A Natural Language Understanding Model COVID-19 based for chatbots

Conference Paper · October 2021

DOI: 10.1109/BIBE52308.2021.9635248

6 authors, including:

Valmir Oliveira dos Santos Júnior
Universidade Federal do Ceará
5 PUBLICATIONS **5** CITATIONS
SEE PROFILE

João Araújo Castelo Branco
Universidade Federal do Ceará
2 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Marcos Antonio De Oliveira
Universidade Federal do Ceará
50 PUBLICATIONS **325** CITATIONS
SEE PROFILE

Ticiana Linhares Coelho da Silva
Universidade Federal do Ceará, Quixadá, Brazil
56 PUBLICATIONS **306** CITATIONS
SEE PROFILE

A Natural Language Understanding Model COVID-19 based for chatbots

Valmir Oliveira dos Santos Júnior, João Araújo Castelo Branco, Marcos Antonio de Oliveira,
Ticiana L. Coelho da Silva, Lívia Almada Cruz, Regis Pires Magalhães
Insight Data Science Lab - Universidade Federal do Ceará (UFC)
Email: {valmir.oliveira, joaocb, marcos.oliveira, ticianalc, livia, regis}@insightlab.ufc.br

***Abstract*—It is increasingly common to use chatbots as an interface to use services. Making this experience more humanized requires the chatbot to understand natural language and express itself using natural language. One crucial step to achieve this is to label the data with intentions and entities. After labeling, one can use the labeled data to train a Natural Language Understanding (NLU) component. The NLU component interprets the text extracting the intentions and entities present in that text. Manually label the data is an onerous and impracticable process due to the high volume of data. Thus, an unsupervised machine learning technique, such as data clustering, is usually used to find patterns in the data and thereby label them. For this task, it is essential to have an effective vector embedding representation of texts that depicts the semantic information and helps the machine understand the context, intent, and other nuances of the entire text. In this paper, we perform an extensive evaluation of different text embedding models for clustering, labeling, and training an NLU model using the text of attendances from the Coronavirus Platform Service of Ceará, Brazil. We also show how different text embeddings result in different clustering, thus capturing different intentions of patients.**

I. Introduction

Currently, it is increasingly common to use chatbots as an interface for interacting with users for the most varied types of services, from obtaining information to making hotel reservations. In most cases, these interfaces present a rigid conversation flow to be followed, providing a less humanized experience. Making conversations with the machine ever closer to human-to-human conversations is an old problem in computer science and artificial intelligence [1].

It is necessary to give the machine greater generalization power in text interpretation to make the conversation experience more flexible and humanized. An essential component of a chatbot is the Natural Language Understanding model (NLU). This component is responsible for interpreting the information provided as input by the user, generating data that the chatbot can understand and interpret. The NLU model identifies user intent and extracts domain-specific entities from unstructured data. An intent represents a mapping between what a user says and what action must be performed by the chatbot. Actions correspond to the steps that the chatbot will take when the user triggers specific intentions. An entity is what or who is talked about on user input [2]. For example, consider the sentence "What are the symptoms of COVID-19?". The user intends to know the symptoms. The entity value is COVID-19.

In most cases training the NLU module is not an easy task. A large amount of annotated data is necessary to train the model and achieve high accuracy. Often the data are annotated manually, which takes a long time to accomplish. An alternative to help the data annotation is to apply unsupervised learning techniques to discover patterns or relationships between the data. In that approach, we aim at finding the most informative clusters of data, when together they can represent a class [3]. One such technique is data clustering. In this way, we can apply a clustering algorithm for a set of existing dialogues to identify the intentions represented by each cluster. Then, we use the intentions as inputs for training the NLU model.

Text clustering typically requires text input to be represented as a fixed-length vector. A common representation is to use bag-of-words or bag-of-n-words [4] due to simplicity. However, these techniques have some disadvantages, such as sparsity and high dimensionality. Pre-trained word embeddings have been widely used due to their ability to capture the context of a word in a document, semantic and syntactic similarity to other words. However, if small changes occur between one sentence and another, word embeddings may not effectively capture the change in meaning. Despite the semantically opposite nature, the cosine similarity can be very high between the vectors obtained from these sentences using word embeddings. An alternative to this limitation is to use sentence embeddings [5][6][7].

