Universidade de Aveiro
2021

Rui Manuel Almeida e Silva

Sentiment Analysis for Improved Interaction with Conversational Agents

Análise de Sentimentos para Interação Melhorada com Agentes Conversacionais

Rui Manuel Almeida e Silva

Sentiment Analysis for Improved Interaction with Conversational Agents

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Joaquim da Silva Teixeira, Professor Associado com Agregação do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro e do Doutor Samuel de Sousa Silva, Professor Auxiliar do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Doutora Pétia Georgieva Georgieva** Professora Associada da Universidade de Aveiro
vogais / examiners committee	**Doutora Liliana da Silva Ferreira** Professora Catedrático Convidada da Faculdade de Engenharia da Universidade do Porto
	Doutor António Joaquim da Silva Teixeira Professor Associado com Agregação da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

À minha família pelo apoio incondicional que me prestaram todo este tempo.

À minha namorada por todo o apoio, carinho e força que me deu neste percurso

Aos meus amigos por todos os momentos de descontração que proporcionaram.

À Universidade de Aveiro e aos meus orientadores António Teixeira, Samuel Silva e Luís Nóbrega pelos preciosos conhecimentos que me transmitiram e orientação ao longo dos últimos meses.

Palavras Chave

Agentes de Diálogo, Sistemas de Diálogo orientados a Tarefas, Chatbots, Análise de Sentimento, Classificação de Texto, Contact Centers, Centro de chamadas, Apoio ao Cliente, Reconhecimento automático de fala, Processamento de Linguagem Natural, Machine Learning

Resumo

O uso de agentes de diálogo para tarefas comuns está a aumentar. O uso de chatbots também é popular, uma indicação da disposição dos humanos em participar numa conversa com um "robô". Apesar de seu apelo, as interações com agentes conversacionais são limitadas e muitas vezes causam reações negativas em humanos. A análise de sentimento é um processo que tenta encontrar expressões de emoções humanas ou sentimentos gerais de fontes como fala, texto, ou linguagem corporal. O objetivo do trabalho apresentado é fornecer ferramentas que possibilitem integrar informações sobre o estado emocional do utilizador de um agente conversacional para melhorar a interação, reduzir reações negativas e melhorar a experiência de utilização. Este trabalho, em colaboração com a GoContact, empresa que oferece uma solução de Contact Center como serviço cloud, com automação baseada em IA, contemplou o desenvolvimento de uma versão inicial de um módulo de análise de sentimento alinhado com requisitos derivados de cenários baseados em experiências da GoContact. Esta primeira iteração processa as sequências de palavras da fala para texto e é baseada no LinguaKit e no MeaningCloud. O trabalho também levou ao desenvolvimento de um bot rudimentar, feito para recriar as condições descritas nos cenários e obter um conjunto de dados (dataset), essencial para a avaliação do módulo e futura evolução (por exemplo, integração de análise de voz). O módulo foi avaliado com o dataset adquirido, fornecendo informações úteis sobre o potencial do LinguaKit e do MeaningCloud para a análise de sentimento de interações com agentes conversacionais em português.

Keywords

Dialogue Agents, Task-oriented Dialogue Systems, Chatbots, Sentiment Analysis, Text Classification, Contact Centers, Call Centers, Customer Support, Automatic Speech Recognition, Natural Language Processing, Machine Learning

Abstract

The use of dialogue agents for common tasks is increasing. The use of chatbots is also popular, an indication of humans' willingness to partake in a conversation with a "robot". Despite their appealingness, interactions with conversational agents are limited and many times cause negative reactions from humans.

Sentiment analysis is a process that tries to find expressions of human emotions and feelings or general sentiments from sources such as speech, text, or body language. The objective for the work presented is to provide tools to make it possible to integrate information regarding the emotional state of the user of a conversational agent to improve interaction, reduce negative reactions and improve user experience.

This work, in collaboration with GoContact, a company that delivers a cloud-based Contact Center as a service solution, with automation powered by AI, contemplated the development of an initial version of a sentiment analysis module aligned with requirements derived from scenarios based on GoContact experience. This first iteration processes the speech-to-text word sequences and is based on LinguaKit and MeaningCloud. The work also led to the development of a rudimentary bot, made to recreate the conditions described in the scenarios and to gather a dataset, essential for module evaluation and future evolution of the module (e.g. integration of speech analysis). The module was evaluated with the acquired dataset, providing useful information regarding the potential of LinguaKit and MeaningCloud for the sentiment analysis of interactions with conversational agents in Portuguese.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence.

API Application Programming Interface.

ASR Automatic Speech Recognition.

CNN Convolutional Neural Network.

EN English.

ES Spanish.

IP Internet Protocol.

JSON Javascript Object Notation.

K-NN K-Nearest Neighbors algorithm.

ME Maximum Entropy.

MFCC Mel-frequency Cepstral Coefficients.

NB Naive Bayes.

NLG Natural Language Generation.

NLP Natural Language Processing.

NLU Natural Language Understanding.

PT Portuguese.

SDK Software Development Kit.

SIP Session Initiation Protocol.

SP Speech.

SVM Support Vector Machines.

TXT Text.

UAR Unweighted Average Recall.

Chapter 1

Introduction

1.1 Motivation

Communication through dialogue is the most natural mean of communication for us, humans. As so, there is a certain natural appeal and childlike curiosity to interact with inanimate objects the way we interact with our fellows, making this object more "human-like" in the process. With our predisposition to this form of communication, the interaction with computational systems using voice dialogue can be a more intuitive way to, for example, issue a command to a certain system in order to get a certain task done. In this regard, and profiting from strong advances in recent years, the obvious example concerns talking to bots, where you can have a conversation with the machine, an experience that more and more people know. However, the actual experience can be lacking, since you are not actually talking to a person.

One of the particularities of speech communication is providing a wide range of information that goes beyond the overall message, expressed, at different levels, in the choice of words (e.g., saying something by choosing words that emphasize dissatisfaction), intonation (e.g., expressing irony), and the acoustic properties of the voice, hinting, for instance, on our emotional state. In this regard, the way we communicate with machines using voice dialogue could benefit from the ability to retrieve information about a person's emotion from the words used or through the sound of their voices, providing a way towards humanizing a bot.

Having sentiment analysis present in dialogue systems can boost user satisfaction with the experience. Information on the emotional state of the user can help the system better tune its responses, ideally making it appear as if it has some form of empathy for the user.

1.2 Work Framework

This work is made in collaboration with the company GoTelecom, a company specialized in integrated IPBX Contact Center solutions, Contact Center and Communications

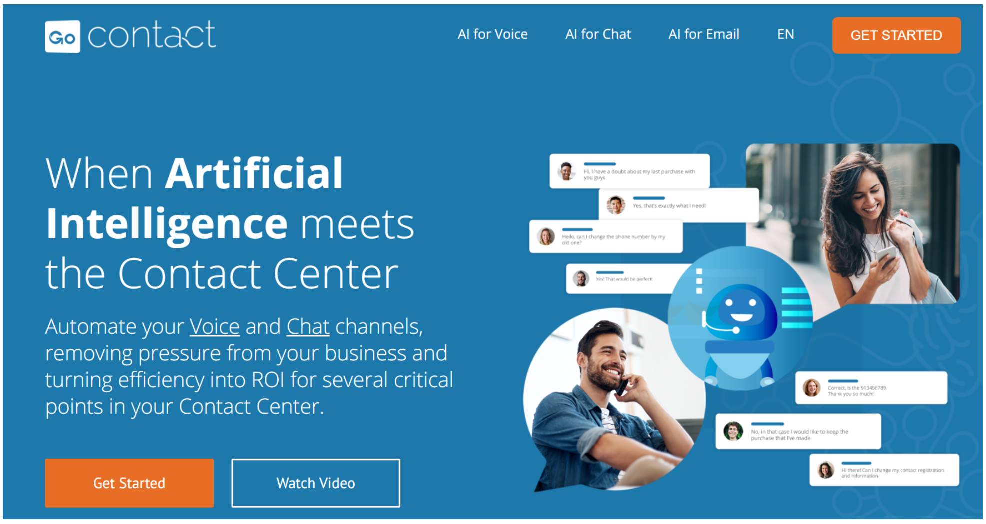

Figure 1.1: GoContact platform webpage

Software. Its GoContact[1] platform (Fig. 1.1) provides a cloud-based Contact Center service solution, with multi-channel automation powered by AI, chatbots and voice assistants. As so, the work is mainly focused on the possibilities of applications of Sentiment Analysis in bots in the context of Contact Center and Customer support.

In all instances in this thesis where the company is referred to, it is treated as GoContact.

1.3 Challenges

The use of dialogue systems such as chatbots has been on the rise through the past decade, and this trend is expected to continue [1]. It is estimated that 40% of users worldwide actually prefer to interact with these types of systems [2], but 46% of the internet users in the USA still think that live person support is better [1]. This can be due to numerous factors, such as unhelpful responses or a biased idea that chatbots can't help them successfully.

Bots can also sometimes feel "disconnected" from its user, after all, it is still a machine, and this often prompts the user to speak in a more "machine-like" manner, which causes the communication to become unnatural and uncanny. This is especially relevant in customer support bots, where the user experience with the service is very important, and to have a displeased client is to be avoided. Having sentiment analysis being performed in the caller can give an idea of the emotional state of the person, and this can be used so the dialogue agent can better suit its responses to the client at hand, or, in extreme cases, know that it would be better to forward the call to a human operator. This way customer satisfaction with the service is improved.

[1] `https://www.gocontact.ai/`

Still in the realm of customer support, sentiment analysis can provide a useful metric to assess operator performance, indicating possible patterns of satisfaction (or dissatisfaction) associated with a given operator.

1.4 Objectives(s)

This work aims to explore how sentiment analysis can be leveraged as a tool to inform dialog management. In this regard, we intend to contribute to defining the stages of a full pipeline to support the research on this topic and demonstrate its feasibility. The exploratory nature of the work to carry out entails a set of stages including:

- Understand the current chatbot solution provided by GoContact based on Dialogflow, and gain knowledge on the current state-of-the-art in sentiment analysis;
- Conceptualize a modular solution for sentiment analysis for European Portuguese speech that can work with any stream of audio and without losing sight of its future integration with Dialogflow;
- Develop a proof-of-concept instantiating different modules of the proposed solution;
- Perform a preliminary validation of the proposed solution considering data from a realistic scenario.

1.5 Dissertation structure

Beyond the current chapter, this document is organized into six more chapters:

- **Chapter 2** presents some base knowledge about human emotion, call centers, dialogue systems, and sentiment analysis, and some related work in sentiment analysis in the context of customer support.
- **Chapter 3** presents the requirements for the development of a sentiment analysis module, based on personas and scenarios created based on GoContact experiences.
- **Chapter 4** makes a brief description of the most important tools used to accomplish some of the work.
- **Chapter 5** gives information regarding the intended final module and the development of the conversational agent, speech-to-text module, and text-based sentiment analysis module.
- **Chapter 6** contains the description of the results obtained from the text-based module and speech-to-text with the dataset of calls. Also is made an evaluation of these two modules.

- Finally, **Chapter 7** presents final remarks about the work and discusses some ideas for further work.

Chapter 2

Background and Related Work

2.1 Emotions and Affects

It is useful to define and differentiate some terms, that sometimes are used interchangeably. Emotions are psychological phenomena characterized by being relatively brief and caused in response to an external or internal stimulus of major importance [3]. From emotions are built sentiments, which are emotional dispositions toward a subject held for longer periods of time but that can evolve [4]. A feeling is similar to an emotion, except that they are conscious phenomena and have a "label". All these are reflections of affect by a person. The affect is non-conscious, and not entirely possible of conceptualizing in language[4], it's like the most subjective core of the mind. This leaves the emotions as the nuclear means of finding expressions of affect.

Theories of emotion

There are two main kinds of emotion theories: emotion as fixed atomic units and emotion as a 2 or 3D space[3].

Fixed atomic units theories often conceptualize some basic emotions from which others are originated, like Ekman's basic emotions [5, 3], surprise, happiness, anger, fear, disgust, sadness, which he proposes are likely to be present in all cultures.

In the other type of theories of emotion, it is represented as a point in a 2 or 3D space. The most common dimensions for this representation are valence, which represents the pleasantness of the stimuli, arousal, the intensity of the stimuli, and, in case there is a third, dominance, that says the degree of control the stimuli exerts [3].

