, the words were separated into discrete words called tokens.

c) Remove stopwords: This part was performed beforehand in order to analyze the dataset. Removing stopwords reduces the dimensionality of the term space, given that the most common words in text documents are articles, prepositions and pronouns that do not provide significance in the documents [7].

For removing these stopwords, was used the Python NLTK library, considering that it provides a list of stopwords par excellence in many languages, including Spanish, which the dataset uses.

d) Stem: This technique is used to remove prefixes and suffixes and thus normalize words, since they have variations that can be written differently actually mean the same [6]. For the implementation of this technique, the aforementioned NLTK library was used. Specifically, we used the string processor Snowball, that implements stemming algorithms for use in information retrieval [8].

e) Normalize case: Most of the texts in Romance languages are written in mixed case, that is, they contain uppercase and lowercase letters. Normalization converts the entire document to either lowercase (in this case) or fully uppercase [6].

In addition, a data feature extraction called Term Frequency inverse document frequency (TF-IDF) was used. This is a statistic measure that reveals how important a word is in a document taken from a set of documents. TF-IDF is often used as a weight factor in information retrieval and text mining [7]. The Term Frequency (TF) is defined as the number of times a term appears in a document.

An Inverse Document Frequency (IDF) is a statistical weight used for measuring the importance of a term in a text document collection. Then Term Frequency - Inverse document frequency [TF-IDF] is calculated for each word multiplying the term frecuency and the inverse document frecuency [7].

In mathematical terms, the TF-IDF score of a word t in a document d of a set of documents D is calculated as follows:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

where:

$$tf(t, d) = log(1 + freq(t, d))$$

$$freq(t, d) = count(t \in d)$$

$$idf(t, D) = log\left(\frac{N}{count(d \in D : t \in d)}\right)$$

4.4. Classification Phase

The pre-processed data was used in four algorithms, comparing their effectiveness for the context in which it is desired to apply. These algorithms are: Logistic Regression, Support Vector Machines, Naive Bayes Classifier and Gradient Boosting Machines.

4.5. Evaluation Fase

The performance of the four aforementioned algorithms will be measured by the following metrics:

- Accuracy
- Sensibility
- Specificity
- Fscore

Accuracy focuses on the effectiveness of the classifier, that is, the ratio of correctly classified sentences to the total number of sentences [8], and is measured with the following formula:

$$accuracy = \frac{tp + tn}{tp + fn + fp + tn}$$

Where, tp means true positive, tn true negative, fp false positive and fn false negative. This notation will also be used in other formulas throughout this document. Sensitivity or recall focuses on the effectiveness of a classifier to identify positive labels [8]. The formula is as follows:

$$sensitivity = \frac{tp}{tp + fn}$$

Specificity measures how effective a classifier is in measuring negative labels [8]. The formula is as follows:

$$specificity = \frac{tn}{tn + fp}$$

Finally, the Fscore measures the relationship between the positive labels in the data and those given by the classifier and its formula is given by:

$$fscore = \frac{(\beta^2 + 1) \cdot tp}{(\beta^2 + 1) \cdot tp + (\beta^2 \cdot fn) + fp}$$

Where beta is a factor of how much more important recall is than precision. The Fscore standard is equivalent to fix beta to one.

5. Results and Discussions

The results are divided into two parts: the first one presents the analysis of the proposed classifiers for handling phrases from conversations, and the second one focuses on the analysis of the chosen classifier algorithm's implementation in an instant messaging platform.

5.1. Performance of the Proposed Algorithms

The first algorithm to analyze is the logistic regression algorithm, which has a good performance in binary classification. The results presents an average precision of 71% and specificity of 76% on the test data. Their metrics are shown in Table I.

Table I: Logistic Regression Metrics

-	Accuracy	Sensitivity	Fscore
Formal	66%	73%	69%
Informal	77%	70%	73%

The second algorithm to analyze is the support vector machine algorithm, which has a lower average precision than the previous algorithm (69%) and a specifity of 74%, and its metrics are shown in Table II.

Table II: SVM Metrics

-	Accuracy	Sensitivity	Fscore
Formal	64%	70%	67%
Informal	75%	69%	72%

The third algorithm, Naive Bayes, has the highest mean precision found, with a 76 % total hit rate in the data and a specificity of 79 %. The rest of the data is shown below in Table III.

Table III: Naive Bayes Metrics

-	Accuracy	Sensitivity	Fscore
Formal	71%	74%	73%
Informal	80%	78%	78%

The last algorithm, called extreme gradient boosting, has the worst performance of all algorithms, with an average accuracy of 65 % and a specificity of 66 %. The remaining metrics are presented in Table IV.

Table IV: Gradient Boosting Metrics

-	Accuracy	Sensitivity	Fscore
Formal	62%	53%	58%
Informal	67%	75%	71%

The metrics of all four algorithms were much better in detecting informal language than they were for formal language. This can be explained due to the use of colloquial words, wildcards, etc., which are ways of expression unique to informal language that don't occur in formal conversations. The algorithm that had the best performance was the naive Bayes, which is characterized by working well in many real-world situations and also requires a small amount of training data to estimate the necessary parameters [9].