Such models take the text as input and generate an output of a single fixed-size vector that represents the entire sentences and their semantic information. That helps the machine understand the context, intent, and other nuances of the entire text. In this work, we use the dialogues of health professional advice carried out on the Platform of the Coronavirus Service[1] (PCS) in the state of Ceará, Brazil to train an NLU component to identify intentions and symptoms from them, based on text embeddings and an unsupervised learning strategy.

Our contributions are: (1) evaluation of different sentence embeddings and word embeddings strategies for the problem of discovering intentions in dialogues about COVID-19. These embeddings are evaluated based on the quality of the generated

[1]https://coronavirus.ceara.gov.br

intention clusters. In this paper, we performed the experiments using Bert LaBSE [8], FLAIR [9], BERTimbau [10], Glove [11], and MUSE [5] embeddings. All of them pre-trained for Brazilian Portuguese; (2) another contribution is an NLU model, which can classify the messages sent by users via chatbot, assigning intentions with the marking of entities found in the sentence.

We organized remaining sections as follows. Section II explains the preliminary concepts required to understand this work and some related works. Section III shows the steps necessary to accomplish our goals. Section IV presents our experiments and their analysis. Finally, Section V summarizes this work and proposes future developments.

II. Background and Related Works

In this section, we discuss the background material as well as related studies to our problem.

A. Word Embeddings

Vector space models transform the text of different lengths into a numeric fixed-length vector to be fed into downstream applications, such as similarity detection or machine learning models. Pre-trained word embeddings have been widely used [12], [11], [9], [10], due to their ability to capture the context of a word in a document, semantic and syntactic similarity to other words.

Word2vec [12] is a framework for learning the word vectors by training a language model that predicts a word given the other words in a context. The main drawback is that it poorly utilizes the statistics of the corpus since the model is trained on a separate local context window instead of on global co-occurrence counts. [11] bypasses this problem and proposes a model that produces a word vector space. [11] trains the model on global word-word co-occurrence counts and makes efficient use of statistics. FLAIR [9] abstracts away from specific engineering challenges that different types of word embeddings add to. FLAIR creates a unified interface for all word embeddings and sentence embeddings, and arbitrary combinations of embeddings.

BERTimbau [10] provides BERT models for Brazilian Portuguese. The models were evaluated on three NLP tasks: sentence textual similarity, recognizing textual entailment, and named entity recognition. BERTimbau improves the state-of-the-art on these tasks over multilingual BERT and previous monolingual approaches for Portuguese.

From word embeddings, we can obtain document vectors by averaging together all word vectors. However, this procedure gives the same weight to both important and unimportant words. Another limitation of representing text using word embeddings is, each word would be embedded with the same vector regardless of the context. An extension of word embeddings is document or sentence embeddings to obtain the document vectors directly. From now on, we consider sentence, document and paragraph embedding as the same thing.

B. Sentence Embeddings

Sentence Embedding represents sentences in an n-dimensional vector space such that semantically similar or semantically related words come together in the training method. Sentence Embedding performs the representation of a sentence, which can have different representations of a word based on its context.

There are plenty of proposals for sentence embeddings as InferSent [13], LaBSE [8], Universal Sentence Encoder [6], Doc2Vec [7], among others. Universal Sentence Encoder proposes two different encoders. One makes use of the transformer architecture [14] and achieves the best performance. The attention mechanism computes context-aware representations of words in a sentence that takes into account both the ordering and identity of all the other words. The context-aware word representations are converted to a fixed-length sentence encoding vector by computing the element-wise sum of the representations at each word position [6]. We refer the reader to [15] for further details about attention mechanism. The other encoder proposed is based on a deep averaging network (DAN) [16] whereby input word embeddings and bi-grams are first averaged together and then passed through a feedforward deep neural network to produce sentence embeddings. [5] extends [6] by proposing MUSE, a text embedding model for sixteen languages into a single semantic space using a multi-task trained dual encoder.