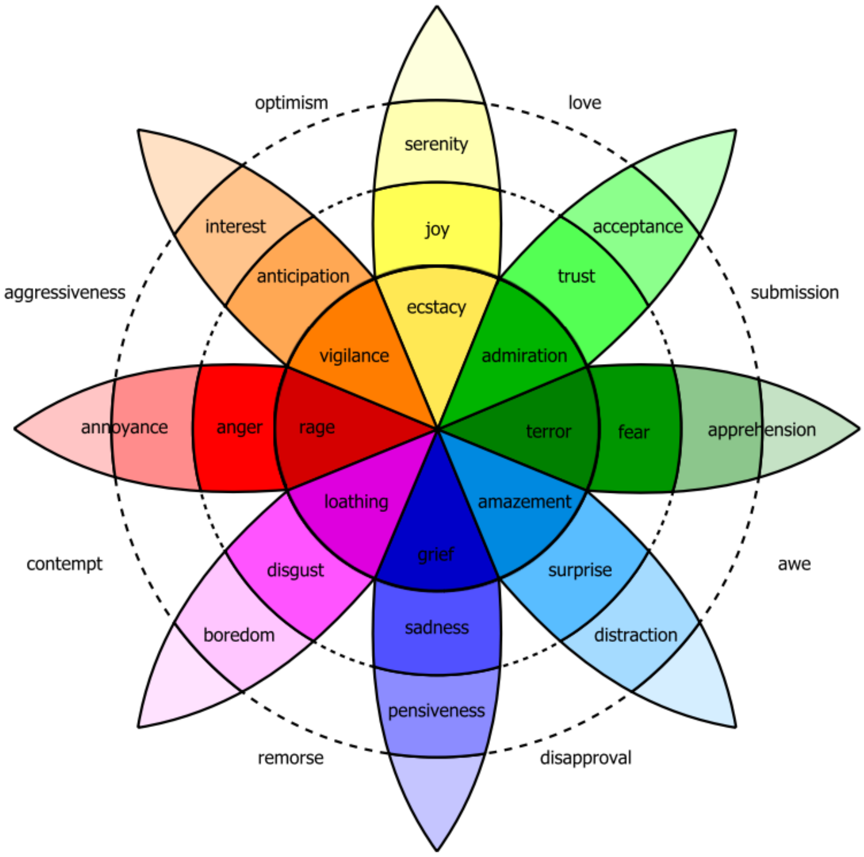

Figure 2.1: Plutchik wheel of emotion (emotion as fixed atomic units theory), taken from Jurafsky's Speech and language processing chap.24 [6]

2.2 Call Centers

A call center is generally thought of as a room with people answering phone calls [7], providing information services for a lot of different entities [8]. Sometimes they are managed by the same entity associated with the service but is very common to outsource the call center department to dedicated businesses of this area [7].

The origins of call centers can be traced back to the 1950s when the first Automatic Call Distributor systems appeared. Before that, the handling of which call goes to whom would have to be done by a telephone operator [9]. In subsequent years many businesses and services adopted the model of the call center for its clear advantages in customer support, while the telephony technology was still being improved. Nowadays, most centers use IP telephony with Computer Telephony Integration, enabling the control of calls over traditional and more modern channels. Also, they do not only deal with phone calls but with e-mail, live chat channels, and other means of communication, being the term Contact Center now used more often to describe this department [9].

The last technological breakthrough in contact centers is the insertion of AI for various tasks, like data analysis in real-time, live chatbots, and "robot agents", which had shown the ability to significantly decrease the volume of customer calls [10], and the use of sentiment analysis to estimate customer satisfaction levels [11].

2.3 Emotions explored in the call-center support systems

In table 2.1 are presented some papers dealing with emotion recognition in the context of call centers, the methods used, and their results. They are further described in a following section. Some common emotions demonstrated are:

frustration – an emotion related to anger, that results from receiving or perceiving resistance to fulfill some will or goal. This can be internal, due to personal reasons, or external, coming from a certain situation that blocks some sort of progress or gives the sensation of wasting time. People often demonstrate passive-aggressive behavior and anger when experiencing frustration.

annoyance – an emotion related to irritation, characterized by an unpleasant state of mind when someone is distracted from its conscious thinking. It can be expressed through many forms such as anger or frustration.

dissatisfaction – is the feeling of not being happy with something, meaning one is not at ease regarding a situation and is hard to conform.

anger – an intense state of mind when a person starts giving non-cooperative responses as a result of some negative stimuli. It is accompanied by strong physical reactions, such as elevated heart rate and breath, and usually, the voice becomes louder and sharper.

2.4 Dialogue Systems and Bots

Dialogue systems are computer systems designed to interact with a user via natural language [6], whether by speech or text. It was in the 1960s that those systems first appeared, with ELIZA [15], a rule-based chatbot that implemented an algorithm to mimic a method in Rogerian psychotherapy. Since then, many dialogue systems have been developed and been made commercially available.

According to Jurafsky et al. [6] and Chen et al. [16] there are essentially two types of dialog systems: Task-oriented dialog Systems, which aim to help a user finish a task, like making an order or a reservation; and Chatbots, in which the goal is to mimic a human conversation with a more open domain and give a more social approach to the conversation. Digital assistants like Siri or Google assistant fit in the first category, but there are also examples of commercially successful chatbots, and they seem to be very popular [16].

Table 2.1: Sentiment analysis methods used to detect certain emotions in call center context

Emotion/sentiment	ref	Speech/ Text	Lang.	methods/ tools	results
frustration (and Big Five)	[12] (2019)	SP	PT	linear SVM	UAR 75.4%
annoyance manifested as: anger, disappointment, fed-up, powerlessness and disagreement	[13] (2019)	SP	ES	naive-Bayes, SVM and K-NN	Maximum accuracy of 0.95 for anger using K-NN
dissatisfaction	[11] (2017)	SP+TXT	ES	Deep CNN based NLP that fuses linguistic and prosodic cues; CNN sentiment classification	F-Score 0.242 for spain spanish and 0.427 for american
positive or negative emotion	[14] (2019)	SP+TXT	EN	Lexical: backoff trigrams with Witten-Bell Smoothing; Acoustic: several methods of feature extraction and deep neural network classifier; Fusion: combine the features from both analysis, or combine the output; SVM fusion model	UAR 0.7797 for Cepstral feature extraction + backoff, SVM

Their “open-endedness” can engage the user in an almost human-like conversation, and this chatbot approach can be used to improve the experience with task-oriented agents too [17]. This way, it is possible to have a more natural social interaction while booking an air trip, for example. The experience further benefits by giving the agent some empathetic behavior [18], like if the user demonstrates signs of sadness, the system can tune its responses to better address the person.

Task-oriented dialog agents and rule-based chatbots follow a general pipeline architecture constituted by an ASR (automatic speech recognition), if to be interacted by voice, an NLU, natural language understanding unit, that retrieves the meaning of the person's message, a dialog state tracking and dialog policy module, that can appear separated but often grouped and are responsible for keeping context and decide what to do next, and an NLG (natural language generation) component that converts the decision of the policy module in natural language [16]. With the rise of deep learning techniques there have appeared more methods to develop a bot by making a neural model that maps directly the input of the user to the output of the bot [17], as well as hybrid approaches [6].

2.4.1 Automatic Speech Recognition

Automatic speech recognition (ASR), or speech-to-text, is a task of speech processing fundamental for interacting with a system using spoken words. It consists in transforming an audio waveform of speech into the sequence of words the person is saying [19]. To do this, given a recording of a speech in discrete samples, parts with no speech and noise are cut out, leaving only the phonemes, which are then sliced in equal timed windows, usually of 25 ms, and categorized according to a probabilistic method with the most probable word [20].

There are some tools available to implement ASR, like code libraries like Kaldi [21], more intended for research, or speech-to-text services included in commercial cloud computing solutions like IBM Watson [22] and Google Cloud Platform [23].

2.4.2 Task-oriented Dialogue Systems

The goal of task-oriented dialogue systems is to help and guide the user to accomplish a specific goal. One of the earliest systems trying to do so was made with the GUS architecture [24], and it still serves as the basis for most modern implementations of commercial dialogue agents [6]. This architecture is based around frames, which are a kind of data structure that can represent the various intentions a user can have using the system, but also other types of data structure. In it is a collection of slots, that are filled with data retrieved from the user, and can either be a requirement of the frame or not. A set of frames that map intentions can be called a domain ontology [6], with the domain meaning the knowledge area of the conversation. Each slot can have a restriction of what type of data is to be inserted there, like a date or a place, and this data type can have a hierarchical structure, like in a date, it has values such as year, month, day. For this case, the slot is itself represented by a frame.

The dialog control and policy in GUS is centered in filling the required slots of the frame that corresponds to the intent the user wants, and with these slots filled execute the end action, defined also in the frame [6]. When the user makes the request, the NLU component matches it to the frame corresponding to the intent and fills slots with data if already present in the request. If all required slots are filled, the system proceeds to the action, if not, it asks the user for the required information, using question templates associated with each slot.

Modern architectures make use of a more refined version of the frame-based architecture, named Dialogue-State architecture [6]. This adds complexity to all parts of GUS, mainly by leveraging machine learning for previously ruled-based tasks, like NLU. This architecture is still centered around frames but has a more sophisticated dialogue policy, based on dialogue acts. They represent the conversational interaction meant for the turn. The dialogue policy basically works by keeping a record of the last dialogue act in the conversation and having a kind of a main frame with the slots to be filled. An algorithm decides what to do next based on the dialogue state and this frame. This gives the chance for more dynamic interactions with the user that better resemble a natural conversation [6].

2.4.3 Development tools

Nowadays some tools greatly facilitate the development of dialogue systems. They can offer a functional base with which to program, or entirely cloud-based applications made for bot building, enabling developers to focus on the specifics of their bot solution instead of implementing their own architecture from scratch. Below is a list of some of these tools.

DialogFlow – Bot development platform part of Google Cloud services; it provides a web platform for easy development and integration with other services, along with the hosting [25].

Microsoft bot framework and azure bot services – Microsoft offers an SDK (Software Development Kit) with predefined functions related to the creation and functioning of a bot, for various programming languages, usable in an environment of choice. It does not offer direct hosting [26].

IBM Watson – Similar to DialogFlow, in which it offers a visual interface to build the bot from any browser and the hosting itself. It has the offer of a limited free tier [27].

RASA – An open-source program that contains a framework to build and train the bot with command-line commands and data files structure [28].

Amazon Lex – A bot development and managing platform associated with Amazon Web Services, also to be used from the browser [29].

botkit – Provides an open-source programming library and tools for local bot development [30].

Pandorabots – It offers free SDKs for various programming languages to develop the bot locally but offers a paid hosting plan through API's [31]

BotPress – Has a free and open-source downloadable program with a visual interface for the bot development, from the intent matching to the conversational flow [32]

2.5 Sentiment Analysis

Sentiment analysis is a field of study, part of NLP (Natural Language Processing), that deals with the subjectivity in human communication [33], primarily in text form but can be enhanced by acoustic or visual cues [14], known as multimodal sentiment analysis [34]. Subjectivity is understood as the emotions, opinions, attitudes, or other attributes reflected in human communication. With the rise of the internet and social networks, lots of resources became readily available with which to train the algorithms that perform sentiment analysis, increasing its interest and usage [35, 33].

Sentiment analysis can be approached as a classification job [36]; usually, it is used a binary approach to sentiment (classifying a text with positive or negative regarding the valence of the emotion) but a more advanced approach is possible, classifying it with the particular emotion that is being manifested [3]. Often, when developing a sentiment analysis application, it is necessary to choose the domain of the communication, as there are words that carry different subjective charges depending on the situation [35], like the word "high", which for "high profits" is positive but for "high infection rates" is negative.

2.5.1 Levels and Tasks

Liu et al. [33] describe three levels of sentiment analysis: sentence level, document level, and aspect level. The word level can also be considered [37, 35]. The level defines what is going to have a classification at the end of the analysis. Yadav et al. [34] considers the levels as being the major tasks of sentiment analysis.