Due to these reasons, and because of it being faster compared to other sophisticated methods like the other algorithms used in this study, it adapts very well to dynamic environments with a small training dataset, such as the one used in this work.

5.2. Analysis of the Implementation of the Classifier in Real Conversations

The chosen classifier is present in many machine learning libraries in a variety of programming languages. Since it is very popular and is present in almost all machine learning libraries. It can work together with a chatbot made completely from scratch with the use of popular natural language processing tools like Keras in Python or it can be used as reinforcement of agents built with platforms for creating chatbots through webhooks or APIs.

For the implementation in the company system, the prediction model was added to a lattice system that connects the PANA backend to DialogFlow (a tool for creating chatbots made by Google) through Twilio, in addition to various connections to other platforms like Stripe, among other connections that are not named since they are outside the scope of this project. Figure 5 shows the communication network between the various platforms used.

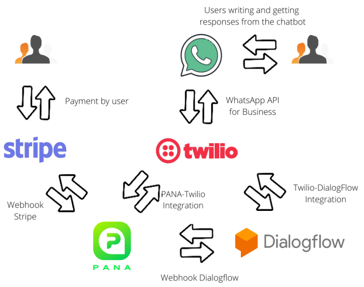
Users writing and getting responses from the chatbot
Payment by user
WhatsApp API for Business
stripe
twilio
Webhook Stripe
PANA-Twilio Integration
Twilio-DialogFlow Integration
PANA
Dialogflow
Webhook Dialogflow

Figure 5: Chatbot Penny Arquitecture

A data persistence model was used to avoid training the model every time a request is made to classify data, giving a faster response time in the chat. In addition to saving the model data in a file for later extraction, the data of the user's conversations is also saved. This is possible since each conversation in an instant messaging system has an associated identifier. The formal style was configured as the default context because it is the style that is less likely to be detected and also for reasons of public attention it is better to start this way.

User conversations change style, applying a response generator adapting to the context when a preponderance (>65%) is detected in the phrases of a style, setting a minimum number of interactions for this feature to start working (1 or 3 phrases is too soon). Since the naive Bayes classifier is very simple and also only retrained very few times, the response speed going through all the intermediate servers is between 1.2 and 2.6 seconds.

6. Conclusion

The fundamental contribution that this research gives to the study of text classification is to be able to give an efficient predictive model that can respond in a few seconds, that also feeds on a small amount of data and has an excellent performance in terms of precision, specificity and sensitivity. Bayes' naive classifier fits perfectly into this approach, providing good scores on all metrics (specificity, sensitivity, F1score, etc.) that were used to measure the performance of the proposed models. The flow and implementation of this model in chatbots provides a better adaptation of the agent's language style to that of the interlocutor when compared to other studied models.

Although this work has a small scope, it opens the way to a more ambitious study regarding an agent model that can detect not only language styles but multiple contexts and respond based on it automatically. Another possibility is the processing of multimedia files in chat, like audios and images, and adapting the response of agents to those files.

Acknowledgment

We would like to thank Sebastián Alarcón and the entire team of the company PANA Technologies for their support during the development of this research.

References

[1] M. Nuruzzaman and O. Hussain, *A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks*, 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE).

[2] C. Aggarwal and C. Zhai, *Mining Text Data*, Springer Science+Business Media, LLC 2012.

[3] Apple developer, *Human Interface Guidelines*. Avalaible https://developer.apple.com/design/human-interface-guidelines/siri/overview/introduction/.

[4] Amazon Alexa, *Alexa Skill Kit, Dialog Management*. Avalaible https://developer.amazon.com/es-ES/alexa/alexa-skills-kit/dialog-management.

[5] A. Rajaraman, J. Leskovec, and J. Ullman, *Mining of Massive Datasets*, Stanford Press, 2010.

[6] D. Delen, A. Fast, T. Hill, J. Elder, G. Miner, and B. Nisbet, *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*, Academic Press, 2012.

[7] S. Vijayarani, *Preprocessing Techniques for Text Mining - An Overview*, International Journal of Computer Science Communication Networks, Vol 5(1), 7-16, 2015.

[8] Snowball: Introduction. Avalaible https://snowballstem.org/.

[9] H. Zhang, *The Optimality of Naive Bayes*, Faculty of Computer Science, University of New Brunswick, Canada.

[10] A. Raju, B. Hedayatnia, L. Liu, A. Gandhe, C. Khatri, A. Metallinou, A. Venkatesh, and A. Rastrow, *Contextual Language Model Adaptation for Conversational Agents*, Amazon Alexa Machine Learning, University of Rochester, 2018.

[11] A. Paul, A. Latif, F. Adnan, and R. Rahman, *Focused Domain Contextual AI Chatbot Framework for Resource Poor Languages*, Journal of Information, 2019, pp. 248-269.

[12] D. François, *Methodology and Standards for Data Analysis with Machine Learning Tools*, ESANN 2008, 16th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2008.

[13] O. Thi Tran and T. Chi Luong, *Understanding what the Users Say in Chatbots: A Case Study for the Vietnamese Language*, VNU International School, Vietnam National University, Hanoi, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam, FPT Technology Research Institute, 82 Duy Tan, Cau Giay, Hanoi, Vietnam, 2019.