Similar to Word2Vec, Doc2Vec trains the paragraph vectors (or sentence embeddings) in the prediction task of the next word given many contexts sampled from the paragraph. The paragraph vector and word vectors are concatenated to predict the next word in a context. LaBSE [8] is trained and optimized for multilingual sentence-level embeddings. It produces similar representations exclusively for bilingual sentence pairs that are translations of each other. LaBSE employs a dual-encoder whereby source and target sentences are encoded separately using a shared BERT-based encoder, then feeding a combination function. The final layer [CLS] representations are taken as the sentence embedding for each input. The similarity between the source and target sentences is scored using cosine over the sentence embedding produced by the BERT encoders.

C. Natural Language Understanding Models

The NLU model identifies user's intents and extracts domain-specific entities. More specifically, intent summarizes the goal of the user input sentence and is used as a mapping between what a user says and what action must be performed by the chatbot. Actions correspond to the steps that the chatbot will take when the user triggers specific intentions. An entity is what or who is talked about on user input [2].

One of the fundamental tasks in NLU is learning vector-space representations of text. There are two popular approaches: multi-task learning and language model pre-training. These techniques are combined in the proposal of a Multi-Task Deep Neural Network (MT-DNN). Human learning has inspired this approach in the sense that they often apply the knowledge learned from previous tasks to help realize a new

task [17] and, as well, using tasks simultaneously can benefit from each other learned skills.

[18] presents a survey with methods that demonstrate how pipelines of sequential tasks are applied to achieve NLU using language model pre-training. To apply a pre-trained model to specific NLU tasks often is necessary to fine-tune it, for each task, with additional task-specific layers using task-specific training data. [17] argues that multi-task learning and language model pre-training are complementary technologies, making possible their combination to improve the learning of text representations, increasing the performance of NLU tasks.

Commonly Accuracy and F1-score are the metrics used to evaluate a model's prediction quality. NLU is a pre-processing step for later modules in a chatbot system, and its performance interferes directly with the overall quality of the chatbot [18]. Multi-class classification through neural approaches is common in recent literature, and that technique is specially used for the tasks of domain and intent classification. For short sentences, where context is necessary to infer information, recurrent and convolutional neural networks are applied because they consider text before the current utterance [19].

As for slots filling or entities identification, often sequence classification is used [18]. In this approach, the classifier predicts semantic class labels for subsequences of the input utterance [20]. Recurrent neural networks are applied for this task offering good results [21].

Rasa[2] is an open-source machine learning framework for automated text and voice-based conversations. Understand messages, hold conversations, and connect to messaging channels and APIs. Rasa NLU module works with a pipeline of components to train a model capable of extracting intents and entities from raw text using as input an annotated dataset. Rasa provides as well tools for testing the NLU model performance. The pipeline can be customized to the necessities of the model and makes possible the fine tune of the dataset. Pre-trained word embeddings can be present in the pipeline adding versatility to the trained model. Each component processes input and/or creates an output. The output of a component can be used by any other component that comes after in the pipeline. Rasa provides a significant number of pre-trained models for different languages, including BERT and GPT[3].

D. COVID-19 Related Chatbots

Due to the high demand for patient follow-ups, other groups have recently worked to develop COVID-19 related chatbots. [22] trains a NER model using scientific articles extracted from COVID-19 Open Research Dataset, CORD-19 [23]. The group has used the papers to extract entities usable to identify symptoms in patients' written sentences. They use Word clouds to find the most frequent symptoms in the articles, and the chatbot NLU model is used to build a knowledge graph that helps keep track of follow-ups from returning patients. [24] applies natural language and argumentation graphs together to build dialog systems that explain why a chatbot gave specific advice on COVID-19 vaccination. [25] article raises questions and problems that a chatbot could address during a pandemic like COVID-19. Initiatives such as Clara[4] from the CDC in the United States come to deal with the spread of conflicting information caused for lack of knowledge and fake news that ultimately can make dealing with the pandemic situation much more difficult.