When using the word level, certain parts-of-speech considered more relevant for sentiment analysis such as adjectives, verbs, adverbs, or sometimes nouns will have a classification based on a sentiment lexicon [35]. In the sentence level is the whole sentence that is going to have a classification, sometimes with a pre-analysis that determines if the sentence is purely objective (states a fact) [33]. If it is, then sentiment analysis makes little sense and the sentence is filtered out, but it is dependent on the situation. At the document level, the full text is classified assuming that the opinion being expressed refers to only one entity [33]. The aspect level is the more complex, as it tries to find not only the sentiment but also the target of the sentiment and what aspects of the target the sentiment refers to [35, 33].

There are also secondary tasks that are associated with sentiment analysis, such as multi-domain sentiment analysis, characterized by finding the domain in a certain text, and multimodal sentiment analysis when using other sources of emotional information such as the speech audio or video [34].

2.5.2 Methods

The methods used for sentiment analysis can be generally included into one of the three categories: Lexicon-based, Machine-Learning, and hybrid [35]. Their use generally depends on the level of the sentiment analysis, being used lexicon-based more for the word level [37], and machine-learning methods for sentence and document level [33]. In Fig. 2.1 is presented a representation of the different types of common methods.

Lexicon-based methods rely on comparing words in a document to a dictionary or corpus of words [35]. A dictionary contains a list of words with their emotional orientation (valence) and corresponding synonyms and antonyms [33]. They are used in a general domain, as its valence classification is often not domain-specific and does not take word context into account. For that, a corpus is used. These are like the dictionaries, but can be more domain aware, given a domain corpus, and take into account certain grammatical rules to infer the emotional value of a word [33].

The most used methods are from machine-learning, and these can be divided into supervised and unsupervised learning [35]. As said above, sentiment analysis is analog to text classification, so several common algorithms already used for this purpose can be applied [33], such as Naive Bayes, Maximum Entropy or Support Vector Machine [35]. These methods are considered to be supervised learning, as they use reference inputs with their classification already done, and know what features are relevant in order to be trained.

Naive-Bayes algorithm uses the concept of bag-of-words, which considers the order of the words in a document to be irrelevant [36]. Instead, it only cares about the number of times a word appears in the text and uses this information to find the most probable class to which the document belongs, beyond the actual words. Using as classes the positive or negative emotional valence it is possible to train a naive Bayes classifier with a set of documents with these labels, that then returns, given a new document, its sentiment classification [36].

Unsupervised learning methods work with unlabeled data or unknown features. That is, an algorithm of this kind doesn't know when trained what features of the input are more relevant to the task it wants to perform, it has to figure them out. This is unlike the Naive-Bayes algorithm, which knows it has to pay attention to the words and their frequency.

Recently there have been more methods using deep learning for sentiment analysis [34], that have come with the advantages of processing more data and extracting more features, which have improved its performance relative to other methods [34]. These methods are always unsupervised in relation to the features, but they know the label, or score, of a training set of text. This is useful because this way it is not necessary to define the features manually, although being necessary much more data to train them. Some common methods based on Neural Networks include Convolutional networks, Recursive networks, Recurrent networks, and Deep Belief networks, each better performing in some level of sentiment analysis than others [34]. For the sentence level, it is often used CNN and Recurrent neural networks, particularly the Long Short-term Memory variant [34].

2.5.3 Tools for Text based Sentiment Analysis

Table 2.2 summarizes relevant tools available to perform sentiment analysis from text and related utilities. It has information on whether or not it supports Portuguese, custom training, how can it be integrated into a program, and if it has a cost to use. Among

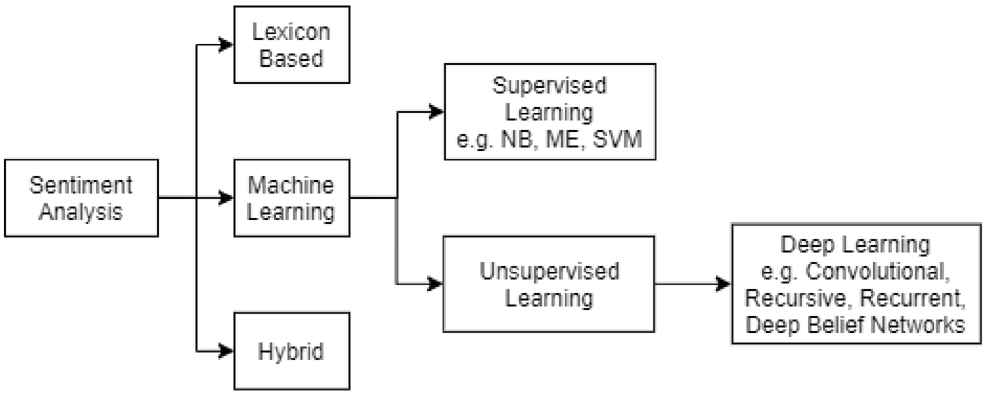

Figure 2.2: A summary of the types of methods used to perform Sentiment Analysis

the list, there are lexicons that offer not only a dictionary of words but also sentences and expressions with corresponding emotional valence [35], stand-alone sentiment analysis systems, commercial services, and python libraries.

Lexicons can be useful for training a classifier, yet is necessary to verify that the required domain is compatible with the lexicon. Systems and services already offer a functional solution, which in many cases can be enough, depending on the application, and some offer a variable degree of analysis customization. However, for greater control over the performance of the system, the development of a new algorithm can be considered using, for example, Keras, which facilitates the use of deep learning techniques.

Some of the tools present in the table were listed by Pereira [57] in a survey of Sentiment Analysis approaches for Portuguese.

2.6 Recent works on analysis of contact centers calls

Cabarrão et al. [12], in 2019, performed sentiment analysis on customer service calls using characteristics from the voice, and searched for how the level of customer satisfaction is related to the results of the analysis, performed both on customers and operators. They used as data a call-center corpus from a non-technical support service in Portugal. The system classified into frustrated vs neutral and was made a manual analysis of personality types according to the big-five model. The model used was linear SVM, using knowledge-based features, and obtained UAR (unweighted average recall) value of 75,4%, having a recall of 81% for unsatisfied customers vs 68% for satisfied, result improved using the eGeMAPS features to 75%. They verified that pitch range and dynamics are very relevant in satisfaction prediction, along with energy dynamics and range. Regarding the manual analysis, they concluded that the personality type has little impact on the degree of customer satisfaction and that the operator's performance is relevant.

Also in 2019, Irastorza et al. [13] tried to detect annoyance rates on customer calls in response to a research question by a Spanish call center. It wanted to evaluate a procedure to deal with annoyed customers. In order to do it they manually classified a set of calls

Table 2.2: Tools for Text-based Sentiment Analysis

Tool	Type	Ref	PT	Cost	Program Interface	Trainable
Sentilex PT	Lexicon	[38]	YES	NO	NA	NA
Linguakit	System	[39]	YES	NO	Perl	NA
Sentistrenght (free)	System	[40]	YES	NO	NO	NO
Sentistrenght	System	[40]	YES	YES	Java	YES
OpLexicon	Lexicon	[41]	YES	NO	NA	NA
LIWC (academic)	System	[42]	YES	YES	NO	YES
LIWC (comercial) (Receptiviti)	System	[42]	YES	YES	API	YES
Onto.PT	Lexicon	[43]	YES	NO	NA	NA
Reli-Lex	Lexicon	[44]	YES	NO	NA	NA
ReLi	Dataset	[45]	YES	NO	NA	NA
OntoLP	Ontology Set	[46]	YES	NO	NA	NA
SenticNet (BabelSenticNet)	Ontology	[47]	YES	NO	API	NA
NILC Word Embeddings Repo.	Pre-trained word emb.	[48]	YES	NO	GENSIM lib	NA
Monkeylearn	Service	[49]	YES	YES	API	YES
Semantria API	Service	[50]	YES	YES	API	Limited
Salience6 library	Service	[51]	YES	YES	Java, PHP, Python, .NET/C#, C/C++	YES
MeaningCloud	Service	[52]	YES	NO (Limited)	API	YES
Rosette	Service	[53]	YES	YES	REST API	YES
Repustate	Service	[54]	YES	YES	API	YES
Vader	Python Library	[55]	NO	NO	Python	YES
Keras	Python Library	[56]	NA	NO	Python	YES

with various labels of forms of annoyance: disappointed, angry 1, angry 2, extremely angry, fed-up, impotent, annoyed in disagreement; then organized in 5 levels of intensity for each category. This data was used to train and test 3 algorithms based on Naive Bayes, SVM, and K-NN (k-Nearest Neighbour) to classify each form of annoyance using speech features. In the end, they measured the accuracy and f-measure of the classifiers, obtaining the best accuracy of 0.95 for classifying anger with the K-NN algorithm using MFCC (12 Mel-Frequency Cepstral Coefficients) and a set of features related to intensity.

Luque et al. [11], in 2017, tried to classify customer satisfaction in high and low using both textual cues and vocal cues separately and together. The sample conversations came from a Spanish call center. They used two methods for the classification by text: using a bag-of-words approach, normalized in order to the frequency of each word, and with a CNN architecture. For the speech analysis, they separated the audio by frames of 500ms in which to extract the several features, used in conjunction with some conversational cues, that relate more to the interaction between the speakers than on a single speaker. Separately the analysis made using textual features outperformed the ones using voice features, but doing analysis using a fusion of both characteristics is more efficient than using each one separately, idea further supported by [14]

Chapter 3

Scenarios and Requirements

This chapter presents the Personas and Scenarios developed to serve as the starting point to define the Requirements for the development of the system, present at the end of the chapter. This way the requirements are found naturally, according to the circumstances that arise in the developed example scenarios.

3.1 Personas and Scenarios

3.1.1 Personas

The personas here presented are based on provided GoContact information and a film from GoContact's youtube [58]. They are representative of the main intervening entities in a GoContact Contact Center.

Dave [David] (Primary Persona) – David, 18 years old, is a customer of NiceClothes. He buys clothes frequently on the online store and calls typically customer support to request returns. He is emotionally sensitive in the way that he does not accept any type of failure. In a call, he can quickly change from a good mood to an angry state if the agent does not treat the issues quickly.

Operator/agent [Sandra] (Primary Persona n. 2) – Sandra is one of the 50 agents of the Contact Center. She is 30 years old and works in a Contact Center for a year. It has experience in all Contact's Center channels, including voice (inbound and outbound), tickets, and web-chat. She speaks with different customers, with different temperaments, daily. Therefore, she typically faces difficult situations that she must carefully handle to ensure client satisfaction.

Supervisor [Paula] (Secundary Persona) – Paula is a supervisor at CallMe Contact Center. She is 42 years old and works in the area for 10 years, managing a team of 50 agents that give customer support to a NiceClothes brand. Her job includes monitoring and ensuring that the established SLAs are accomplished, maximizing the number of calls answered, minimizing the number of calls not answered, ensure customer satisfaction, just to cite the more important tasks.

GoContact Bot (Persona 3) – To offer a certain level of autonomy, GoContact offers a solution that allows the creation of voice bots capable of replacing the agent for simple and recurrent tasks. However, contrary to the agent's sentiment capabilities, the bot is not capable of understanding the customer's sentiments or mood. Therefore he is not capable of adjusting the conversation according to the customer's spirit.

3.1.2 Initial Scenarios

In this section, we present three small scenarios defined at the beginning of the work. These are intended to give a general overview of how a sentiment analysis module could be useful in the context of customer support.

3.1.2.1 Scenario 1 - Call with a human operator

- Dave calls to customer service and is answered by a human operator.
- Dave states a problem he wants to see sorted out or requests some information.
- Operator complies with Dave's questions or requests.
- The system performs sentiment analysis on Dave's speech.
- After the call is done the results of the sentiment analysis are stored.
- The company can access the results to assess Dave's satisfaction with the customer service.

3.1.2.2 Scenario 2 - Complex call with GoContact Bot

- Dave calls to customer service and is answered by a bot.
- Dave states a problem he wants to see sorted out or requests some information.
- The bot tries to help Dave, but either it does not understand Dave or Dave's intent reveals itself too complex.
- Dave starts to become annoyed with the service provided.
- The system detects this through sentiment analysis and informs the bot to ask Dave if he wants to be transferred to a human operator.
- Dave confirms and is transferred to a human operator.