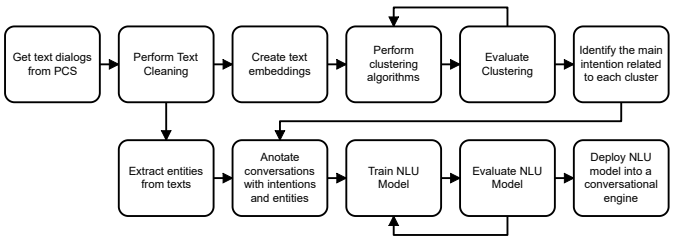


Fig. 1. NLU Pipeline.

In what follows, we describe our pipeline from collecting the dataset to the deployment of the NLU model.

III. Data and Methods

A. Dataset Description

The PCS dataset has mechanisms for screening patients through interaction via chatbot. The service is an online chat where a chatbot performs the first interaction. Based on the patient's answer to some predetermined questions, the chatbot classifies the patient's condition according to criticality, which can be mild, moderate, or severe. After this first interaction, depending on the criticality classification, the patient is directed to Tele assistance with a health professional. The interaction between patient and health professional provides more details about patient conditions, including more specific symptoms, which can be physical or psychological. In the end, the patient evaluates the service. For this evaluation, the PCS requires the patient to answer the question "Are you pleased with the service?", and the response should be "Yes" or "No"; and an evaluation score ranging from 0 to 10, where 0 is the lowest satisfaction rating and 10 the highest satisfaction rating. Our proposal aims to find the best dialogue representations to automatically identify intentions and symptoms while the patients report their health conditions. The intentions we are interested in here are the ones related to the patient's diagnosis.

So the PCS dataset used in this approach is the set of dialogues between patients and health professionals with a positive evaluation. In practice, the dialogs that the patient has informed to be satisfied with the attendance, and has assigned the score of 10. It is composed of $1,237$ dialogues collected from May 1, 2020, to May 6, 2020, with a total of $53,633$ sentences. The sentences from the dialogues are annotated with their actors (patients or health professionals). From the total of sentences, $26,647$ sentences are from the patients,

[2]https://rasa.com/open-source/

[3]see https://huggingface.co/models for a complete list of available models

[4]https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/testing.html

and $26,986$ sentences are from the health professionals. The pipeline from the data collection to the deployment of the NLU model is represented in Figure 1 and described as follows. As the chatbot's goal is to identify the criticality of the patient, we have used only the patient's sentences in the clustering process.

B. Text Cleaning

The first step is the text cleaning approach, and it consists of removing duplicate sentences from dialogues. After, we select the dialogues highly evaluated by patients, i.e., conversations with a score equal to 10. Remember, at the end of each conversation, the patients should rate the conversation service with a score between 0 and 10. We also remove sentences represented by the following entities: ZIP code, Social Security Number (SSN), phone number, URLs, Emoticons, patient names, and places. We remove such entities manually to avoid the creation of clusters not directly related to a relevant intention. Also, it prevents the use of any personal information from users in the chatbot training process.

C. Extract Entities

We use a tool called SINTOMATIC [26] to detect and identify symptoms present in natural language text dialogues. SINTOMATIC helps identify patterns of disease signs and new symptoms or rare symptoms, which health professionals have not mapped yet. Thus, following the evolution of COVID-19 symptoms from user dialogues.

Therefore, we use SINTOMATIC to extract symptoms from the conversations of the PCS dataset. In this work, the symptoms are interpreted as proper entities for recognizing structured data in NLU models. It is worth to mention we may use different NER models [27], like the ones available on spaCy NER[5], to extract the entities. Still, we prefer to focus on symptoms as entities since they are related to the patient's diagnosis intentions that we are interested in.

D. Extract Intentions

As we aim at identifying the intentions from the conversations related to the patient's diagnosis, we apply an unsupervised learning-based method consisting of i) Embedding vectors generation, ii) Clustering and iii) Intentions Labeling.

i) Embedding vectors generation: The first step aims to generate a vector representation for the patient sentences in the conversations. The vectors resulting from this step should capture syntactic and semantic information of sentences such that sentences related to the same user's intention are close to each other in the vector space. We create the sentences embeddings using a pre-trained model from the state-of-art. In our proposal, we evaluate the sentence embedding models from MUSE [5] and LaBSE, [8]; and the word embedding models, FLAIR [9], BERTimbau [10] and Glove [11]. Concerning the word embedding model, we use the average vector of all word vectors as sentence representation.