3.1.2.3 Scenario 3 - Successful call with GoContact Bot

- Dave calls to customer service and is answered by a bot.
- Dave states a problem he wants to see sorted out or requests some information.
- The bot tries to help Dave.
- The bot is capable of answering Dave's needs but the process is annoying and slow.
- The system performs sentiment analysis on Dave's speech and detects a certain level of dissatisfaction.
- The bot uses the results of the sentiment analysis to better suit Dave's needs and improve his satisfaction with the service, for instance by pointing out moments in the process when Dave gets impatient.

3.1.3 More Detailed Scenarios

After acquiring more in-depth information of the GoContact System and related work, two evolved scenarios were defined, which are presented next. Derived requirements are integrated with the scenarios inside square brackets.

The first one is related to a common situation reported by GoContact, the bot having difficulty understanding the user, and as a consequence, negative reactions occur.

The second is related to design problems of the bot conversational flux, creating difficulties for the users and, in consequence, negative reactions.

3.1.3.1 Scenario 4 - Bot with difficulties

Dave ordered a blazer from NiceClothes and upon arrival noticed he didn't like the texture of the fabric. With the intent of returning the product, he calls customer service and establishes a dialog with the GoContact bot:

Bot: Good afternoon, how can I help you?

Dave: I ordered a blazer from NiceClothes that I want to return.

Bot: I see. For the return to be accepted the blazer should never have been used and have its tag. Is this correct?

Dave: Yes. Never used and with original tag.

Bot: Please confirm never used.

Dave, a bit annoyed: Yes. I never used it.

Communication Monitorization Module (CMM) (detects Dave's negative emotion in voice[**P0 – Detect negative emotion in voice**] [**P0 – Detect negative emotion in transcription**] and informs the Bot)

Bot: Thanks for your collaboration and I'm sorry if and failed something in your answers. Could you please provide your client ID?

Dave: My client ID is A1234/1995.

Bot: Thanks. I see... I'm talking with Mr. Scott Rupert a valuable client since 1995.

Dave, getting angry: No! My name is Dave, not Scott.

Communication Monitorization Module: (detects Dave's starting to get angry [**P1 – Detect Angriness**] and sends information to Bot)

Bot: I'm so sorry! As it seems that I'm not on my best days I'll pass your call to one of my human colleagues. Please don't disconnect. Continuation of a good day...

3.1.3.2 Scenario 5 - Flux problems

Dave ordered a blazer from NiceClothes and upon arrival noticed he didn't like the texture of the fabric. With the intent of returning the product, he called customer service and was answered by a bot;

Bot: Good afternoon, how can I help you?

Dave: I ordered a blazer from NiceClothes that I want to return.

Bot: I see. For the return to be accepted the blazer should never have been used and have its tag. Is this correct?

Dave: Yes, I know I can return it.

Bot: Do the blazer has the tag?.

Dave: (...) yes.

Communication Monitorization Module: (Detects that Dave is becoming impatient[**P2 – Detect impatience**])

Bot: And it was never worn, correct?

Dave: Yes, I already said that!

Communication Monitorization Module: (Recognizes annoyance in the expression "I already said that" [**P0 – Must be capable of infer emotions from a sequence of words**] and in the tone of his voice [**P1 – process speech signal**] [**P1 – use tone of voice information**] and sends information to the bot);

Bot: Thank you for your time, you want to return to the store or by mail?

Dave: I bought online so I want to return by mail! And when do I get a refund?

Communication Monitorization Module: (Detects that Dave is growing [**P1 – track time evolution of user reactions**] more angry/annoyed [**P1 – detect angriness, annoyance**] and trying to skip and hop between topics in the conversation, so decides to tell the bot to transfer the call to a human operator);

Bot: I'm so sorry! As it seems that I'm not on my best days I'll pass your call to one of my human colleagues. Please don't disconnect. Continuation of a good day...

3.2 Requirements

From the scenarios and company practices were derived the set of requirements presented in Table 3.1. In the table, three levels of priority were used, with P0 denoting the higher priority. Requirements with P0 should be contemplated in first developments; with P1 should be part of first developments if possible; P2 can be addressed in later development stages.

Table 3.1: List of requirements for development of the module; P0 = to be contemplated in first developments; P1 = part of first developments if possible; P2 = can be addressed in later stages of development.

Pr.	Requirement
P0	Development and testing of a dialog agent for data gathering
P0	Perform Speech to Text conversion
P0	Detect negative emotion in transcriptions through valence
P0	Detect negative emotion in voice through valence
P1	Process speech to extract features
P1	Fusion of sentiment outcomes from transcriptions and voice
P1	Detect angriness in voice
P1	Detect angriness in transcriptions
P1	"real-time" monitoring (of mood, sentiment, emotion..), meaning real-time at turn level
P1	Have a "module" decoupled and easy to use (implies API etc)
P1	Track evolution of user emotions
P2	Detect changes in the emotion of users (annoyed, impatient...)
P2	Integration of extracted new information with the dialog agent

Chapter 4

Tools

In this chapter are presented the tools chosen for the various parts of development. DialogFlow is used by GoContact in their bot infrastructure so it was a natural choice. LinguaKit and MeaningCloud are sentiment analysis tools that were chosen to provide a starting analysis as a starting point to eventually develop a new algorithm.

4.1 Tools for Bots Development

Based on the GoContact system, Google DialogFlow was adopted. This section provides relevant information regarding DialogFlow.

4.1.1 DialogFlow

DialogFlow is a tool developed by Google that allows the easy creation and deployment of Dialog Agents (Task-oriented Dialog Systems) with a simple web interface, NLP, and ASR built-in, enabling the developer to focus on the features of its particular agent [59].

It works by defining intents, comparable with frames, that represent the desire of the user when interacting with the agent (for example, placing an order or checking status), that can be populated by entities (the slots to be filled with the information retrieved). For intent recognition is defined a set of training phrases, ideally representing a large number of ways the user could utter the request, for posterior training of the agent with machine learning. Response phrases can be defined to provide feedback to the user. The entities can be given as a list of words that the agent can recognize in the input phrase, or signaled in the training phrases. DialogFlow already possesses system entities like time, location, several measurement units, color, to name a few. Entities can also be complex, composed of more than one attribute (for example, a blue shirt is defined by the fact that it is a shirt and by its color). To keep the flow of the conversation between intents contexts are used, which provide the grounding between the agent and the user. This way the system has a way to know in which stage of the conversation it is going.

4.2 Tools for Sentiment Analysis

From the list of tools in table 2.2 were chosen two to be used in the text-based module. They were chosen according to the following criteria:

- Support for sentiment analysis for the Portuguese language;
- Can provide an analysis at the sentence level or turn level;
- Free to use, or at least with the possibility to, within reasonable limitations;
- Configurable, and, as much as possible, programmable;
- Usable in a Python program or possible to integrate;
- Possibility for adding custom models for the sentiment analysis (optional).

LinguaKit and MeaningCloud were the discovered tools that mostly meet the requirements, except the last one, with Linguakit not being trainable, but offering other tools useful in NLP.

4.2.1 LinguaKit

LinguaKit is a collection of multilingual text analysis, extraction, and annotation tools [39]. It is usable with text in Portuguese, Spanish, English, and Galician. Its architecture is made with different interdependent modules, that together can accomplish several NLP tasks, like part-of-speech tagging, tokenization, lemmatization to name a few, and the one used in this work, sentiment analysis.

Its sentiment analysis algorithm provides a result of the polarity of the text ("P" for positive, "N" for negative, or "NONE" for absence of emotion) and is based on the Naive Bayes Method [39], performed at the sentence level.

It is available to try as an online tool [1] but also as an open-source program, and this is the way it is used. Despite it not being written in Python nor having a Python API, it has a command-line tool that is called as a subprocess inside a Python program, being in this way integrable with Python.

4.2.2 MeaningCloud

MeaningCloud is a multilingual text analytics service that offers a set of APIs for different tasks related to text classification in different categories and some common NLP tasks. These are also available in "non-code" integrations such as excel addons [52].

Its sentiment analysis algorithm performs it on the sentence level or document level, according to the text given, and returns its polarity, like LinguaKit, but with two degrees of intensity for both positive and negative classification, with the two extra "P+" and "N+".

[1] `https://www.linguakit.com/pt/analise-completa`

It uses a combination of statistical methods and rule-based classification to obtain the final result. The analysis can be customized with custom sentiment models, that are useful to change the results to be more suitable to the conversation domain. It also provides an aspect level analysis, with a list of detected entities and their aspects and corresponding polarity, being this customizable too. As a final detail, the sentiment analysis comes with a field that says if the person is communicating subjectively or objectively and if the text contains irony or not [60].

It offers a free plan with a free account, which gives access to the service for a maximum of 20000 requests per month [52], which is enough for the application given in this work.

Chapter 5

Communication Monitorization System

5.1 Overall System Architecture

In Fig. 5.1 is presented a concept for the overall architecture of a Communication Monitorization Module like the one described in the scenarios in Chapter 3. Despite the work developed only being a proof-of-concept and not a complete implementation, it was still developed trying to approximate this.

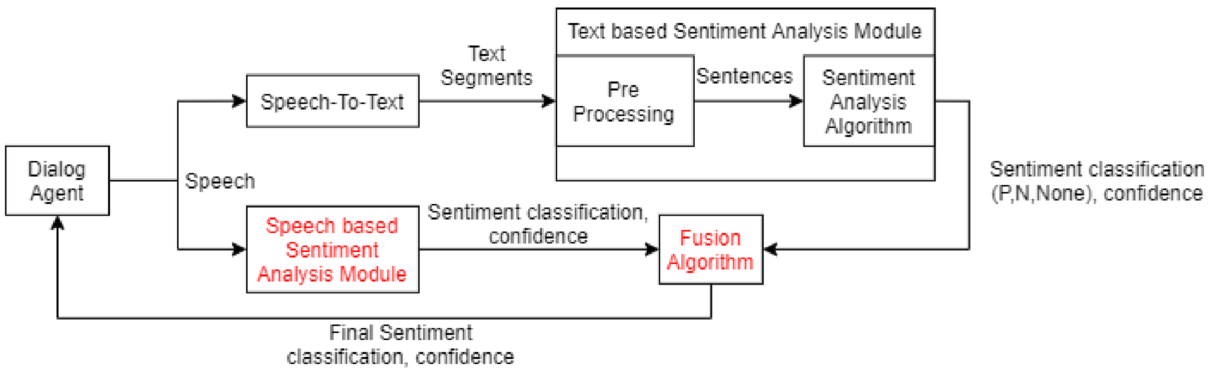

Figure 5.1: Concept of the overall module architecture. Modules with text in red are not contemplated in the developments

5.2 Dialog Agent

In order to gather conversational data to be used for the training of the sentiment analysis algorithm, a simple task-oriented dialog system had to be done to simulate simplified real-life cases of customer-bot interactions.

The developed bot implements an agent for a clothes store (named NiceClothes), for which the client can make some requests. The different requests that the bot can understand are defined by the chosen intents implemented in DialogFlow. This tool is used only to handle the matching of an intent to a sentence (retrieve the meaning), to give the conversion a flow it is necessary to follow a flux of interactions (dialog state tracking and policy). This was implemented by GoContact using in-house tools, that then use a DialogFlow block for the intent matching. The design of the conversational flow was made having in mind that the experience with the bot should be a little clunky to trigger some negative reactions from the user.

The bot had several iterations of the conversational flow, being the final flux presented in Fig. 5.2.

5.2.1 Functionalities

When calling to Nice Clothes, one is greeted with: "Good afternoon, how can I help you?". Here the agent is ready to take the customer's request.

The conversational agent recognizes 3 basic customer intents: make a purchase, make a return, or request certain information. If it detects the intention to make a purchase, the agent asks which item is to be ordered, even if it had already been in the original request. Given the item, the agent asks for the item's color. All items are available in all colors. Next, the size of the garment is asked. The sizes that the agent recognizes are 39 to 42 and S, M, and L, and it makes no distinction between what kind of size the part in question should be. Finally, the agent asks for confirmation of the order, and, receiving it, informs the customer that he will receive an email with the purchase information.

When the intent of making a return is detected, the agent asks for the order number. This number is five digits long and can be any number. Then the agent asks if the customer wants to return the item at a store or wishes for it to be collected at home. If the customer wants to make the return at a store, the agent only informs that he will receive an email with the order information and to present the email in the store. If the home collection option is detected, the agent asks for the address. This can be any fictitious address, just street name, or include house number and town. Once given, the agent informs the time when the courier will be at the address and asks for confirmation, and, receiving it, informs that the customer will receive an email with the information.

If the customer says something like "I want to ask a question" or "I want to ask for information", the agent responds by asking what they want to know. The information it can provide is:

- The list of "sold" products
- Available colors for the products
- How do you place an order

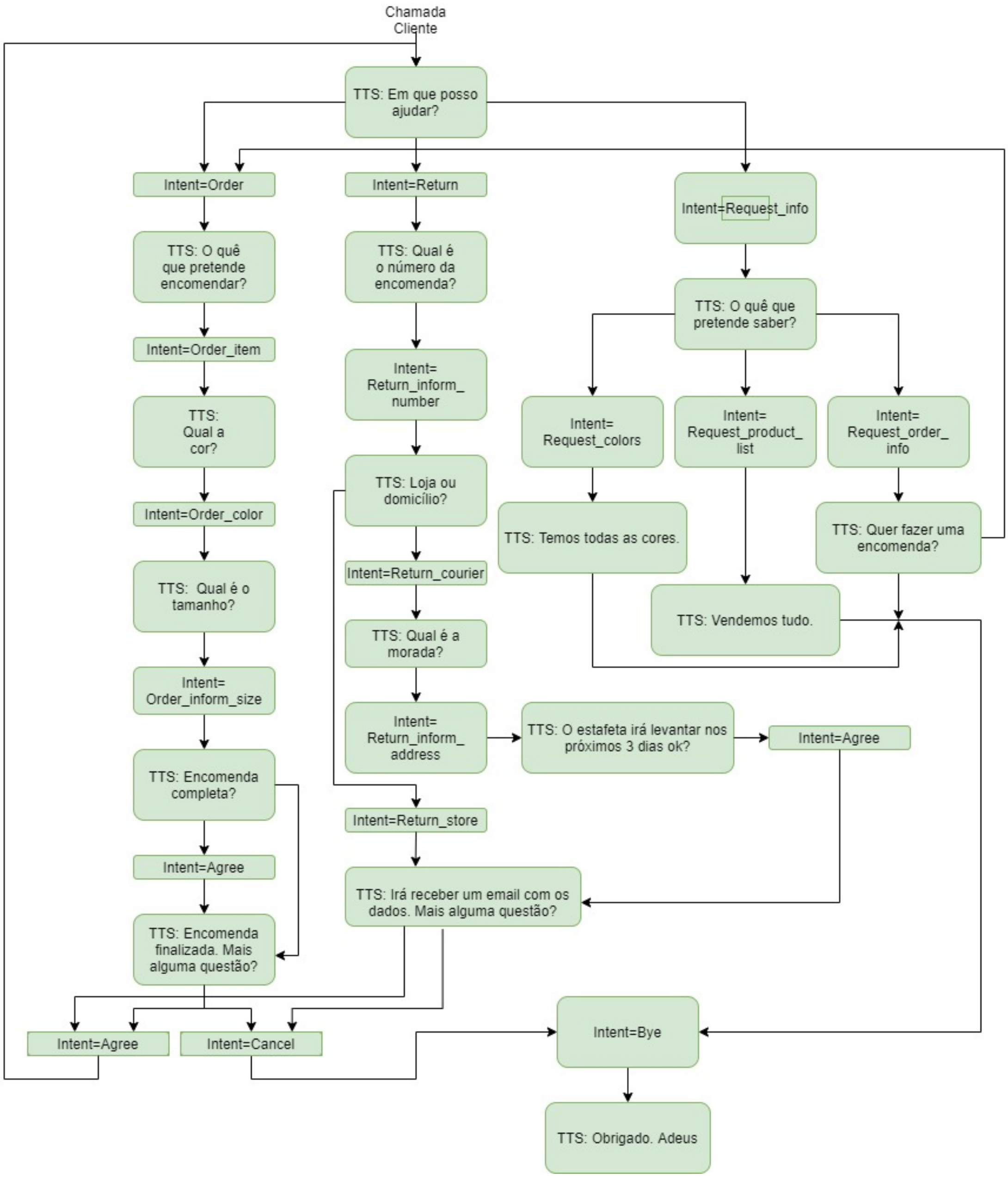

Figure 5.2: Final version of the dialogue agent conversational flow chart

After answering the question, the agent asks if they can help with anything else, which means they are ready to respond to a request from the main three intents (order, return, or information) again.

5.2.2 Development Method

The dialog agent is implemented with a combination of GoContact in-house tools as dialog state tracking and policy, and DialogFlow as the natural language unit. As so, DialogFlow is responsible for matching the intent to the sentence, with the intent being then used to inform the conversational flux in GoContact. To do this, it is necessary to create a list of intents with its corresponding characteristic phrases, used by DialogFlow for training its algorithm. It was made a set of exemplary dialogues between a user and the dialog agent to iteratively find the required intents for the system, in the form of tables like in table 5.1.

From this process resulted the following list of intents:

order – When there is the intention to buy something

order:item – The user informs the item he/she wants to order

order:color – The user informs the color desired for the item

order:inform_size – Informs the size for the item

return – The user wants to return an order

return:inform_number – The user is giving the number of the order he/she wants to return

return:courier – The user wishes to have a courier pick up the return item

return:inform_address – For when the user informs an address

return:store – The user wishes to leave the return item at a store

request – There is the intention to request some information, by the user

request:product_info – Wants to know what products there are available

request:colors – The user wants to know what colors there are available

request:order_info – Wants to know how to make a purchase

agree – The user wants to comply with a given request

bye – The user wants to end the call

cancel – The user wants to cancel the operation

Table 5.1: Example of a conversation with the dialogue agent

User	NLU	Bot Action	Observ.
		Boa tarde. Em que posso ajudar. Relembro que percebo coisas como "Quero comprar um blazer" ou Quero fazer uma devolução.	
Quero fazer uma encomenda	Intent=order	O quê que pretende encomendar?	Item=?
Quero encomendar umas calças	Intent=order:item	Qual a cor que pretende	Item = calças Cor = ?
Pode ser preto	Intent=order:color	Vai então ser umas calças pretas. Qual o tamanho?	Cor=preto Tamanho=?
40	Intent= Order:inform_size	Ok. Vai custar XXX. Quer que termine a encomenda?	Tamanho=40
Ok	Intent=agree	Ok. Vai receber email para efectuar o pagamento. Em que posso ser mais útil?	
Por mim é tudo. Obrigado.	Intent=bye	Um bom dia/tarde/ etc para si. Continuamos ao seu dispor 24 horas por dia . . .	

And some that were dropped in the implementation in order to simplify the conversational flux:

return:confirm_number – For when the user is confirming the number repeated by the agent.

agree_wait – The same as agree but for a specific case of returning an item

disagree – For when the user is not satisfied because of a return being impossible

The list of intents was inserted into DialogFlow, along with a list of training phrases for each of the intents. DialogFlow uses these to match the intent with the user utterances that are similar to its training phrases. The platform also enables the definition of an entity in the training. For instance, with the sentence “I want to buy a t-shirt”, ’t-shirt’ can be defined as an entity of the type ’`item`’, and DialogFlow will train knowing that in that type of sentence comes an entity ’item’. It was defined as 3 types of entities in DialogFlow, ’item’, that contains all the items a client can order, ’size’, to specify the possible sizes, and ’client_id’, which was later removed for simplicity.

With the example dialogues and a list of intents, it was possible to design a flow chart to represent the conversational flux, for it to be translated to the GoContact design tool, responsible for handling the conversational flux. The first iteration of the flowchart can be seen in Fig. 5.3.

To this, it was made some adjustments to simplify the implementation in the GoContact platform, to speed the construction of the agent. The confirmation of the inputs given by the user was removed, along with the question for the client id number. In the return branch, was removed the waiting to check if the item can be returned, cutting also the branch of the interaction where the client would be unhappy with the impossibility of the return. The questions related to the return address were adapted to reflect the omission of a client id. Now, instead of the agent asking if the address for the return is the one associated with the client account, and if not, asking for a new address, it just asks for an address.

5.3 Speech to Text

For the first implementation of the Speech to text (or Automatic Speech Recognition - ASR) component of the module the Google Speech-to-Text service API, part of the Google Cloud Platform, was selected. A free account was set up, and a project was created. This is necessary to enable the Speech-to-text service, which gets associated with the project. Next, it was requested the generation of a key file in `JSON` format, necessary for the authentication of any request made to the API.

The program that handles the communication with the API was written in Python, with the API client library installed. The first version of the program receives through the command line the file name of the desired audio file to be transcribed. The content of this .wav file is read and stored in an object to be sent to the API. Along with the audio object is sent a configuration object, which in this case specifies the language as `pt_PT`. After sending the request, the API returns a response object, which can be dealt with as a `JSON`. The response is a list of alternatives for the transcription with its corresponding confidence value, with one indicated as the best, but, for most cases, there is only one transcription given. When there is more than one is used the best alternative. Then the program prints in the terminal the transcriptions, and their confidence value, detected in the file with the speech. This same output is stored in a text file.

An audio file to be processed ideally contains a whole raw recording of a call made to the developed bot, to simulate the scenario where the conversation was being analyzed in real-time. However, some files had to be split into less than 1-minute segments because

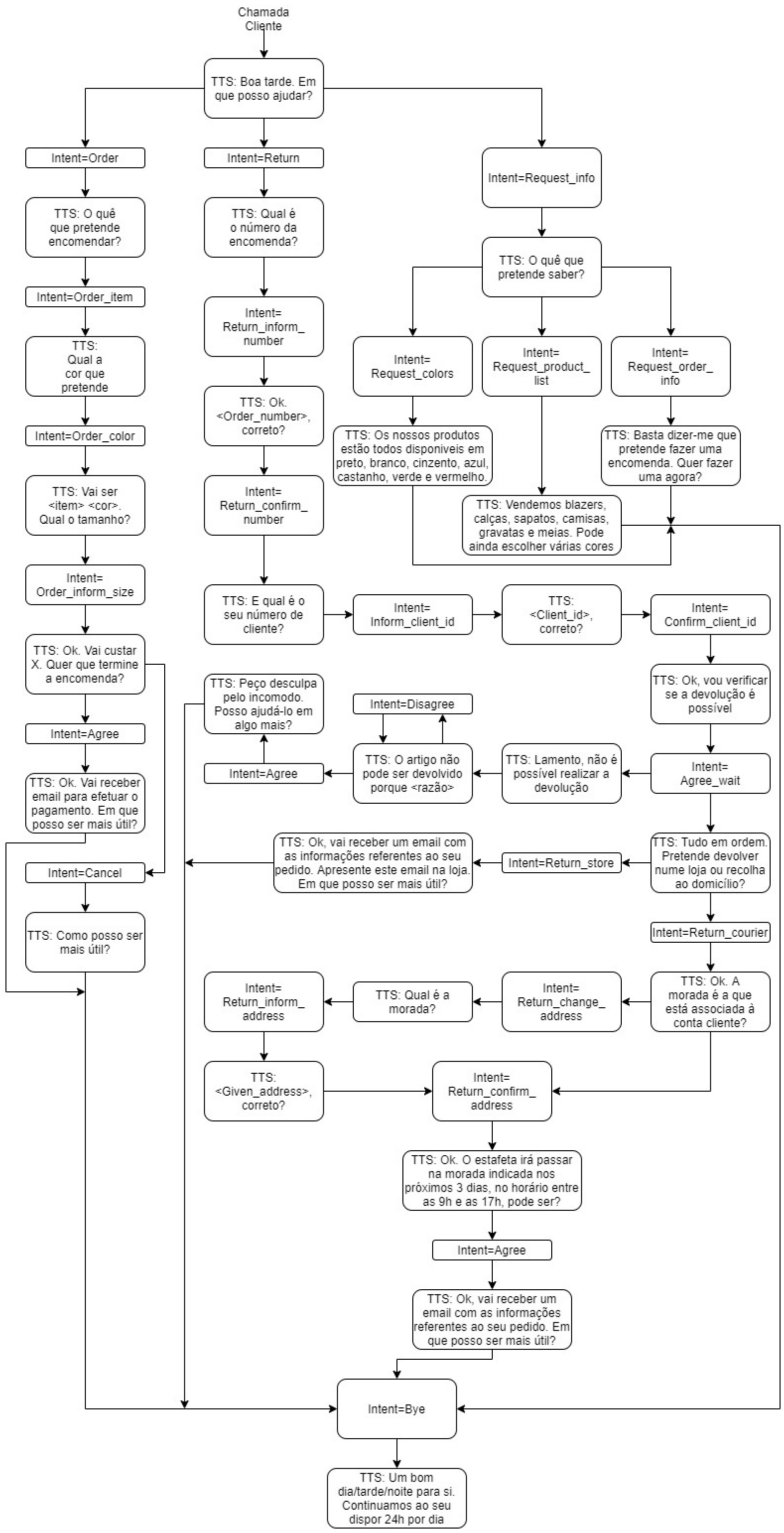

Figure 5.3: First iteration of conversational flux flow chart

the API only accepts this maximum for local files.