[5]https://v2.spacy.io/api/entityrecognizer

ii) Clustering of embedding vectors: To identify the set of patient's intentions and to label the conversation text, this step performs the clustering of sentence embeddings generated. For this task, we use the well-known K-means algorithm with cosine similarity. We have chosen the best value of k based on the *Davies-Bouldin Score* [28], *Silhouette Score* [29] and visual inspection. Davies-Bouldin Score is a separation metric given by the average similarity between a cluster and its most similar cluster. The best clustering minimizes that similarity, and the lowest similarity value is 0. Thus, lower values of the Davies-Bouldin Score indicate better clustering. The idea behind using a separate metric is to reduce the overlap of intentions between clusters. Silhouette Score indicates a ratio between cohesion and separation, given by similarity between the object and its cluster compared to the similarity between the object and the other clusters. It is important to observe that each embedding approach may result in different values of k.

iii) Intents Labeling: In this step, we aim to identify the intention corresponding to each cluster. Given the high number of sentences, we apply an empirical evaluation through visual analysis. We use t-SNE [30] for the visualization of clusters in high-dimensional data. This tool allows visualizing the distribution of intentions and the clusters overlapping. We use the word cloud for the visualization of clusters' content. Then, we label the sentences with their respective cluster intention to create the training set for the NLU model.

E. Train and Evaluate NLU Model

Before training the NLU Model, we annotate the PCS dataset with the intentions and entities obtained from the previous steps *Extract Entities* and *Extract Intentions*.

NLU training data consists of examples of user utterances categorized by one intent. The training data can also include entities, which are structured into pieces of information extracted from the dialogues.

We use the Rasa NLU component to train our NLU model. It looks for NLU training data and saves a trained model at the end of the process.

We evaluate the precision, recall, F1 Score, and accuracy metrics to assess the trained NLU model. The evaluation consists of assessing the effectiveness of the NLU model for classifying intents and entities using the metrics mentioned above.

F. Deployment of the NLU Model

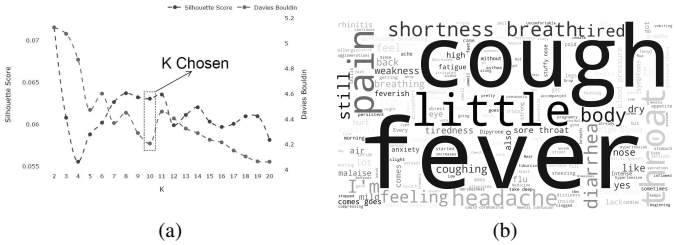
Finally, we deploy the trained NLU Model into a conversational engine. We set up a Continuous Deployment (CD) pipeline to upload the trained model to a conversational server application like a chatbot or a virtual assistant when the evaluation results are satisfactory. NLU models convert user' messages into structured outputs, including the original text, intents, and entities.

IV. Experimental Evaluation

In this section, we discuss the experimental evaluation.

TABLE I
K VALUE CHOSEN FOR EACH EMBEDDING MODEL REPRESENTATION

Embedding	Encoder	Dimensionality	K	SS	DBS
MUSE	Sentence	512	6	0.080	4.283
LaBSE	Sentence	768	10	0.063	4.207
BERTimbau	Word	768	7	0.038	3.634
Glove	Word	300	9	0.070	**3.265**
FLAIR	Word	4096	6	**0.090**	3.586



(a) (b)

Fig. 2. Results for LBSE embedding model. (a) Davies Bouldin Score and Silhouette score. (b) Word cloud representing an intent cluster about symptoms.

A. Choosing the optimal number of clusters

We evaluate the optimal number of clusters for each text embedding model. We performed the experiments by varying the number of clusters k between 2 and 20. However, we chose the range from 5 to 10 since it is feasible to analyze and visualize the possible intentions represented in the clusters. More than ten clusters would be unrealizable to check the cluster's intentions for a human. In addition, we used the *Davies-Bouldin Score (DBS)* and the *Silhouette Score (SS)*, with cosine distance, as the metrics to choose the best value for k within the range from 5 to 10.