Despite the file being the whole conversation, the response from the Speech-to-Text service comes separated in different text segments, but with no way of knowing why it is separated the way it is. As the recording of the call has the client and the bot, sometimes one segment of text in the response can have speech from both, as we can see in the example below of a returned `JSON`.


```
results {
  alternatives {
    transcript: "Em que posso ajudar Boa tarde preciso de
    uma informação"
    confidence: 0.947372317314148
  }
}
results {
  alternatives {
    transcript: " o que pretendo saber o que é que vendem"
    confidence: 0.8938900828361511
  }
}
results {
  alternatives {
    transcript: " gémeas tudo mais alguma questão Sim em que cor"
    confidence: 0.8752858638763428
  }
}
total_billed_time {
  seconds: 30
}
```


5.4 Text-based Sentiment Analysis Module

This is the first sentiment analysis module developed, a constituent of the overall architecture depicted earlier, but also usable alone. Its purpose is to receive a text in its input and output the results of the sentiment analysis made to that text. This first version uses LinguaKit and MeaningCloud as the providers of the analysis.

5.4.1 Objectives

The main objective of this module is to provide a simple result in the form of positive, negative, or none for a given sentence, with its corresponding confidence value. It also decomposes the text in its sentences if the input text is more than one sentence, giving in the output the list of sentences with its corresponding sentiment value. It is also possible to choose which service to use for the analysis and is usable alone.

5.4.2 Submodules

This module is actually in the form of a server that receives requests through the use of an API. Through the API a user sends the input text, the language (which can only be Portuguese), and the service that is pretended to do the analysis, either LinguaKit, MeaningCloud, or both. The program in the server processes the request, passing the text through the pre-processing submodule where it is decomposed in sentences and then to Sentiment Analysis, sending back to the client a `JSON` file with the list of sentences with corresponding sentiment classification and confidence given by the desired service.

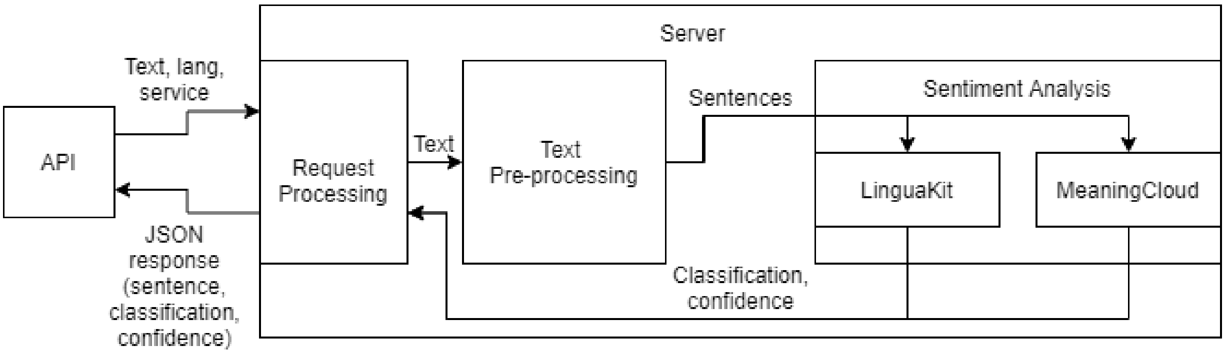

Figure 5.4: Software architecture of the Text-based Sentiment Analysis Module

5.4.3 Implementation

This module was developed using the Python programming language. As a first step were written two functions to wrap the use of the two "services" (LinguaKit and MeaningCloud), one for each, where it is given in the input the text and language and returns the raw output of each.

Communication with MeaningCloud is done through an API, where it is sent the text, the language, and an authentication key to the MeaningCloud sentiment API URL. Its response is in the form of a `JSON` with a very detailed analysis of the text.

LinguaKit sentiment analysis is run locally as a command-line program, indicating as arguments the text and language. The output of the program is captured and returned as the output of the Python function. This provides only the classification in positive, negative, or none, and the confidence, but in text form.

Then a program was written which can be run through the command line passing the text as an argument. Optionally the specific service can be chosen, defaulting to both if none is specified. The inserted text is split into sentences, if possible, using TextBlob

Python library resources [61]. At first, it showed on the terminal the raw results of LinguaKit and MeaningCloud (storing it in a .txt file as well). Later it was improved to format the results, filtering the response from MeaningCloud to only classification and confidence, and transforming LinguaKit's response to `JSON`.

After that, it was developed the solution for the functionalities to be accessible via an API. This API defines `sendToSentsys` function to implement the functionalities of the Text-based Module:

```
def sendToSentsys(txt,lang="pt",service="both"):
    payload={
        'txt': txt,
        'lang': lang,  # 2-letter code, like en es fr ...
        'service': service
    }
    response = requests.post(url, json=payload)
    return response.json()
```

A server uses the previously developed program as a library for the functions that deal directly with the providers of the service, pre-processing, and organization of the results. It handles the `POST` request done to him by the API and the choosing of which method for the analysis when specified.

Below is an example of a `JSON` returned by the server to the API.

```
"Results": [
    {
        "MeaningCloud": {
            "Sentences": [
                {
                    "Sentence Number": 1,
                    "Text": "Os atores eram bons, mas o protagonista
                    deixava a desejar.",
                    "Result": {
                        "score_tag": "N",
                        "confidence": "100"
                    }
                }
                {
                    "Sentence Number": 2,
                    "Text": "Apesar disso gostei bastante",
                    "Result": {
                        "score_tag": "P",
                        "confidence": "100"
                    }
                }
            ]
```

```
        }
    },
    {
        "Linguakit": {
            "Sentences": [
                {
                    "Sentence Number": 1,
                    "Text": "Os atores eram bons, mas o protagonista
                    deixava a desejar.",
                    "Result": {
                        "score_tag": "P",
                        "confidence": 0.999422695291359
                    }
                }
                {
                    "Sentence Number": 2,
                    "Text": "Apesar disso gostei bastante",
                    "Result": {
                        "score_tag": "P",
                        "confidence": 0.974102470965905
                    }
                }
            ]
        }
    }
]
```

Chapter 6

Results

6.1 New Dataset

In order to have data to test the module a set of people were recruited to make calls to the dialog agent, which were recorded. A total of 15 people participated, with various ages and levels of education and different occupations, such as teaching, selling, healthcare, marketing, and students. They were mainly family members and colleagues, with a few friends. A document was given to them describing, very generally, what interactions the bot can have, and some basic tasks to perform in the calls, with slight hints of how to say the request. They were also asked to, to the better of their abilities, act like they were talking to a bot from a store, and really wanted to have the task done. Some were provided help through a phone call or in person. The calls were made using a SIP client application to connect to a domain in a GoContact SIP server.

From the calls to the dialog agent resulted a dataset constituted by 1709 ASR segments, originated from 246 calls. Each call has an average of 6,95 ASR segments, with the longest having 51. There are a total of 9210 words, giving an average of 5,39 words per segment.

This and additional information on the dataset is presented in table 6.1.

Table 6.1: Some statistics about the dataset

Participants	15
Participants age (min, max, mean)	8, 60, 32
mean number of calls per participant	16
number of calls	246
ASR segments (total, min, max, mean)	1709, 1, 51, 7
words (total, min, max, mean)	9210, 1, 67, 5

The division of each ASR segment was made according to the Google Speech-to-Text API output, with the input being an audio file in .wav format mono, bit rate of 128 kb/s, and sample rate of 8 kHz, which contains the raw recording of the call, with both the operator and the client speeches. For this reason, some ASR segments in the dataset may have both as well, like in the example in table 6.2, with the bot speech between parenthesis.

Table 6.2: Example of a call transcription from the dataset from 28/09/2021

1	(Em que posso ajudar) Boa noite queria fazer o encomenda (o que pretendo comprar) um blazer
2	(Qual a cor)
3	preto
4	(Qual é o tamanho)
5	M
6	(encomenda completa)
7	não quero também os sapatos
8	(qual a cores)
9	pretos
10	(Qual é o tamanho)
11	39
12	(encomenda completa) sim
13	(encomenda finalizada mais alguma questão) mais nada obrigada

Sometimes it is well separated and in others it gets mixed. This can be due to the speed of the response given by the person after the end of the bot line, and the conditions of the recording.

Another observation is since the participants in the calls were aware that they were talking to a robot, their responses appear to be conditioned, tending to be neutral and direct.

6.2 Evaluation of Speech-to-Text

As a first step in the analysis, the confidence of the speech-to-text step was evaluated, despite the mixing of speech lines.

In Fig. 6.1 is plotted the confidence value returned by the speech-to-text (for the best alternatives) by number of occurrences.

In the figure is shown that the majority of transcriptions have a confidence above 90%, but none appear to reach 100%. Nonetheless, the performance can be considered good, as the amount of segments with confidence below 70% is minimal and below 50% is negligible.

In table 6.3 are some more useful values for the performance evaluation, where is shown that the confidence does not exceed 95%.

Table 6.3: ASR statistics

mean	min	more than 50%	max
87.46	16	92	95

It is possible to verify some of the cases with maximum ASR confidence that actually correspond to the spoken words in table 6.4.

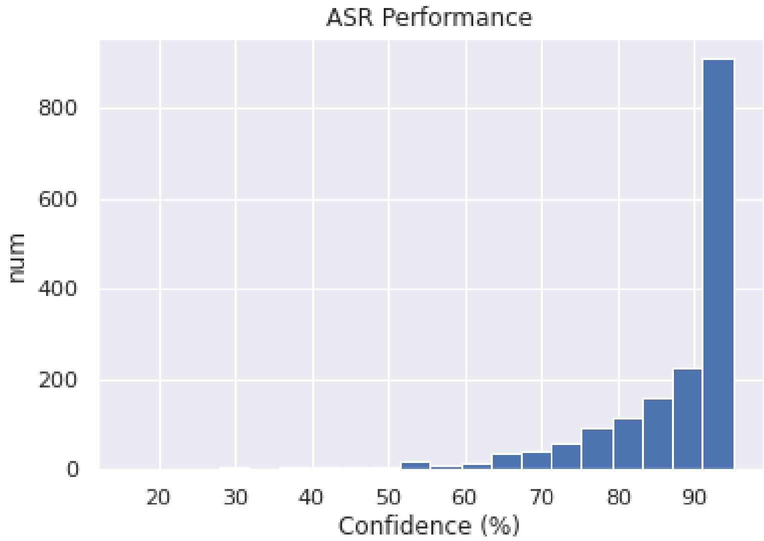

Figure 6.1: Confidence of the ASR algorithm by number of transcriptions

Concerning the bot speech lines, some that are presented in the table have appeared various times with confidence 95%. Taking into account that these were not specifically desirable, they may be skewing the ASR confidence to better values than if only the client's words were present.

6.3 Evaluation of text based sentiment analysis

In this section, it is assessed the performance of the text-based sentiment analysis module through the confidence levels of the output of LinguaKit and MeaningCloud for the created dataset. The presence of both client and bot in the same segment of text is a problem, because the module analyses it as one sentence, potentially giving very different results from what would be expected. Yet, let's assume that the text corresponding to the dialogue agent speech is neutral (which is not).

Later both analyses provided by the processing methods are correlated to find if there are any meaningful results.

In fig 6.2 is the number of occurrences by classification tag for the two methods.