Figure 2a shows the results for the DBS metric applied LaBSE. Remember, we aim to minimize the DBS value (the best value is 0). Notice that if we did not define a range to vary the values for k, the number of clusters that achieved better performance would be $k = 20$. To avoid too many clusters with repetitive intents and aiming the best set of clusters, we thought, as a strategy to find a reasonable number of clusters, varying the range from $k = 5$ to $k = 10$, and the ideal number of clusters found was $k = 10$. Consider the best value for SS metric is 1, we can easily see that the values found are less than 1 and slightly the same. Looking at the same range where we found the best k values for the DBS metric, when we analyze the SS metric in Figure 2a, the optimal values are within the interval from $k = 8$ to $k = 10$.

In summary, when we analyze DBS and SS metrics, we end up with $k = 10$ clusters because this k was the best according to DBS and was among the best concerning SS. This methodology was similarly applied to choose k for the other embedding models. Due to space limitations, we omit the analyzes of DBS and SS for each embedding model experimented from the presentation.

According to the results presented in Table I, the values found for SS are far from the best, that is when SS is equal to 1. Indeed, the results are close to zero, indicating an overlapping between clusters found by all the embedding models used in these experiments. This means the distance between a sentence S and the other sentences in the same cluster are almost the same as the distance between S and the sentences assigned to other clusters. This occurs because the vectors have high dimensionality, i.e. above 50. The distances between any two sentences is high. Another factor to point out is related to the DBS metric, which is also far from the best value (zero). So analyzing the best embedding model by looking only at these metrics would not help. Besides, all the embedding models obtained clusters with values for SS and DBS slightly the same.

To further complement the evaluation of the clustering quality, we *visually inspect* some of the results produced by the clusters from the sentences embedded with each embedding model. In general, visual inspection turns out to be a useful tool whenever (1) different approaches produce clusters that have different semantics, (2) different sets of parameters yield clusters that perform well in terms of quality metrics but clearly show different characteristics, or (3) when ground truth is not available. We report that visual was adopted in several past works as well [31].

B. Intents Labeling

Even the ten clusters found by LaBSE did not achieve the best values for DBS and SS metrics, by visual inspection, we could see the intentions from the clusters were more intuitive than those found by the other embedding models. Besides, the clusters could capture the patient's diagnosis that involves the description of the patient's symptoms and the medical prescription. Figure 2b shows a word cloud for one of the LaBSE clusters, which was labeled with the intention *inform_symptoms*. The word cloud was created from the most frequent words gathering all the sentences in the cluster.

Overall, the intentions identified on the clusters were as follows:

1) **greeting**: Sentences related to greetings or salutations, for example, when the patient says thank you, start a conversation, or even says goodbye, in sentences such as *"Hello", "Good morning", "Thank you very much"*
2) **inform**: General information sentences, for example, *"I went to the hospital yesterday", "It happened yesterday"*
3) **inform_symptoms**: Phrases where patient symptoms are reported, examples are: *"I dawned with a headache", "A slight headache"*
4) **inform_medicine**: Phrases where the patient informs some medication they are taking, for example, *"I took paracetamol last night", "I only take dipyrone"*
5) **request_inform**: When the patient requests some information from the health professional, like in these examples: *"Is Dipyrone more effective?", "Where do I get tested?"*

However, only the **LaBSE** model made it possible to identify all the intentions described above, as is shown in

Table II. From the ten clusters found by LaBSE, three clusters are associated with the intention inform_symptoms, four with inform, one with inform_medicine, and one with a greeting.