The most common classification is "NONE", for both, what is to be expected in this scenario of consumer interaction with a bot. There are more instances of positive rather than negative sentiment overall, with the difference much more accentuated in LinguaKit. MeaningCloud seems to have more balance between positive and negative. This algorithm has other classification tags, "NEU", which can be considered the same as "NONE", indicating neutral sentiment, "P+", a higher degree of positiveness, and "N+", more negative, which had no occurrence.

Table 6.4: Some correct ASR segments

1	(Em que posso ajudar) preciso de uma informação
2	(vendemos tudo mais alguma questão) não
3	(Em que posso ajudar)
4	Queria informações
5	Quero fazer uma compra
6	(encomenda finalizada mais alguma questão)
7	o casaco em preto e as calças em azul
8	(Qual é o tamanho)
9	(encomenda completa)
10	sim
11	não
12	(Em que posso ajudar) boa noite Eu queria fazer uma devolução de umas calças
13	que artigos vendem
14	(Qual a morada)
15	Quero comprar uma camisa
16	quero devolver umas calças
17	ok
18	(encomenda finalizada mais alguma questão) não é tudo
19	como posso fazer uma compra
20	(o que pretende comprar)
21	rosa
22	S

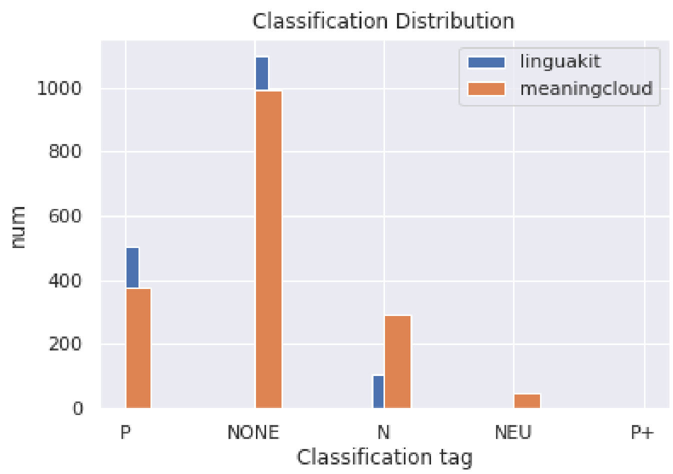

Figure 6.2: Absolute number of occurrences of classification tags

6.3.1 Confidence

The confidence values provide a way to generally evaluate the performance of both processing methods. These values are presented in figure 6.3, by the number of occurrences.

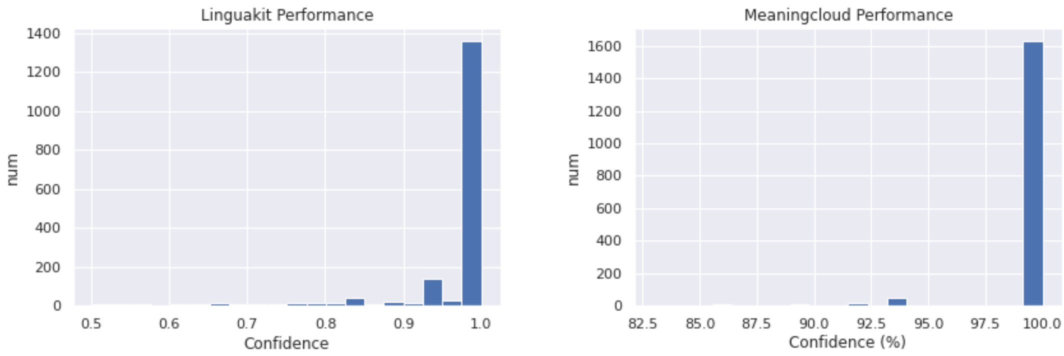

Figure 6.3: Confidence by number of transcriptions of linguakit (left) and meaningcloud (right)

By the figures is safe to assume that both algorithms are very confident in their final classification for the great majority of the text segments. MeaningCloud is the most confident having 100% in the great majority of segments, with more than 1600. LinguaKit returned a wider range of values for the confidence, but it is very high for the majority of segments as well.

In figure 6.4 is presented the mean confidence per classification tag for LinguaKit and MeaningCloud, respectively.

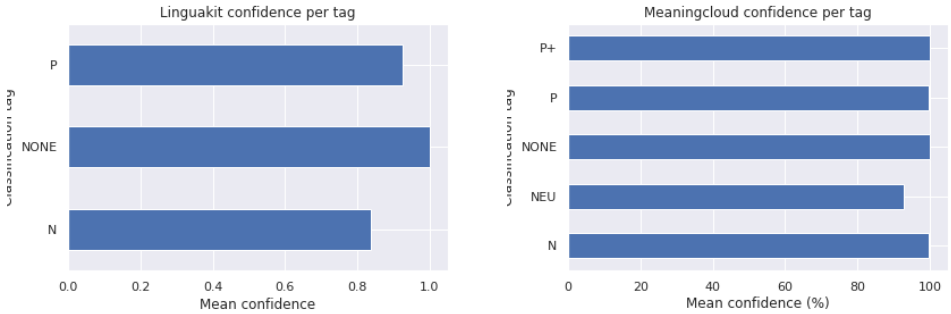

Figure 6.4: Mean confidence per classification tag of linguakit (right) and meaningcloud (left)

For Linguakit we can see that the negative classification has the worst confidence in general, slightly above 0.8. The "NONE" classification is always 1, so it probably is a default result.

The only significant variation in the mean of confidence of MeaningCloud is just for the tag "NEU", which is around 90%, all the others have a mean of 100%.

These values of confidence can indicate that the classification given by Meaningcloud can have the same weight independently of the tag that it gives, except for "NEU". In the case of Linguakit, it probably has less confidence in the result when returning a negative classification. With the distribution in fig 6.2 it is expected that the system may be worst at detecting true negative sentiment than positive.

For a clearer view of the confidence per sentiment classification the histograms in figure 6.5 were made. The `NONE` classification has always maximum confidence so it is not displayed.

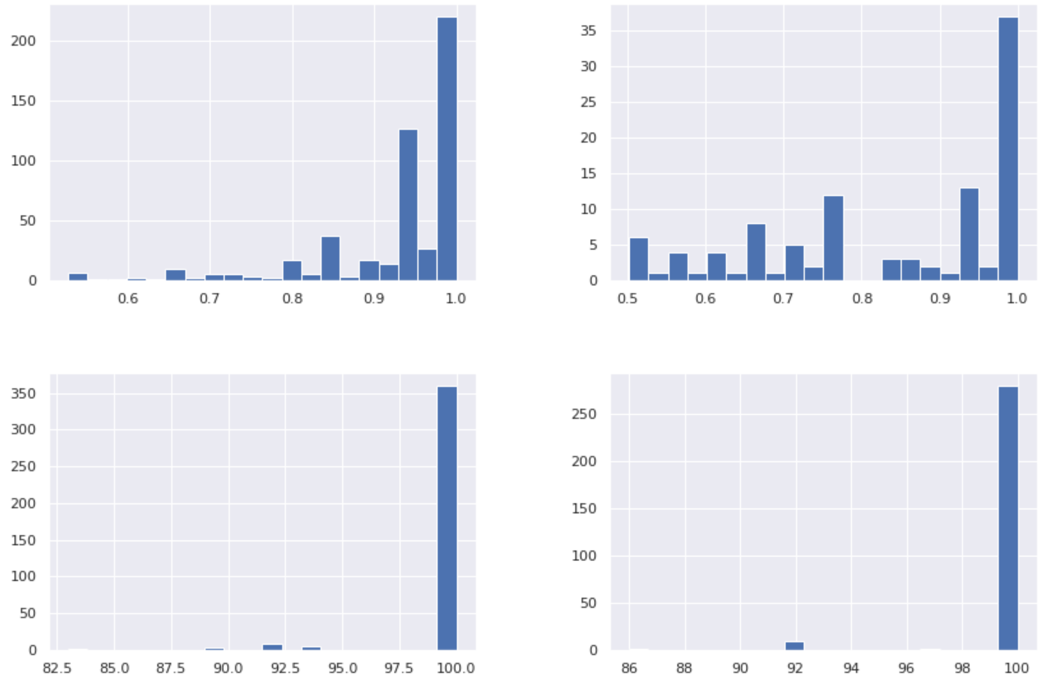

Figure 6.5: Histograms of confidence obtained with Linguakit (top) and Meanincloud (down) for: P (left), N (right).

The lack of confidence in the negative result is further confirmed, with a much more evenly spread out occurrence of confidence values from 0.5 to 0.95. For the `P` classification, the majority of times the confidence has a value higher than 0.8, so it can be considered good. This is, in the LinguaKit results. MeaningCloud practically always returns very high values of confidence, with values below 99% being almost a blip.

6.3.2 Linguakit vs MeaningCloud

The advantage of using two different processes is that, in theory, one can be more certain that the result for a certain segment is correct if it is the same in both. A confusion matrix is useful in this situation, as it enables an easy overview over the occurrences of the different combinations of results, like in Fig. 6.6. LinguaKit does not return as classification "NEU" or "P+", those are only present for the matrix to be squared.

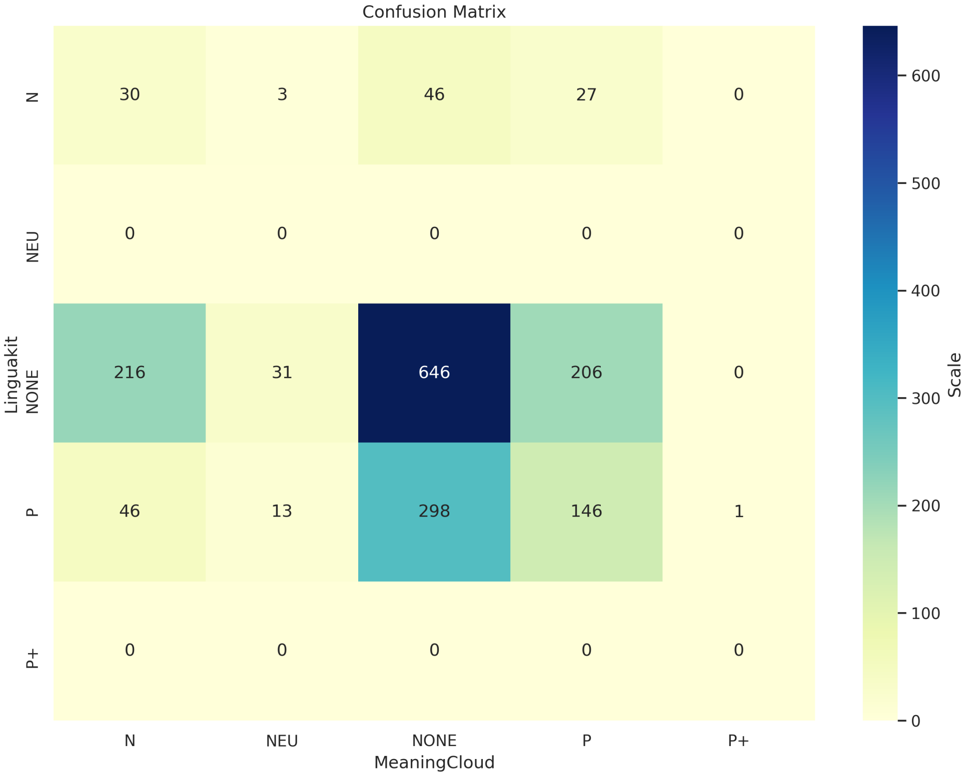

Figure 6.6: Correlation between the results from linguakit and meaningcloud

As seen in the figure, it is in the `NONE` classification that both agreed more by far, being this combination the most common case of result, predictably, as this is the most common classification. The second most agreed is positive, with 146 matches, and the worst matching occurred for the negative classification, with only 30 cases where this was the result for both algorithms. However, there are more three significant cases. The combination of `P` by LinguaKit and `NONE` by MeaningCloud is the second most common, followed by `NONE` by Linguakit and `P` or `N` by MeaningCloud, very close from each other. Considering the `NONE` the default classification, the one given when none other was found, it is expected that the case where one method gives `NONE` and the other gives one of the two polarities would be common. As LinguaKit returned significantly less negative results than MeaningCloud it is not paired as often with a `NONE`. There are also some instances where both give opposite classification to the same text, and this occurred more often than both agreeing with a negative classification.