TABLE II
INTENTIONS DEFINITION

Intention	MUSE	LaBSE	BERTimbau	Glove	FLAIR
greeting	x	x	x	-	x
inform	x	x	x	x	x
inform_symptoms	x	x	x	x	x
inform_medicine	-	x	-	-	-
request_inform	x	x	-	x	x

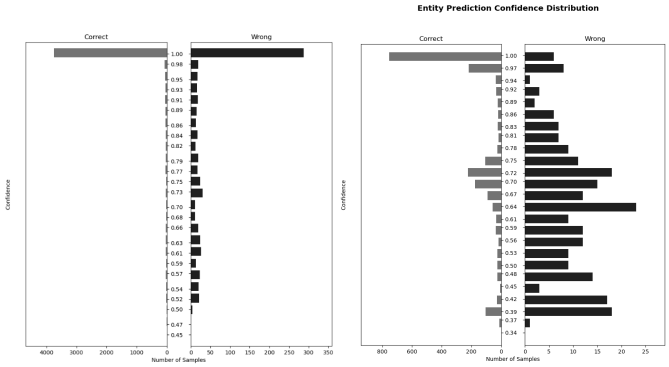
We formulated the intentions based on the chatbot's goal, selecting whether the patient's condition for the COVID-19 is mild, moderate or severe. That is the original goal for the PCS. Our aim is build a proactive and goal-oriented chatbot as an *Intelligent Agent*[1].

C. Training the NLU models

In the Rasa framework, to train an NLU model, we must specify a pipeline of components. They work sequentially and transform the user input, a sentence in natural language, into structured data, later being passed to a machine learning model and attaching more information for decision-making at a given moment in the dialog.

Given the data source, the pipeline should be flexible to possible errors in spelling, punctuation, and abbreviations. Therefore, a pipeline based on spaCy has some favorable features for solving the problem of training an NLU model. First, we must activate the spaCy's module framework in Rasa with the SpacyNLP component that receives a pre-trained template from spaCy itself in the desired language; we use the model "pt_core_news_lg". Next, it is necessary to apply a tokenizer because punctuation errors generate sentences like "comorbidity... has". To perform this task, we used SpacyTokenizer. In addition, it is necessary to include the components responsible for including features in the data. We use SpacyFeaturizer, to include entity information recognized by spaCy. Finally, we apply the *DIETClassifier* classification model for training the NLU model. It is worth mentioning that this algorithm can extract both intentions and entities from the text.

NLU models were trained and tested using the entities obtained by applying SINTOMATIC [26] classifier and the intentions obtained from the clustering, we split the data in 80% to training and 20% to test. For each embedding model, we represent the sentences and run the clustering algorithm. Each sentence is assigned to a cluster and we train an NLU model with such dataset. Table III shows the values of Precision, Recall, F1-score and Accuracy referring to the prediction of intentions. Figure 3a contains the histogram that shows the distribution of prediction of intentions for the model trained with LaBSE. On the left side is the distribution of predictions correctly made, and on the right side, those made incorrectly. The predictions are distributed according to the confidence score. It can be seen that almost all predictions performed correctly had a confidence level equal to 100%. Still, a few incorrect predictions had the same confidence level, this might be due to significant overlap of clusters and distance of samples from different clusters shown during the application of the silhouett score. Although, this model has all the defined intentions and presents a good result compared to the other embeddings.



(a) Prediction of intentions. (b) Prediction of entities.

Fig. 3. Histogram of the NLU model's trained on LaBSE-annotated data.

TABLE III
AVERAGE WEIGHTED OF NLU MODELS BY THE NUMBER OF SENTENCES ASSOCIATED WITH EACH INTENTION

Embedding	Precision	Recall	F1-Score	Accuracy
MUSE	0.872	0.869	0.869	0.869
LaBSE	0.869	0.867	0.867	0.867
BERTimbau	0.851	0.853	0.849	0.853
Glove	0.883	0.881	0.881	0.881
FLAIR	**0.912**	**0.911**	**0.911**	**0.911**

As we can see in Table III on the classification metrics of intentions, the embedding that obtained the best values was FLAIR. However, LaBSE was able to find all the expected intentions like describing symptoms, medical prescriptions, and information requests. LaBSE provides a richer model than the others, giving the chatbot a greater power to identify intentions expected for PCS goal, present in the sentences.

Table IV presents the results for the prediction of entities for each model. The values were approximate because they do not depend directly on the embeddings used, but on the annotations performed using the [26]. So the variations can be due to the division of training and test data, which in general are split randomly for each embedding. Figure 3b shows a histogram containing the distribution of the classifications correctly performed by the model.

V. CONCLUSION

In this work, we study the possibility of using embedding models, in conjunction with clustering algorithms to annotate a dataset with intentions empirically created from the formed clusters. Additionally, we have used this dataset

TABLE IV
METRICS OF THE NLU MODEL ENTITY CLASSIFICATIONS

Embedding	Precision	Recall	F1-Score
MUSE	0.864	0.941	0.901
LaBSE	**0.9**	0.94	0.919
BERTimbau	0.892	0.961	**0.926**
Glove	0.862	**0.966**	0.911
FLAIR	0.886	0.943	0.914

to train a Rasa NLU model to be applied to a chatbot's pipeline. We experimented with different embedding models to represent the sentences in the dialogues, such as FLAIR, LaBSE, Glove, BERTimbau, and MUSE. The Glove and FLAIR word embeddings offered better score in the DBS and SS metrics, respectively, indicating better partitioning. However, when performing the visual analyses of the produced clusters, using word clouds and t-SNE techniques, the LaBSE sentence embedding was the one that better partitioned the dataset allowing for the identification of five intentions more clearly. Furthermore, these five intentions work better to build a chatbot to separate patients among mild, moderate or severe COVID-19 conditions as a proactive intelligent agent.

Complementarily, we used the SINTOMATIC tool to identify COVID-19 symptoms in the data and then annotate the NLU training dataset with these labels as well as was done with the intentions. Finally, when training the NLU model with DietClassifier implemented on the Rasa framework, we obtained an accuracy score of approximately 0.869, with good precision, recall, and F1-Score when we used the LaBSE embedding. As the future direction, we intend to evaluate the clustering algorithm with different embedding models. Moreover, we will experiment with other classification algorithms to train the NLU model in a goal-oriented chatbot pipeline.

ACKNOWLEDGMENT

The research reported in this work was supported by the Cearence Foundation for Support of Research (FUNCAP) project "Big Data Platform to Accelerate the Digital Transformation of Ceará State" under the number 04772551/2020.

REFERENCES

[1] Stuart J Russell and Peter Norvig. *Inteligência artificial*. Elsevier, 2004.
[2] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *IFIP AIAI*, pages 373–383. Springer, 2020.
[3] Maria Carolina Monard and José Augusto Baranauskas. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32, 2003.
[4] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
[5] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*, 2019.
[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.
[7] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196. PMLR, 2014.
[8] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.
[9] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of NAACL (Demonstrations)*, pages 54–59, 2019.
[10] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th BRACIS*, 2020.
[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.
[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
[13] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*, pages 670–680, 2017.
[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.
[15] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE TNNLS*, 2020.
[16] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd ACL-IJCNLP (volume 1: Long papers)*, pages 1681–1691, 2015.
[17] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.
[18] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR*, pages 1371–1374, 2018.
[19] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks, 2016.
[20] Ye-Yi Wang, Li Deng, and Alex Acero. Spoken language understanding. *IEEE Signal Processing Magazine*, 22(5):16–31, 2005.
[21] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013.
[22] Hannah Lei, Weiqi Lu, Alan Ji, Emmett Bertram, Paul Gao, Xiaoqian Jiang, and Arko Barman. Covid-19 smart chatbot prototype for patient monitoring, 2021.
[23] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020.
[24] Bettina Fazzinga, Andrea Galassi, and Paolo Torroni. An argumentative dialogue system for covid-19 vaccine information, 2021.
[25] Adam S Miner, Liliana Laranjo, and A Baki Kocaballi. Chatbots in the fight against the covid-19 pandemic. *NPJ digital medicine*, 3(1):1–4, 2020.
[26] Ticiana L Coelho da Silva, Marianna Gonçalves F Ferreira, Regis Pires Magalhaes, José Antônio F de Macêdo, and Natanael da Silva Araújo. Rastreador de sintomas da covid19. *SBBD*, 2020.
[27] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE TKDE*, 2020.
[28] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.
[29] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *JCAM*, 20:53–65, 1987.
[30] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
[31] Martin Ester, Hans-Peter Kriegel, Jörg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.