6.3.3 Negative examples

In table 6.5 are presented the segments for which the classification was negative from both processing methods.

For the great majority of these segments is understandable the negative classification. There are instances where we can see that the client is trying to say to the bot that it is mistaken, like in segment 4 in the table, where he is asked what he wants to buy, and answers saying that he doesn't want to buy anything, he wants to do a return. In other cases it is caught the frustration of the client, as in segment 22, "Forget it, I don't want anything".

Table 6.5: Transcriptions classified with negative in both algorithms (duplicates omitted)

1	o casaco em preto e as calças em azul
2	(Quero fazer uma encomenda) não quero pedir uma informação
3	(Em que posso ajudar) fazer uma devolução (o que pretende comprar) não quero fazer uma devolução
4	(o que pretende comprar) comprar não quero comprar nada queria fazer uma devolução
5	(Em que posso ajudar) precisava de informações (o que pretende sabias) que tudo vendem (vendemos tudo mais alguma questão) Sim (Em que posso ajudar) Estava a fazer uma pergunta (o que pretendo comprar) Quero fazer uma pergunta (o que pretende comprar)
6	não quero fazer uma devolução de umas calças 12345 (recolhe loja última cílio)
7	não quero pedir informações informação assim
8	bêbada Quero comprar uns boxers
9	(Qual é o tamanho) o teu pode pedir o teu tamanho completa não quero pedir informações
10	(recorde loja ao domicílio) domicílio caralho
11	(Em que posso ajudar) ela devolver vários produtos (qual é o número da encomenda) na algum preço (olhe loja o domicílio mais alguma questão)
12	não está conseguindo Então eu era
13	pode ir Queria fazer uma reclamação Queria fazer uma reclamação vou comprar um um Blazer preto
14	(Qual é o número da encomenda) 12345 não quero encomendar
15	(o que pretende comprar) Não quero devolver uma compra
16	(Qual a morada) é rua do tapete (nos próximos 3 dias aqui) OK (o e-mail com os dados mais alguma questão) sim
17	(Pretendo comprar) quero devolver uma encomenda
18	é preto
19	(Pretendo comprar) não quero fazer uma devolução
20	(o que pretendo comprar) comprar Quero quero devolver
21	quero perguntar como é que se faz uma encomenda
22	Esquece não quero nada
23	seu alarme
24	enganei-me
25	(mais alguma questão) quero devolver um produto
26	(Em que posso ajudar) queria pedir uma informação (o que pretende sabias)
27	(o que pretendo comprar) comprar não quero comprar
28	(encomenda completa) não gostava de fazer uma função

Chapter 7

Conclusion

7.1 Work Summary

The work done in this dissertation is described in several phases. The first is characterized by obtaining knowledge in necessary areas of research, such as emotion theory, sentiment analysis, machine learning and dialogue systems, and the first approach to DialogFlow. This process continued a bit throughout the extent of the work.

Then initial scenarios regarding possible use cases of the Communication Monitorization Module were created and later refined. From these scenarios were extrapolated the requirements the following work would follow.

There was research about existing tools that could be used in sentiment analysis, NLP, speech-to-text, and bot development to fulfill some of the requirements, and some were evaluated in their potential to be integrated.

With tools chosen the development of the module was initiated, first the speech-to-text component followed by the first iteration of the text-based sentiment analysis module. This was supposed to have a self-trained algorithm, but data was needed.

The next phase was the development of the bot, that was needed to acquire the data. With the bot finalized the work proceeded with the acquisition of the data through calls made to the bot.

Finally, the data gathered was analyzed along with the performance of the module. The data seems insufficient for training an algorithm.

7.2 Main results

The main results of the work developed in this thesis are:

Simple conversational agent – A working bot was developed in DialogFlow in partnership with GoContact with in-house tools, that serve as a conversational interface for a made-up clothing business, allowing to purchase items, make returns, and provide simple initial forms of accessing information regarding products. The agent was developed to create difficulties in the conversation, aiming at fostering negative reactions from users.

Call dataset – As a result of calls to the conversation agent, trying to mimic consumers' interactions, a first dataset was created, containing the recordings and text transcriptions.

First evaluation of text-based sentiment analysis tools for the task – A first version of the Text-based Sentiment Analysis Module, based on the existing tools (Google speech-to-text, LinguaKit, and MeaningCloud) was evaluated with the acquired data.

7.3 Future work

There are many possible continuations of the work presented in this Dissertation, including some requirements that were not met. Some possibilities are:

- Continue dataset acquisition for having enough data for machine learning;
- Train a neural-network-based sentiment analysis algorithm, following, for example, the method based on a Recurrent Neural Network described in [62];
- Explore technologies for extracting emotions from speech and implement the first version of a speech analysis module;
- Method for the fusion of the text and speech analysis;
- Enhance API functionality to reflect new capabilities added to the module;
- Explore the possibility of "real-time" application of the module, during the call;
- Explore existing works on sentiment analysis results integration with chatbots and task-oriented dialogue agents;
- Close the loop by integrating the results from the module processing in the bot's logic;
- Improve sentiment classification method to detect different emotions and moods.

Bibliography

[1] BRAIN [BRN.AI] CODE FOR EQUITY. Chatbot report 2019: Global trends and analysis. Chatbots Magazine, 2019. (Accessed 28 oct 2021.

[2] Chatbot market 2021: Stats, trends, size & ecosystem research. Insider, 2021. (Accessed 28 oct 2021).

[3] D Jurafsky and JH Martin. *Speech and Language Processing*, chapter 20: Lexicons for Sentiment, Affect, and Connotation. 2021. (draft).

[4] Myriam Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111, 2014.

[5] Paul Ekman. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16, 1999.

[6] D Jurafsky and JH Martin. *Speech and Language Processing*, chapter 24: Chatbots & Dialogue Systems. 2021. (draft).

[7] Contact centre vs communication centre vs call centre – but which one is right? EWA Bespoke Communications Blog, 2010. (Accessed 21 oct 2021).

[8] Bob Russell. Call centres: A decade of research. *International journal of management reviews*, 10(3):195–219, 2008.

[9] J Pearce. The history of the call centre–updated. *United Kingdom, Call Centre Helper*, 2018.

[10] Keith Kirkpatrick. Ai in contact centers. *Communications of the ACM*, 60(8):18–19, 2017.

[11] Jordi Luque, Carlos Segura, Ariadna Sánchez, Marti Umbert, and Luis Angel Galindo. The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls. In *INTERSPEECH*, pages 2346–2350, 2017.

[12] Vera Cabarrão, Mariana Julião, Rubén Solera-Ureña, Helena Moniz, Fernando Batista, Isabel Trancoso, and Ana Isabel Mata. Affective analysis of customer service calls. *ExLing 2019*, 25:37, 2019.

[13] Jon Irastorza and María Inés Torres. Tracking the expression of annoyance in call centers. In *Cognitive Infocommunications, Theory and Applications*, pages 131–151. Springer, 2019.

[14] Bryan Li, Dimitrios Dimitriadis, and Andreas Stolcke. Acoustic and lexical sentiment analysis for customer service calls. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5876–5880. IEEE, 2019.

[15] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[16] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

[17] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374, 2018.

[18] Yukun Ma, Khanh Linh Nguyen, Frank Z Xing, and Erik Cambria. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70, 2020.

[19] D Jurafsky and JH Martin. *Speech and Language Processing*, chapter 26: Automatic Speech Recognition and Text-to-Speech. 2020. (draft).

[20] Jack Cahn. Chatbot: Architecture, design, & development. *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*, 2017.

[21] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[22] Ibm watson is ai for smarter business. IBM. (Accessed 22 oct 2021).

[23] Accelerate your transformation with google cloud. Google Cloud. (Accessed 22 oct 2021).

[24] Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. Gus, a frame-driven dialog system. *Artificial intelligence*, 8(2):155–173, 1977.

[25] Dialogflow. GoogleCloud. (Accessed 22 oct 2021).

[26] Kamran Iqbal and Jonathan Fingold. What is the bot framework sdk? Microsoft, 2021. (Accessed 22 oct 2021).

[27] Watson assistant: Intelligent virtual agent. IBM. (Accessed 22 oct 2021).

[28] Rasa - the future of customer experience. RASA. (Accessed 22 oct 2021).

[29] Amazon lex. AWS. (Accessed 22 oct 2021).

[30] Ben Brown. Botkit: Building blocks for building bots. Github, 2021. (Accessed 22 oct 2021).

[31] Pandorabots. Pandorabots. (Accessed 22 oct 2021).

[32] Meet the botpress stack the building blocks for building chatbots. Botpress. (Accessed 22 oct 2021).

[33] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[34] Ashima Yadav and Dinesh Kumar Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, 2020.

[35] Shamsuddeen Hassan Muhammad, Pavel Brazdil, and Alípio Jorge. Incremental approach for automatic generation of domain-specific sentiment lexicon. In *European Conference on Information Retrieval*, pages 619–623. Springer, 2020.

[36] D Jurafsky and JH Martin, editors. *Speech and Language Processing*, chapter 4: Naive Bayes and Sentiment Classification. 2021. (draft).

[37] Akshi Kumar and Mary Sebastian Teeja. Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, 4(10):1, 2012.

[38] Paula Carvalho and Mário J Silva. Sentilex-pt: Principais características e potencialidades. *Oslo Studies in Language*, 7(1), 2015.

[39] Pablo Gamallo and Marcos Garcia. Linguakit: uma ferramenta multilingue para a análise linguística e a extração de informação. *Linguamática*, 9(1):19–28, 2017.

[40] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558, 2010.

[41] Marlo Souza and Renata Vieira. Sentiment analysis on twitter data for portuguese language. In *Proceedings of the 10th International Conference on Computational Processing of the Portuguese Language*, PROPOR'12, pages 241–247, Berlin, Heidelberg, 2012. Springer-Verlag.

[42] Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[43] Hugo Gonçalo Oliveira, Leticia Antón Pérez, and Paulo Gomes. Integrating lexical-semantic knowledge to build a public lexical ontology for portuguese. In *International Conference on Application of Natural Language to Information Systems*, pages 210–215. Springer, 2012.

[44] Cláudia Freitas. Sobre a construção de um léxico da afetividade para o processamento computacional do português. *Revista Brasileira de Linguística Aplicada*, 13(4):1031–1059, 2013.

[45] Cláudia Freitas, Eduardo Motta, R Milidiú, and Juliana César. Vampiro que brilha... rá! desafios na anotaçao de opiniao em um corpus de resenhas de livros. *Encontro de Linguística de Corpus*, 11:22, 2012.

[46] Grupo de Pesquisa em Processamento da Linguagem Natural da Pontifícia Universidade Católica do Rio Grande do Sul Porto Alegre RS Brasil. Ontolp ¿ recursos.

[47] Erik Cambria and Amir Hussain. Senticnet. In *Sentic Computing*, pages 23–71. Springer, 2015.

[48] Nathan Hartmann, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues, and Sandra Aluisio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017.

[49] Monkeylearn api. MonkeyLearn. (Accessed 28 oct 2021).

[50] Semantria api. Lexalytics. (Accessed 28 oct 2021).

[51] Salience. Lexalytics. (Accessed 28 oct 2021).

[52] How can you use meaningcloud? MeaningCloud. (Accessed 25 oct 2021).

[53] Rosette sentiment analyzer. Rosette Text Analytics. (Accessed 28 oct 2021).

[54] Repustate sentiment analysis. Repustate. (Accessed 28 oct 2021).

[55] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

[56] Francois Chollet et al. Keras, 2015.

[57] Denilson Alves Pereira. A survey of sentiment analysis in the portuguese language. *Artificial Intelligence Review*, 54(2):1087–1115, 2021.

[58] GoContact. Real time decisions — um filme gocontact. Youtube, November 2019. (Accessed 24 oct 2021).

[59] dialogflow documentation — google cloud_2021, 2021.

[60] Sentiment analysis api version 2.1 - what is sentiment analysis? MeaningCloud. (Accessed 25 oct 2021).

[61] Steven Loria. Textblob: Simplified text processing. Github, 2021. (Accessed 28 oct 2021).

[62] Paul J Deitel and Harvey Dietal. *Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud - Chapter 16.9*. Pearson Education, Incorporated, 2020.